

Film Világ

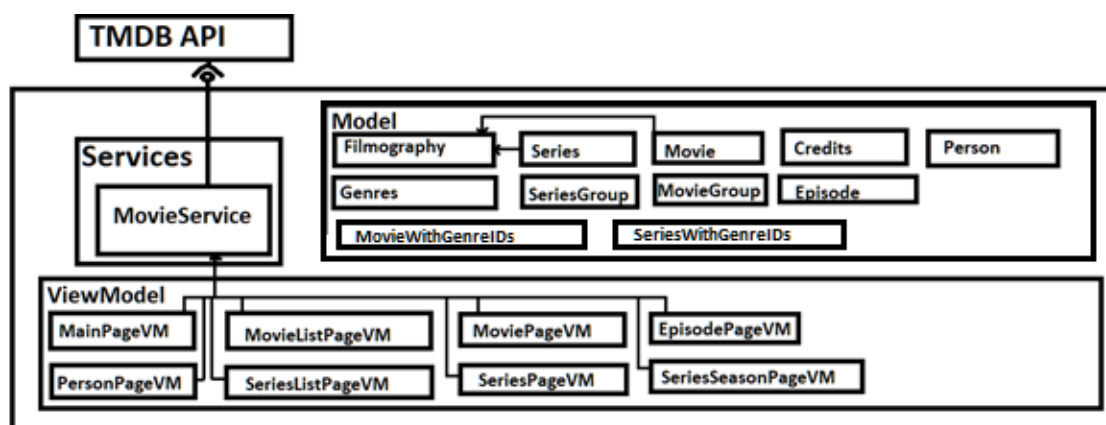
Kliensoldali technológiák házi feladat – Készítette: Csik Levente Balázs (VR6H0R)

Készíts egy alkalmazást, mely általános segítséget ad az esti filmválasztásban. Az alkalmazás tegye lehetővé, hogy kulcsszavak alapján megkeressünk filmeket, sorozatokat. A találati listából kiválasztott elemhez jelenítse meg a részletes adatlapját legalább 5-6 fontosabb adatot kiemelve, továbbá sorolja fel a szereplőket, sorozatok esetében pedig az epizódokat (amennyiben az API biztosítja ezt). Az alkalmazás tegye lehetővé a böngészést, így pl. legyen a színészeknek is saját adatlapjuk, melyen a színész fontosabb adatai és a szerepei jelennek meg. Ezen felül az alkalmazás adjon tippeket a filmválasztáshoz, így például tegye lehetővé általánosan a legnépszerűbb filmek, vagy éppen egy adott filmhez/témához/műfajhoz kapcsolódó legnépszerűbb filmek listázását is (az API lehetőségei szerint). A feladat megoldásához a következő 2 API valamelyikét ajánljuk:

- <http://docs.trakt.apiary.io/>
- <https://www.themoviedb.org/documentation/api?language=hu>

Architektúra:

Architektúrát az UWP frontenddel valósítottam meg és a themoviedb.org API-ját használtam backendként.



Szolgáltatások:

A MovieService szolgáltatás végzi az API és a program közötti kommunikációt. Lekéri a megadott paraméterek alapján az API-ból az információkat.

Fontosabb modellek:

A Filmography őssztályból származik le a Series és a Movie osztály, a Filmography a Movie és a Series közös elemeit tartalmazza.

Series: A sorozatokról tárol bőséges információt ez a modell. Többek között a leírásokat tartalmazza, az epizódok és évadok számát, azt hogy befejezték-e a sorozatot, vagy még továbbra futó sorozat.

Movie: A modell a filmekről tárol információkat, többek között a leírást, a színészeket, azt hogy felnőtteknek készült-e és a film hosszát.

A SeriesGroup-ra és a MovieGroup-ra azért volt szükség, mert amikor megadott paraméterek alapján filmet keresünk akkor az eredmény az API-ról olyan formátumban jön vissza, hogy a filmek/sorozatok listája és még három integer változó (találatok száma, oldalak száma stb.) és ezeknek a szűrésére

hoztam létre a két csoport osztályt. A többi modellre egy-egy követelmény funkció megvalósításához volt szükség.

A ViewModellek:

Minden oldal külön ViewModelt kapott, készítettem külön View-t arra, amikor egyes filmeket/sorozatokat jelenítünk meg, és arra, hogy ha a film/sorozat nevekből egy listát kapunk, amire kattintva, vagy a felső szövegmezőbe a nevét beírva tudjuk elérni az adott film/sorozat részletes leírását. A filmek esetén böngészési lehetőségként a film színészeire tudunk kattintani, ami az adott színész oldalát nyitja meg, ahol a sorozatait és a filmjeit megtekinthetjük és azokat is megnyithatjuk. Sorozatok esetén lehetőség nyílik az adott évadok megtekintésére, és az évadokon belül az epizódokat megtekinteni. Ezt vagy úgy lehet, hogy az évad oldalán a keresett epizód számot írjuk be, vagy pedig, ha már az epizódokat nézzük, akkor a Previous és Next gombok segítségével az előző és következő epizódot tekinthetjük meg.

Nézetek:

A főoldalon két opcióra nyílik lehetőség, az első az, hogy megadhatjuk a keresésünk paramétereit, és utána a Discover gombra kattintva megkeresi a program azokat, amik kielégítik a paramétereket vagy pedig lekérhetjük a heti vagy napi legnépszerűbb filmeket vagy sorozatokat.

Bármelyiket választjuk, a program kilistázza az eredményeket, és vagy a fenti szövegmezőbe a keresett film nevét begépelve majd a Check out this movie gombra kattintva, vagy pedig magára a film nevére kattintva megjeleníthetjük a részletes leírást.

Filmek esetében a sztorin és a nevéen kívül még van információ a hosszáról, arról, hogy felnőtteknek szóló film-e, a népszerűségéről, megjelenés dátumáról, a weboldaláról, valamint a színészekről, sorozat esetében pedig néven, sztorin, népszerűségen kívül arról, hogy még fut-e a program, illetve, hogy milyen típus-ba kategorizálták. Ezen felül van egy évad kiválasztó rész, ahová begépelhetjük a kiválasztott évadot majd a mellette lévő gombra kattintva megeleveníti az évadról szóló részletes információkat.

Az évadnál nincs sok információ, a sorozat nevéen, évad számán és a sztorin kívül csak az epizódok számát jeleníti meg, ha elérhető az információ. Ezek után kiválaszthatjuk, hogy melyik epizódot nyitjuk meg, vagy a Previous Next gombokra kattintva a következő évadokat tekinthetjük meg.

Film Világ – API

A (<https://www.themoviedb.org/documentation/api?language=hu>) API-t választottam, ahová egy regisztráció után kaptam egy kulcsot, amivel azonosítani tudom magamat, így elérhetővé válik az API-ból való információ lekérés. A megvalósítás főbb részletei a következők:

A gyakorlati mintákból indultam ki, először is elkészítettem egy nézetet az egyes filmeknek és sorozatoknak, majd a főoldalon megcsináltam a keresés-t, amihez szükségessé vált egy lista, ahová rakhatja az eredményeket a program, így a következő lépésben a MovieListPage és a SeriesListPage készült el. Itt volt egy kis nehézség a kilistázással, de a gyakorlati példát követve sikerült elkészítenem a listát. A nehézség lényege abban állt, hogy az API nem a filmek listáját adja vissza, hanem egy osztályt aminek az egyik változója a lista, így szükség volt a MovieGroup és SeriesGroup osztályokra. Ezek után a sorozatok egyes epizódjainak készítettem oldalt, ezeknél egészen könnyen sikerült elkészíteni azt a lehetőséget, hogy az egyes epizódok között lépkedhessen a felhasználó. Az évadoknak az elkészítése valamennyivel körülményesebb volt, ugyanis sokáig bajlódtam azzal, hogy némely információt meg tudjam jeleníteni. A színészek oldal megtervezésére a tervezettnél több időt szántam, ugyanis az alias-nevek megjelenítéséhez külön módszert kellett alkalmaznom.

Példa: egy Film keresése, majd a film szereplőjének megtekintése, aztán az egyik sorozatának megtekintése.

1. Keressünk rá a filmekre: Mivel ez egy példa filmbemutató, ezért inkább a nehezebben kezelhető példát mutatom be: Mivel filmeket keresünk, adjuk meg Média-nak a „**movie**” szót, nyelvnek sok mindent lehetne megadni (pl. az en-US és a hu működik) Írjuk be, hogy „**hu**”.

A Sort by Descending csak az eredmények megjelenítési sorrendjét mutatja népszerűség alapján, mivel nagy adatbázisról van szó, elképzelhető, hogy teljesen más filmeket javasol, hogyha a növekvő sorrendbe állítjuk a keresőt.

Az adult mezővel azt pipálhatjuk be, hogy felnőtteknek szóló tartalmakat keresünk-e, vagy sem.

A Genre fül mezőben adjuk meg a film típust, legyen Action, mert azok a legélvezetesebb filmek általában, de lehetne Drama, Comedy meg ezekhez hasonlót is megadni.

Az average vote- nál lehetőségünk van olyan filmek szűrésére, amiknek az értékelése 10-es skálán nagyobb egy értéknél.

Van lehetőség a kulcsszavas keresésre, de úgy tapasztaltam hogy az adatbázis/API-ban hiába kérünk le filmeket, nem adnak filmet, feltehetőleg mert a filmekhez nem, vagy csak kevés kulcsszavakat állítottak be, így elég nehéz eltalálni egy olyan kulcsszót, amivel eredményt kapunk.

Discover Movies and Series from the TMDB database

Media:

Language:

Sort by descending: ☒

Adult: ☐

Genre:

Average votes are greater than:

Keywords (put ',' symbol between keywords)

Discover

Get the most trendy movies! Choose your media:

☐ TV:

☐ Movie:

You want to see the top hit of the day or the week?

☐ Daily

☐ Weekly

Get the most trending movies!

```
public async Task DiscoverMovie()
{
    Keyword.Replace(".", "%2C");
    var service = new MovieService();
    var discovered = await service.GetDiscoverMovieAsync(Media, Language, SortDecreasing, IsAdult, Vote, Keyword);
    var genreList = await service.GetGenres(Media);
    foreach (var item in genreList.genres)
    {
        if (Genre == item.name)
        {
            ChosenGenre = item;
        }
    }
    DiscoveredMovie.Clear();
    foreach (var item in discovered.results)
    {
        if (item.genre_ids.Contains(ChosenGenre.id))
        {
            DiscoveredMovie.Add((Movie)item);
        }
    }
    NavigationService.Navigate(typeof(MovieListPage), DiscoveredMovie);
}
```

Sajnos a típusra nem lehet a Discover-függvénnyel keresni, mert az API nem ad ilyenkor vissza filmeket. Az API-tól megkapott információkban azonban van egy genre_ids tulajdonság, amikből vissza kell fejteni az adott filmtípusokat, hogy a keresést lehetővé tegyük.

2. Lekérdezés, ekkor lekéri a filmek listáját az alábbi kérést az API felé:

```
public async Task<MovieGroup> GetDiscoverMovieAsync(string media, string language, string sort, bool adult, float vote, string keywords)
{
    return await GetAsync<MovieGroup>(new Uri(movieServerUrl, $"3/discover/{media}?api_key={my personal key}&language={language}&sort_by={sort}&include_adult={adult}&page=1&vote_average_gte={vote}&with_keywords={keywords}"));
}
```

3. Ezek után megjelenik a filmek listája. Kattintsunk rá a Bloodshotra.

4. Megnyitja a film részletes oldalát, és egy újabb kéréssel megszerzi a Bloodshot információit.

```
public async Task<Movie> GetMovieAsync(int id)
{
    return await GetAsync<Movie>(new Uri(movieServerUrl, $"3/movie/{id}?api_key={my api key}&language=en-US"));
}
```

5. Kattintsunk rá Vin Diesel-re mert egy igazán ismert színész, és kapunk róla részletes információkat
Lefut az alábbi kérés

```
public async Task<Person> GetPersonAsync(int id)
{
    return await GetAsync<Person>(new Uri(movieServerUrl, $"3/person/{id}?api_key= my api key &language=en-US"));
}
```

Vin Diesel

Aliases: , Mark Sinclair Vincent, Mark Vincent, Він Дізел, Вин Дизель, فان دیزل, วิน ดีเซล, ヴィン・ディーゼル, 빈 디젤, 馮·迪索, Βιν Ντιζελ, 范·迪塞尔

Story:
Vin Diesel is an American actor, producer, director, and screenwriter. He came to prominence in the late 1990s, and first became known for appearing in Steven Spielberg's Saving Private Ryan in 1998. He is most known for his portrayals of Riddick in The Chronicles of Riddick trilogy (2001–2013), and of Dominic Toretto in The Fast and the Furious film series (2001–present), two franchises in which he also acted as producer. He also starred in xXx (2002) and Sidney Lumet's Find Me Guilty (2006). His voice acting work includes Brad Bird's The Iron Giant (1999), the video games of The Chronicles of Riddick franchise, and the upcoming Guardians of the Galaxy adaptation of the Marvel comics of the same name. As a filmmaker, Diesel directed, wrote, produced and starred in the drama film Strays, as well as in two short films. He is also the founder of the production companies One Race Films, Racetrack Records, and Tigon Studios.

Adult: False
Popularity: 23.365 % liked him/her
Birth day: 1967-07-18
Homepage: http://vindiesel.com
Cast:

The Ellen DeGeneres Show

The Daily Show with Trevor Noah

The Tonight Show with Jay Leno

Web Junk 20

Late Night with Conan O'Brien

The Late Late Show with Craig Kilborn

6. Kattintsunk rá a Late Night with Conan O'Brien-re mert erről már hallottam

Late Night with Conan O'Brien

Late Night with Conan O'Brien

Story:
Late Night with Conan O'Brien is an American late-night talk show hosted by Conan O'Brien that aired 2,725 episodes on NBC between 1993 and 2009. The show featured varied comedic material, celebrity interviews, and musical and comedy performances. Late Night aired weeknights at 12:37 am Eastern/11:37 pm Central and 12:37 am Mountain in the United States. From 1993 until 2000, Andy Richter served as O'Brien's sidekick; following his departure, O'Brien was the show's sole featured performer. The show's house musical act was The Max Weinberg 7, led by E Street Band drummer Max Weinberg.

The second incarnation of NBC's Late Night franchise, O'Brien's debuted in 1993 after David Letterman, who hosted the first incarnation of Late Night, moved to CBS to host Late Show opposite The Tonight Show. In 2004, as part of a deal to secure a new contract, NBC announced that O'Brien would leave Late Night in 2009 to succeed Jay Leno as the host of The Tonight Show. Jimmy Fallon began hosting his version of Late Night on March 2, 2009.

Popularity: 13.15 % liked it
Type of series: Talk Show
Season number: 17 Episode number: 2922
Is it ongoing series: Ended minutes
Homepage:

Which seasons do you want to check out? Check out!

Return to Menu

Amikor kiválasztjuk a keresett évadot, az alábbi programkód fut le

```
public void NavigateToSeason()
{
    SeasonToCheckOut.Add(Series.id);
    SeasonToCheckOut.Add(SeasonNumberChosen);
    NavigationService.Navigate(typeof(SeriesSeasonPage), SeasonToCheckOut );
}
```

Egy ObservableCollection-be gyűjtjük azokat a paramétereket amik kellenek az API-nak, és azokat adjuk át a navigáláskor a másik ViewModell-nek.

7. Nézzük meg az első évadát

Late Night with Conan O'Brien

Name of the Season: Season 1

Number of the Season: 1

Story:

Episode Count: 0
 Check out first episode!
Previous Next

Az epizódra kattintva a korábban ismertetett ObservableCollection módszerrel adjuk meg a szükséges paramétereket

```
public void NavigateToEpisode()
{
    SeasonToCheckOut.Clear();
    SeasonToCheckOut.Add(Series.id);
    SeasonToCheckOut.Add(Season.season_number);
    SeasonToCheckOut.Add(Episode);
    NavigationService.Navigate(typeof(EpisodePage), SeasonToCheckOut);
}
```

8. Láthatóan nem sok információ van róla, de hátha van információ az egyes epizódokról.

John Goodman, Drew Barrymore, Tony Randall

Season 1

Episode 1

Story:
Conan interviews John Goodman, Drew Barrymore & Tony Randall. George Wendt leg wrestles John Goodman & Conan sings a song.

Air Date: 1993-09-13

Previous

Next

Return to Series

Az epizód lekérdezésekkor az alábbi kérés megy az API felé:

```
public async Task<Episode> GetSeriesEpisodeAsync(int id, int season_number, int episode_number)
{
    return await GetAsync<Episode>(new Uri(movieServerUrl, $"3/tv/{id}/season/{season_number}/episode/{episode_number}?api_key= my api key &language=en-US"));
}
```

9. Nézzük meg a következő epizódokat a next-gombra kattintva: (A Pokémonnak és a Rick and Morty-nak sokkal szebben, gondosabban van elkészítve az API-ban a dokumentuma)

Mary Tyler Moore, Chris O'Donnell, Radiohead

Season 1

Episode 2

Story:
Mary Tyler Moore, Chris O'Donnell, Radiohead

Air Date: 1993-09-14

Previous

Next

Return to Series

Mercedes Ruehl, Chris Connelly, Mary Matalin

Season 1

Episode 3

Story:

Air Date: 1993-09-15

Previous

Next

Return to Series

```
public void NavigateToNextEpisode()
{
    int episode = Episode.episode_number+1;
    EpisodeToCheckOut.Add(Series.id);
    EpisodeToCheckOut.Add(Season.season_number);
    EpisodeToCheckOut.Add(episode);
    NavigationService.Navigate(typeof(EpisodePage), EpisodeToCheckOut );
}
```

10. Végül visszalépve a sorozathoz további évadokat tekinthetünk meg, vagy visszatérhetünk a főoldalra