

Trónok Harca Adatbázis

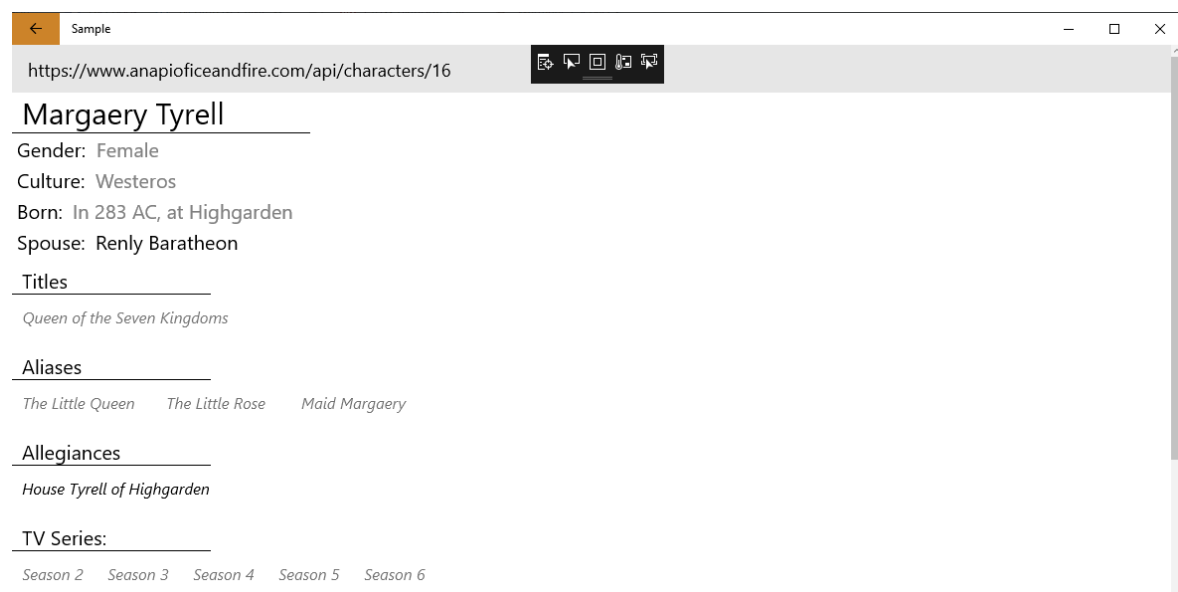
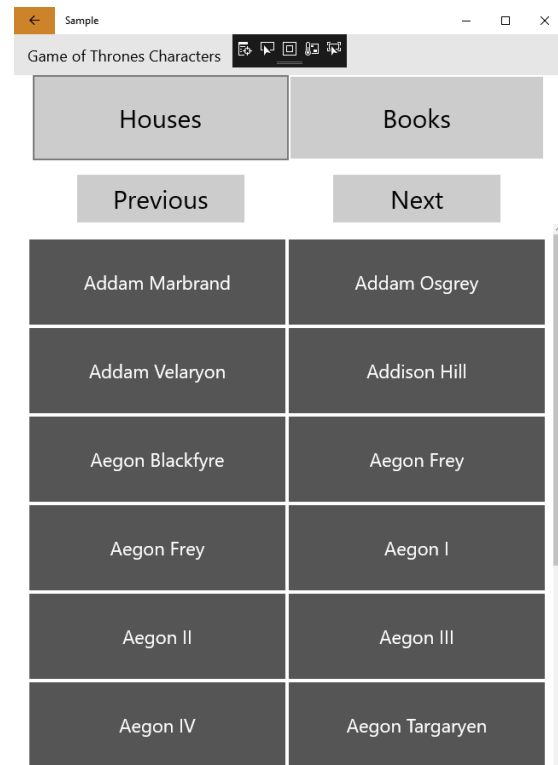
Kliensoldali Technológiák Házi Feladat
Készítette: Hajas András (CN98HE)

Összefoglaló

A Trónok Harca Adatbázis alkalmazás lényege, hogy bemutassa a Trónok harca könyvsorozat kiadásait, szereplőit, valamint a megjelenő házakat és családokat. Az alkalmazásban interaktívan és ergonomikus módon bejárhatjuk a Trónok harca világát, ugyanis az alkalmazásban lehetőség van az előbb felsoroltak listázására és megtekintésére, valamint köztük való navigálásra.

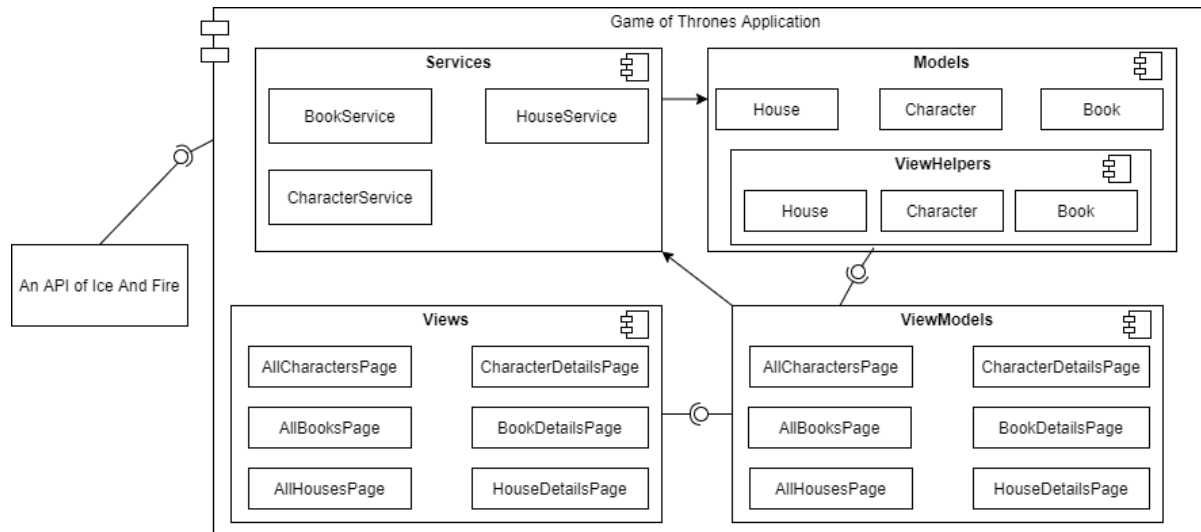
Továbbá bármely listázott elem kiválasztásával egy hozzá tartozó részletes nézetre kerülhetünk. Ezen a részletező oldalon megjelenik az adott tárgy minden adata, továbbá a kapcsolódó entitások is elérhetőek, amennyiben szerepel az adatbázisban a neki megfelelő részletes nézetre válthatunk.

Az alkalmazás az "An API of Ice And Fire" (<https://anapioficeandfire.com/api>) API-re épül, amelyet világ legjobb trónok harca strukturált adattárának tartanak.



Architektúra

Az megvalósítás alatt a frontendet Universal Windows Platform alkalmazásként készítettem el. Backendnek a “An API of Ice And Fire” API-t szolgáltatásait használtam. A kialakítás során az MVVM mintát próbáltam követni. A rendszer felépítése és az elemek főbb kapcsolási pontjai a következők:



Models

Az alkalmazásban tárgyalt entitások állapotának tárolásáért felelősek.

- **Character**
- **House**
- **Book**

Views

A megjelenítésért felelős komponensek. A nézet, amit a felhasználó lát a képernyőn

- **AllCharactersPage**
- **AllBooksPage**
- **AllHousesPage**
- **CharacterDetailsPage**
- **BookDetailsPage**
- **HouseDetailsPage**

ViewModels

Nézet absztrakciója, ami a publikus tulajdonságokat és metódusokat tartalmazza. Közvetít a nézet és az adatok között

- **AllCharactersPageViewModel**
- **AllBooksPageViewModel**
- **AllHousesPageViewModel**
- **CharacterDetailsPageViewModel**
- **BookDetailsPageViewModel**
- **HouseDetailsPageViewModel**

Services

Az hálózati hívások kezelését végző osztályok, minden Modelnek a saját specifikus tulajdonságaival.

- **CharacterService**
- **HouseService**
- **BookService**

```
private async Task<T> GetAsync<T>(Uri uri)
{
    using (var client = new HttpClient())
    {
        var response = await client.GetAsync(uri);
        var json = await response.Content.ReadAsStringAsync();
        T result = default(T);
        try
        {
            result = JsonConvert.DeserializeObject<T>(json);
        }
        catch (WebException ex)
        {
            Debug.WriteLine(ex.ToString());
        }
        return result;
    }
}
```

Models.ViewHelper

A részletes nézeteken az esetleges nem kitöltött adatok mezőinek eltüntetésében segít

- **CharacterViewHelper**
- **HouseViewHelper**
- **BookViewHelper**

Fontosabb komponensek

Az egy mappába tartozó osztályok megvalósítása nagyon hasonló, a funkciója mindnek ugyanaz, csak másféle adatokat kezel.

<>DetailsPageViewModel

Az oldalra navigáláskor egy url-t kap paraméterként ahonnan lekéri az entitás részleteit. Néhány tulajdonság, aminek az értéke további az API-ban szereplő egyedek további url-eket tartalmaznak. A tárgyalt objektum letöltése után elkezdődik ezen további url-ek aszinkron módon történő letöltögetése is. Azok a tulajdonságok, amikhez több elemet tartalmaznak ObservableCollectionbe fogja betölteni. Az opcionális cellaméret érdekében a több elemet tartalmazó tulajdonságokat hossz szerint csökkenő sorba rendezi. Ellenőrzi az üres tulajdonságokat és menti az eredményt a ViewHelperben ez alapján fogja eldönteni a View, hogy mely tulajdonságokat kell megjelenítenie. Ezen kívül navigációs függvényeket tartalmaznak.

A Könyvek nézetéhez tartozik két gomb, amivel a szereplő karaktereket tudjuk töltögetni, erre azért van szükség, hogy ne kelljen több száz kérést küldeni, csak ha szükséges. Ehhez tartozik egy számláló, ami az API Pageing-et kihasználva az oldalszámot menti csak és két függvény ami ezt a számláló növeli, illetve csökkenti.

```
public override async Task OnNavigatedToAsync(  
    object parameter, NavigationMode mode, IDictionary<string, object> state)  
{  
    var charUrl = (string)parameter;  
    var service = new CharacterService();  
    Character = await service.GetCharacterAsync(charUrl);  
  
    Array.Sort(Character.aliases, (y, x) => x.Length.CompareTo(y.Length));  
    Array.Sort(Character.titles, (y, x) => x.Length.CompareTo(y.Length));  
    Array.Sort(Character.allegiances, (y, x) => x.Length.CompareTo(y.Length));  
  
    checkVisibility();  
  
    await LoadPovBooks();  
    await LoadBooks();  
    await LoadSpouse();  
    await LoadAllegiances();  
  
    RaisePropertyChanged("Character");  
  
    await base.OnNavigatedToAsync(parameter, mode, state);  
}
```

All<>PageViewModel

Az oldalra navigáláskor letölti az adott típusú entitásokat egy ObservableCollection<*> tárolóba és ezt jeleníti meg a hozzá tartozó View-n. Karakterek és könyvek esetén itt is csak egy bizonyos darabszámú elemet tölt le egyszerre a kéréssel és számon tartja a jelenlegi oldalszámot, amivel lépkedhetünk a találatok között.

Ezen kívül a rendelkezik három navigálásra szolgáló függvénnyel, amivel a másik kettő felsorolás nézet között lépkedhetünk, valamint kiválasztva egy elemet a listából a hozzá tartozó részletes nézetre vezet.

An API of Ice And Fire

A feladat során használt egyszerű API. A karakterekről, a családokról és a könyvekről kérhetünk le információkat vele. Ez egy nyílt API, nincs szükség semmilyen autentikációra, hogy elérjük. Kizárólag GET metódust támogat, mivel nincs szükségünk és értelme se lenne az adatokon változtatni, amit előre elkészítettek nekünk.

Támogatja a lapozhatóságot, azaz nem egyben kapjuk vissza az adathalmazt, hanem ahogy a lista nézetek kialakításán is látszik egy előre megadott oldalszámmal végig lapozhatjuk az összes adatot. Hátrány, hogy a napi független hívások számát 20000 darab hívásban korlátozták IP címenként, ezzel megelőzve rosszindulatú felhasználást.

A válaszokat JSON üzenetben küldi.

List all books:

```
$ curl "https://www.anapiofireandice.com/api/books"
```

Example response:

```
[
  {
    "url": "https://www.anapiofireandice.com/api/books/1",
    "name": "A Game of Thrones",
    "isbn": "978-0553103540",
    "authors": [
      "George R. R. Martin"
    ],
    "numberOfPages": 694,
    "publisher": "Bantam Books",
    "country": "United States",
    "mediaType": "Hardcover",
    "released": "1996-08-01T00:00:00",
    "characters": [
      "https://www.anapiofireandice.com/api/characters/2",
      ...
    ],
    "povCharacters": [
      "https://www.anapiofireandice.com/api/characters/148",
      ...
    ]
  },
  ...
]
```

Források

A diagramok a <https://www.draw.io/> segítségével készültek.

A JSON válasz a <https://anapiofireandice.com/Documentation> oldalról.