# Angular Application – Movie DB (part 1)

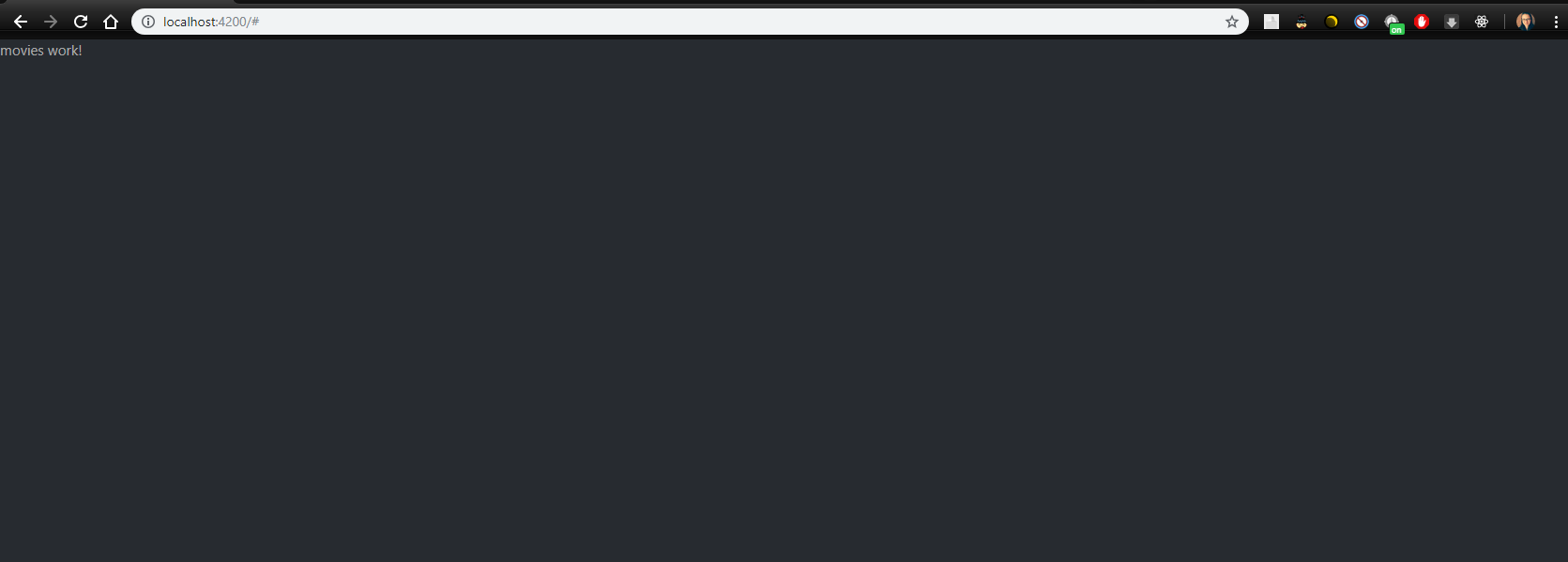
Exercise and homework for the ["Angular Fundamentals" course @ "SoftUni"](https://softuni.bg/courses/angular-2-fundamentals)

## Create new application

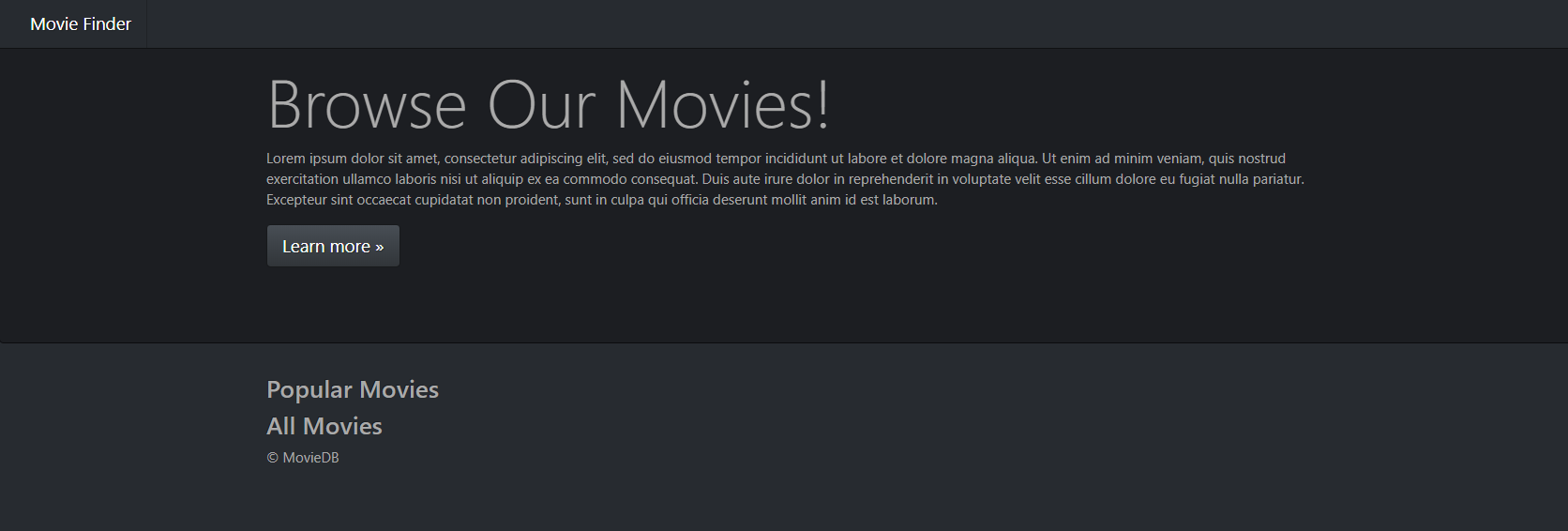
Using the CLI create new angular app typing in console "**ng new movie-finder**". That will create a new angular app. To make sure everything is up and tuning, navigate in console to the project "src" directory and there type "ng serve –open" → that will start the app on **localhost: 4200** and will open it in browser for you.

## Create components and add style

Type in the console "**ng g c movies**" (that is short command for ng generate component movies). Make sure that the new component is added in "**app.module**". Go to [*https://bootswatch.com/*](https://bootswatch.com/) and search for "Slate" (if you want you could choose different). Click on it and copy this link from URL [**https://bootswatch.com/4/slate/bootstrap.min.css**](https://bootswatch.com/4/slate/bootstrap.min.css). After that go to **index.html** and post this in the head part   
**<link rel="stylesheet" href="**[**https://bootswatch.com/4/slate/bootstrap.min.css**](https://bootswatch.com/4/slate/bootstrap.min.css)**">**.  
After that go to **app.component.html** delete everything and type **<app-movies></app-movies>**. Save the whole project and you should see this on the browser:



Now get the html of the component that is given to you and **paste** it in the **movies.component.html** file. Now if you start the app you should see this:

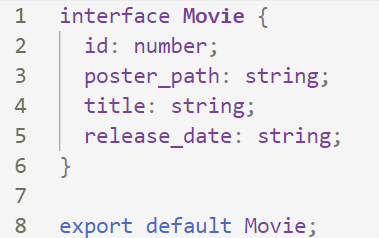


## Sign up

Now go to [moviedb](https://www.themoviedb.org/) and create an account. Go to your account in **settings** and navigate to **API**. There you have to make a **Request**. Choose the **'Developer'** option and **fill in the form**. Then you will see your **API Key**

## Create a Movie interface

Create a separate folder called **models**. In that folder create a js file holding an interface that will be our **template** for each movie



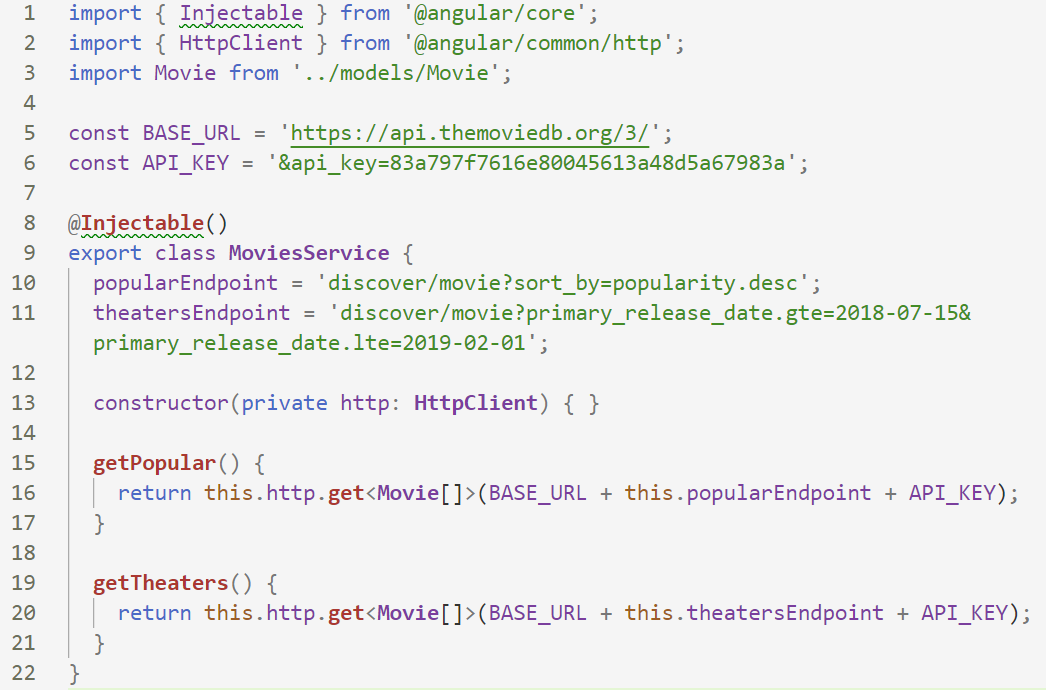
## Create a service and connect to the API

We will create a **service** in terminal which will be responsible for **fetching the data** from the API.

From the terminal in "src/app" type "**mkdir service"**, after that "**cd service"** and then type "**ng g s movies"**. That will create a service.

*Reminder: Do not forget to import the service and register it as a provider in* ***app.module****. In order to make requests to the API we should add to our project* ***HttpClientModule****. After that we should import* ***HttpClient*** *in our service and* ***inject*** *it through the constructor.*

In the service make **constant** which will hold our **api\_key**.



After that go to out movies component and inject as a dependency (through the constructor) the movies service we just created.



**Popular** will store all the **popular movies**, **theaters** will store **theaters** **the movies**

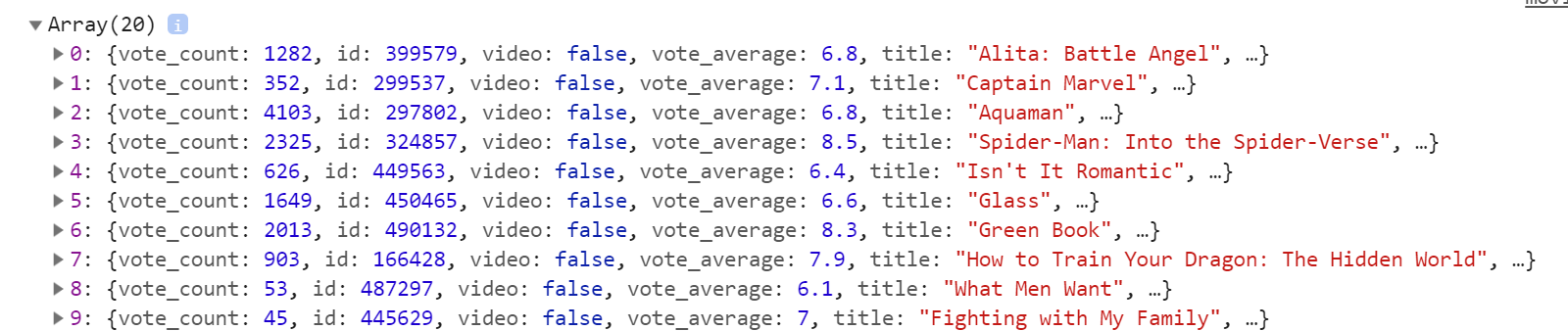
Ok, now let’s make our first call to the API.

Let’s go to the service and create a **function** which will be responsible for calling the API for the most popular movies.

The function should look like this:

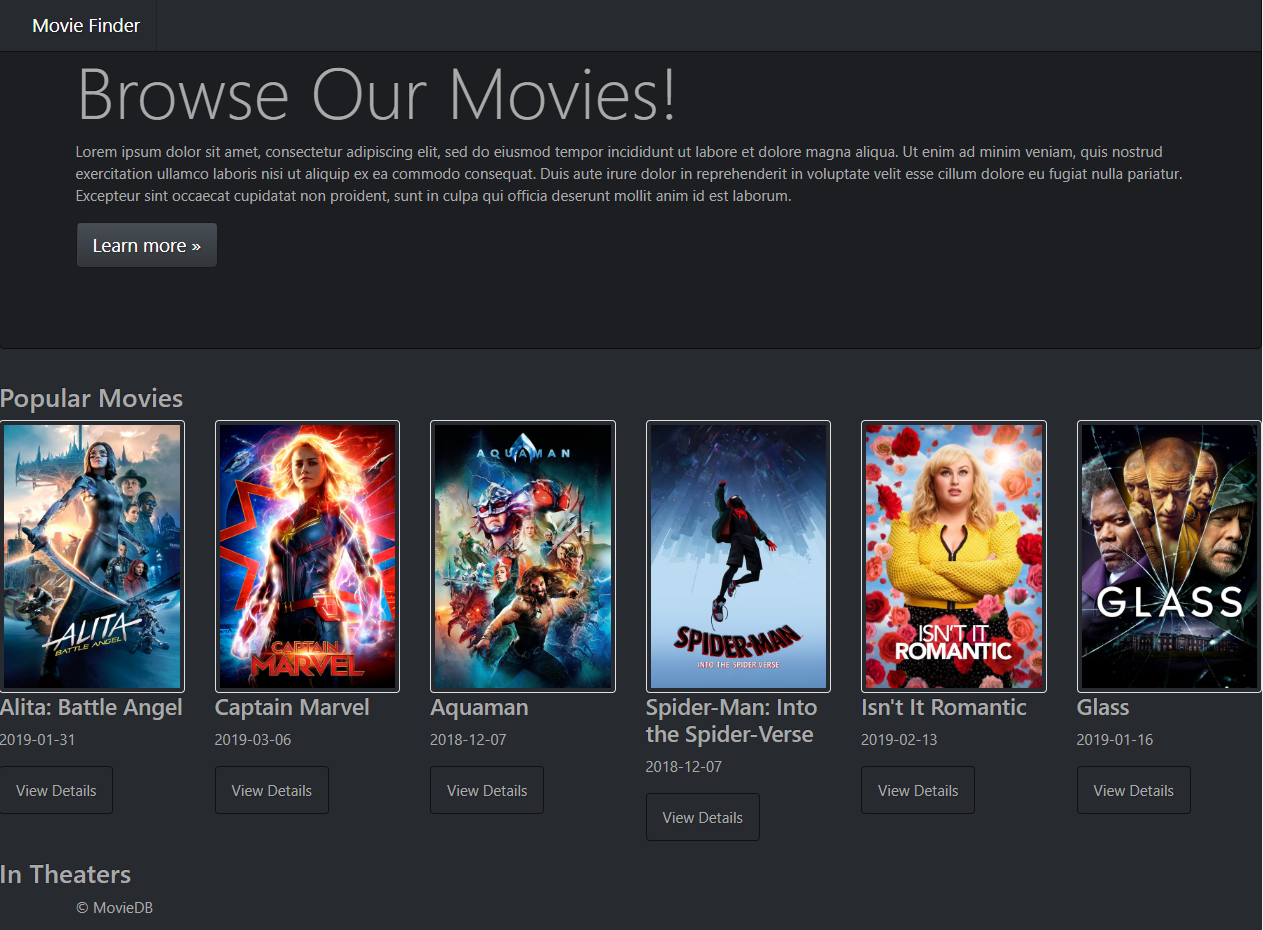


Check the **browser** **console** to see how the information is structured:



## List the popular movies in the view

After doing that, reload the page and you should see this:

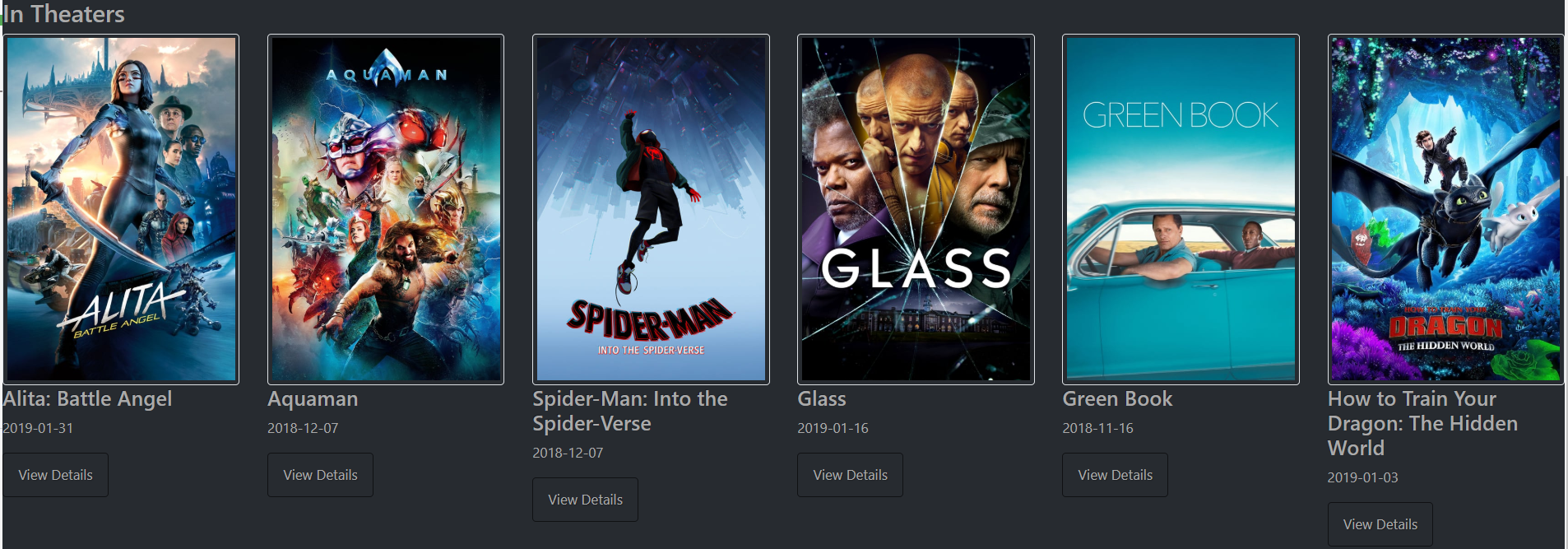


## List theater movies in the view

Add another function to get the **theater** movies:



Refresh and you should also see this:



## Create a MovieInfo Component

Let us separate the all the movie info into a component. Create a **movie-info** component and in the html file add this:

**<img \*ngIf="movie.poster\_path" class="img-thumbnail" src="http://image.tmdb.org/t/p/w500/{{movie.poster\_path}}">**

**<h4>{{movie.title}}</h4>**

**<p>{{movie.release\_date}}</p>**

**<p>  
 <a class="btn btn-default" routerLink="movie/{{movie.id}}">View Details</a>  
</p>**

Then in the **movie-info component**:



Then in **all of the containers** with movies, replace the old html with this:



In the next part we will add **routing** to our website to create an actual **Single-Page Application**.