Angular Application – Movie DB (part 2)

We will continue working on our previous project.

Implement search logic

Go to app.component.html and copy the jumbotron section. Remove it from there and paste it in movies.component.html. Make changes to see the component look like this:

The whole html:

<div class="jumbotron">

<div class="container">

<h1>Find a movie</h1>

<p>Search for a movie using the form below:</p>

<form action="">

<input type="text" class="form-control">

<button >Search</button>

</form>

</div>

</div>

<div class="panel panel-default">

<div class="panel-heading">

<h3 class="panel-title">Popular Movies</h3>

</div>

<div class="panel-body">

<div class="row">

<div \*ngFor="let movie of popular.results;let i=index" class="col-md-2">

<div \*ngIf="i < 6">

<img \*ngIf="movie.poster\_path" class="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" href="#">View Details</a></p>

</div>

</div>

</div>

</div>

</div>

<div class="panel panel-default">

<div class="panel-heading">

<h3 class="panel-title">In Theaters</h3>

</div>

<div class="panel-body">

<div class="row">

<div \*ngFor="let movie of theaters.results;let i=index" class="col-md-2">

<div \*ngIf="i < 6">

<img \*ngIf="movie.poster\_path" class="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" href="#">View Details</a></p>

</div>

</div>

</div>

</div>

</div>

List more categories

Using your knowledge from previous exercise list two more section. One for popular kids movies and one for the best drama movies.

Your project should look like this:

and for drama:

Prepare for navigation

When we click on “view details” button we should be redirected to another page and be able to see the details for the selected movie.

To achieve that we will need navigation. Go to the directory where the app.module is and create new file app.routing.ts

and import

Note that you should import your MoviesComponent and create a new component called MovieComponent which will be responsible to list the details for a single movie.

After that export a varriable of type ModuleWithProviders and assigned to be: RouterModule.forRoot({here place the array you just created})

Go to app module and add to imports the routing varriable we have just exported form the app.routing file.

Set up the movie component

We should import in our component the service and ActivatedRoute from “@angular/router” and inject them through the constructor:

Go to our service and create new function called getMovie(id) which will take a parameter (id of the movie) and return the call from the API for the movie.

Return to the movie component and in the OnInit method subscribe to the router’s parameters and take the id. Below call the service with the new function and give the id as an argument

Now we should implement the link. Go to movies.component.html and replace the href links in “view details” buttons with routerLink=”/movie/{{movie.id}}”

Do not forget to replace all in app.component.html in container with <router-outlet></router-outlet>

When we click on some of the “view details” buttons the page should like this:

In order to finish this part of application we should add HTML template for our MovieComponent

<div \\ensure that there is a movie in order to render this div>

<div class="panel panel-default">

<div class="panel-heading">

<h3 class="panel-title">{{place movie title here}}</h3>

</div>

<div class="panel-body">

<div class="row">

<div class="col-md-5">

<img class="thumbnail" src="http://image.tmdb.org/t/p/w500/{{place poster path here}}">

</div>

<div class="col-md-7">

<ul class="list-group">

<li class="list-group-item">Genres: <span \*ngFor="let genre of movie.genres">{{list all genres for this movie}}</span></li>

<li class="list-group-item">Release Date: {{show the release date of the movie}}</li>

</ul>

<br>

<a \*ngIf="movie.homepage" href="{{place movie's home here}}" target="\_blank" class="btn btn-default">Visit Movie Website</a>

</div>

</div>

</div>

</div>

</div>

\* Working search form (optional)

We make our UI for our search form but it does not include any functionality. So we are going to use forms here.

In app.module import FormsModule from '@angular/forms' and add FormsModule to the imports

Our HTML should look like this:

<form #form="ngForm" (submit)="search(form.value)">

<input type="text" class="form-control" name="search" ngModel>

<button type="sumbit" class="btn btn-primary">Search</button>

</form>

Next steps:

1. Go to our service and create a function findAMovie(name)

2. After that go to moves.component.ts and create a function which will take the string from user input and will call the new service function. Subscribe to it and list the results in the view.

3. Render the results only if the user has clicked search button.

The results should look like this