

Single-Page-Applications

Goal

In this lab, you will:

- Convert an application into an SPA
- Use the Angular-Router

Your mission

In this lab, you will update an Angular application to make use of the browsers navigation capabilities to support bookmarking and the back and forward buttons.

Type it out by hand?

Typing it drills it into your brain much better than simply copying and pasting it. You're forming new neuron pathways. Those pathways are going to help you in the future. Help them out now.

Display the list of movies based on the browsers URL

In this section, you will add the Angular router and use to display the list of movies.

1. Open **app.module.ts** and import **RouterModule** and **Routes** from **@angular/router**.

```
import { RouterModule, Routes } from '@angular/router';
```

2. Add the **movies** route to display the **MovieListItemComponent** component:

```
var routes: Routes = [  
  { path: 'movies', component: MovieListItemComponent }  
];
```

3. Import the **root** router module using the routing table defined in the previous step.

```
imports: [  
  BrowserModule,  
  ReactiveFormsModule,  
  HttpClientModule,  
  RouterModule.forRoot(routes)  
],
```

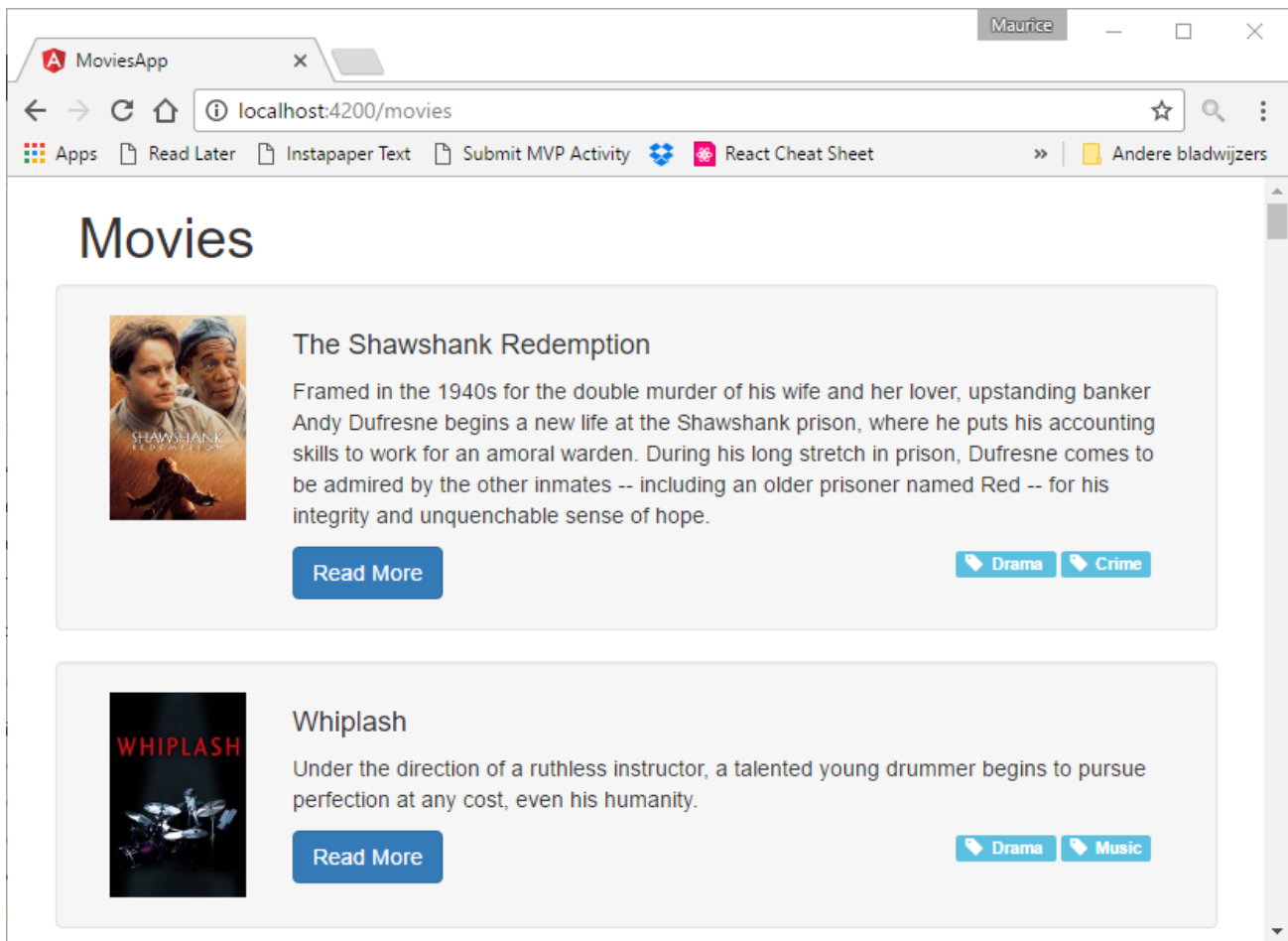
4. Open `app.component.html` and replace the markup with the `router-outlet`.

```
<div class="container">  
  <router-outlet></router-outlet>  
</div>
```

5. Open `app.component.ts` and delete the `selectedMovie` property and the `movieSelected()` and `showList()` functions.

```
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})  
export class AppComponent {  
  constructor() { }  
}
```

6. Navigate to <http://localhost:4200/> and you should see an empty page without any errors in the console window.
7. Navigate to <http://localhost:4200/movies> and you should see the complete list of movies again. Note that clicking on **Read More** has no effect.



Redirect to the movies list

In this section, you will redirect to the movies list if an unknown route is detected.

1. Open **app.module.ts** and add a wildcard route to **redirect** to the movies list.

```
var routes: Routes = [  
  { path: 'movies', component: MovieListItemComponent },  
  { path: '**', redirectTo: 'movies' }  
];
```

2. Navigate to <http://localhost:4200/> and observe that you are redirected to <http://localhost:4200/movies> and the movie list appears.

Add a route for an individual movie page

In this section, you will add a route for an individual movie page. You will also update the "Read More" button to navigate there.

1. Open **app.module.ts** and add a new **route** for the movie editor. Add a parameter named **movieId** to contain the movie **id** property.

```
var routes: Routes = [
  { path: 'movies', component: MovieListItemComponent },
  { path: 'movie/:movieId', component: MovieEditorComponent },
  { path: '**', redirectTo: 'movies' }
];
```

2. Open `movie-list-item.component.ts` and delete the `movieSelected` output property.
3. Import the `Router` service from `@angular/router`.

```
import { Router } from '@angular/router';
```

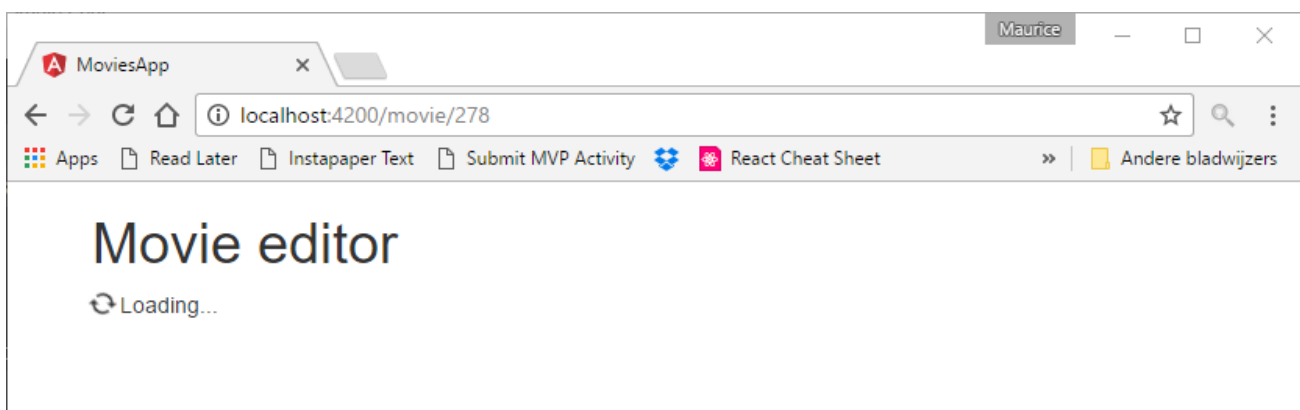
4. Inject the `Router` service into the `MovieListItemComponent` constructor.

```
constructor(private moviesService: MoviesService, private router: Router) { }
```

5. Replace the body of the `showMore()` function with a call to the router `navigate()` function.

```
showMore(movie: Movie) {
  this.router.navigate(['movie', movie.id])
}
```

6. Make sure you can navigate to the movie editor. Note: this page will not show the movie and produce an error. You will fix that in the next section.



Update the movie editor

In this section, you will fix the movie editor by retrieving the movie ID from the route data.

1. Open `movie-editor.component.ts` and delete the `movieId` and `showList` properties.

2. Import the **Router** and **ActivatedRoute** from **angular/router**.

```
import { Router, ActivatedRoute } from '@angular/router';
```

3. Add the **switchMap()** RxJS operator to the observable.

```
import 'rxjs/add/operator/switchMap';
```

4. Inject the **Router** and **ActivatedRoute** services into the **MovieEditorComponent** constructor.

```
constructor(  
  private moviesService: MoviesService,  
  private FormBuilder: FormBuilder,  
  private router: Router,  
  private activatedRoute: ActivatedRoute) {  
}
```

5. Update the **ngOnInit()** to read the **movieId** from the **ActivatedRoute** and use the **switchMap()** to trigger loading of the movie.

```
ngOnInit() {  
  // Previous code.  
  this.activatedRoute.params  
    .switchMap(params => this.moviesService.getMovie(params.movieId))  
    .subscribe(movie => {  
      this.movie = movie;  
      this.movieForm.reset(movie);  
      this.validateMovie();  
    });  
}
```

6. Update the **onCancel()** to use the **Router** service.

```
onCancel() {  
  this.router.navigate(['movies']);  
}
```

6. Update the **onSubmit()** to use the **Router** service

```
onSubmit() {  
  const updatedMovie = {...this.movie, ...this.movieForm.value};
```

```
this.moviesService.updateMovie(updatedMovie)
  .subscribe(() => this.router.navigate(['movies']));
}
```

7. The application should now be fully functional. Test it to see if everything is working as expected. Note: you can now refresh the browser while looking at the movie editor and the same page should reappear.

The screenshot shows a web browser window with the title 'MoviesApp'. The address bar displays 'localhost:4200/movie/278'. The browser's bookmark bar includes 'Apps', 'Read Later', 'Instapaper Text', 'Submit MVP Activity', 'React Cheat Sheet', and 'Andere bladwijzers'. A tooltip for 'Submit MVP Activity' is visible, containing a JavaScript snippet. The main content area is titled 'Movie editor' and contains a form with the following fields:

- Title:** A text input field containing 'The Shawshank Redemption'.
- Overview:** A text area containing the text: 'Framed in the 1940s for the double murder of his wife and her lover, upstanding banker Andy Dufresne begins a new life at the Shawshank prison, where he puts his accounting skills to work for an amoral warden. During his long stretch in prison, Dufresne comes to be admired by the other inmates -- including an older prisoner named Red -- for his integrity and unquenchable sense of hope.'
- Release date:** A text input field containing '1994-09-10'.

At the bottom of the form are two buttons: 'Save' (blue) and 'Cancel' (red).

Solution

The solution can be found in **complete** folder