

Yosra Souissi

Tp3: Service & Rest-API

Travail à faire:

1. Créer un projet

```
$ ionic start movies blank --verbose
```

2. Créer les pages suivantes «**movies**» «**moviedetails**»

```
ionic g page pages/Movies
```

```
ionic g page pages/movieDetails
```

3. Créer le service «**movieService**»

```
ionic g service services/movie
```

➤ **Qu'est ce qu'un service ?** Un service est généralement une classe avec un but étroit et bien défini. Généralement, les composants se limite à l'affichage et à la gestion des événements utilisateurs dans la vue du composant. L'exécution des traitements en local ou en back-end sont attribués aux services : le composant fait appel à des fonctions dans les services pour effectuer des traitements et fournir des résultats. C'est les services qui interagissent avec la partie back-end de l'application en envoyant des requêtes HTTP.

L'utilisation d'un service se fait via le principe de l'injection des dépendances.

4. Changer la page de démarrage par défaut «**home**» par la page «**movies**»

```
const routes: Routes = [  
  { path: "", redirectTo: "movies", pathMatch: "full" },
```

5. Importer HttpClientModule dans app.module.ts

Lorsque nous souhaitons passer des appels HTTP. Nous devons donc importer HttpClientModule à notre fichier app.module.ts et l'ajouter au tableau des importations.

```
@NgModule({
  declarations: [AppComponent],
  entryComponents: [],
  imports: [
    BrowserModule,
    IonicModule.forRoot(),
    AppRoutingModule,
    HttpClientModule
  ],
})
```

6. Pour utiliser la base de données OMDb API(The Open Movie Database), vous devez demander une clé API via ce lien :

<http://www.omdbapi.com/apikey.aspx>

Avec l'API, nous pouvons maintenant rechercher des chaînes et obtenir des résultats sous forme de films et d'épisodes. De plus, nous pouvons obtenir des informations détaillées sur un objet spécifique de ces résultats.

7. Récupérer les données de l'API dans notre service

```
url = "http://www.omdbapi.com/";
apiKey = " ";
```

Notre service n'a besoin que de 2 fonctions:

searchData (): Cette fonction recherche les résultats d'un titre et d'un type spécifiques.

getDetails (): Cette fonction retourne les informations détaillées pour un objet spécifique, sera utilisé sur notre page de détails.

8. Modifier la page « **movie.service.ts** »

```
export enum SearchType {
  all = "",
  movie = "movie",
  series = "series",
  episode = "episode"
}

@Injectable({
  providedIn: "root"
})
export class MovieService {
  url = "http://www.omdbapi.com/";
  apiKey = " "; // ← insérer votre apikey ici

  constructor(private http: HttpClient) {}

  searchData(title: string, type: SearchType): Observable<any> {
    return this.http
      .get(
        `${this.url}?s=${encodeURIComponent(title)}&type=${type}&apikey=${this.apiKey}`
      )
      .pipe(map(results => results["Search"]));
  }

  getmovies(): Observable<any> {
    return this.http.get(this.url);
  }

  getDetails(id) {
    return this.http.get(`${this.url}?i=${id}&plot=full&apikey=${this.apiKey}`);
  }
}
```

9. Injecter le service « **Movie** » dans le composant « **movies.ts** »

```
constructor(private movieService: MovieService, public navCtrl: NavController) {}
```

10. Modifier le contrôleur « **movies.ts** »

```
export class MoviesPage implements OnInit {
  results: Observable<any>;
  searchTerm = "";
  type: SearchType = SearchType.all;
  constructor(
    private movieService: MovieService,
    public navCtrl: NavController
  ) {}

  ngOnInit() {}

  searchChanged() {
    this.results = this.movieService.searchData(this.searchTerm, this.type);
  }
}
```

11. Modifier la vue « movies.html » afin de faire la liaison avec le composant « movies.ts »

```
<ion-header>
  <ion-toolbar color="primary">
    <ion-title>My Movie Search</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>

  <ion-searchbar [(ngModel)]="searchTerm" (ionChange)="searchChanged($event)">
</ion-searchbar>

  <ion-item>
    <ion-label>Select Searchtype</ion-label>
    <ion-select [(ngModel)]="type" (ionChange)="searchChanged($event)">
      <ion-select-option value="">All</ion-select-option>
      <ion-select-option value="movie">Movie</ion-select-option>
      <ion-select-option value="series">Series</ion-select-option>
      <ion-select-option value="episode">Episode</ion-select-option>
    </ion-select>
  </ion-item>

  <ion-list>

    <ion-item button *ngFor="let item of (results | async)"
[routerLink]="['/', 'movies', item.imdbID]">
      <ion-avatar slot="start">
        <img [src]="item.Poster" *ngIf="item.Poster != 'N/A'">
      </ion-avatar>

      <ion-label text-wrap>
        <h3>{{ item.Title }}</h3>
        {{ item.Year }}
      </ion-label>

      <ion-icon slot="end" *ngIf="item.Type == 'movie'" name="videocam"></ion-icon>
      <ion-icon slot="end" *ngIf="item.Type == 'series'" name="tv"></ion-icon>
      <ion-icon slot="end" *ngIf="item.Type == 'game'"
name="logo-game-controller-b"></ion-icon>

    </ion-item>

  </ion-list>

</ion-content>
```

12. Passer maintenant à la page « movieDetails », nous allons changer la route de cette page.

```
{
  path: "movies/:item.imdbID",
  loadChildren:
    "./pages/movie-details/movie-details.module#MovieDetailsPageModule"
}
```

13. Récupérer l'id passé via l'URL et stocker le dans la variable id.

```
ngOnInit() {
  let id = this.activatedRoute.snapshot.paramMap.get("item.imdbID");
}
```

14. Ensuite, récupérer les informations de l'API toujours au niveau de la fonction ngOnInit()

```
this.movieService.getDetails(id).subscribe(result => {
  this.information = result;
});
```

15. Créer la fonction OpenWebsite() qui permet d'ouvrir le site Web imdb.com à l'aide de l'objet window et des informations provenant des données de l'API stockées dans la variable information.

```
openWebsite() {
  let id = this.activatedRoute.snapshot.paramMap.get("item.imdbID");
  window.open("https://www.imdb.com/title/" + id + "/");
}
```

- Donc finalement notre composant « **movieDetails.page.ts** » doit être comme cet exemple :

```
import { Component, OnInit } from "@angular/core";
import { MovieService } from "src/app/services/movie.service";
import { Router, ActivatedRoute } from "@angular/router";

@Component({
  selector: "app-movie-details",
  templateUrl: "./movie-details.page.html",
  styleUrls: ["./movie-details.page.scss"]
})
export class MovieDetailsPage implements OnInit {
  information = null;
  constructor(
    public route: Router,
    private activatedRoute: ActivatedRoute,
    private movieService: MovieService
  ) {}

  ngOnInit() {
    let id = this.activatedRoute.snapshot.paramMap.get("item.imdbID");

    this.movieService.getDetails(id).subscribe(result => {
      this.information = result;
    });
  }

  openWebsite() {
    let id = this.activatedRoute.snapshot.paramMap.get("item.imdbID");
    window.open("https://www.imdb.com/title/" + id + "/");
  }
}
```

16. Finalement, on termine par l'affichage de la vue movie-details.

```
<ion-header>
  <ion-toolbar color="primary">
    <ion-buttons slot="start">
      <ion-back-button defaultHref="/"></ion-back-button>
    </ion-buttons>
    <ion-title>{{ information?.Genre }}</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content padding>

  <ion-card *ngIf="information">
    <ion-card-header>
      <ion-card-title>
        {{ information.Title }}
      </ion-card-title>
      <ion-card-subtitle>
        {{ information.Year }}
      </ion-card-subtitle>
    </ion-card-header>
    <ion-card-content text-center>
      <img [src]="information.Poster" class="info-img" />

      <ion-item lines="none">
        <ion-icon name="star-half" slot="start"></ion-icon>
        <ion-label>{{ information.imdbRating }}</ion-label>
      </ion-item>

      <ion-item lines="none">
        <ion-icon name="clipboard" slot="start"></ion-icon>
        <ion-label text-wrap>{{ information.Director }}</ion-label>
      </ion-item>

      <ion-item lines="none">
        <ion-icon name="contacts" slot="start"></ion-icon>
        <ion-label text-wrap>{{ information.Actors }}</ion-label>
      </ion-item>

      <ion-button expand="full" (click)="openWebsite()"
        *ngIf="information.Website && information.Website != 'N/A'">
        <ion-icon name="open" slot="start"></ion-icon>
        Open Website
      </ion-button>
    </ion-card-content>
  </ion-card>
</ion-content>
```

17. Ajouter un peu de css à la page « **movieDetails.scss** » comme suit :

```
.info-img {  
  max-height: 30vh;  
  object-fit: contain;  
  padding: 10px;  
}
```

18. Exécuter

