Angular 2 - Lab 1

Angular 2 - Router

1- Objectif :

Nous allons fournir avec ce Lab un projet «LabRouter» déjà développé avec Angular 2.

Le but du Lab sera d'intégrer le module « @angular/router» dans les composants et de rendre les liens du site dynamique.

2- Préparer le projet :

Télécharger le projet à partir de GitHub https://github.com/CHAABANERamy/LabRouter ou bien avec la commande git :

```
git clone https://github.com/CHAABANERamy/LabRouter.git
```

D'abord, il faut changer le répertoire :

```
cd LabRouter
```

Puis, Lancer la commande pour installer les packages manquants :

```
npm install
```

3- Règles de routages :

On va créer les règles de routage de chaque composant dans un fichier nommé « app.routing.ts » :

```
import { RouterModule } from '@angular/router';
import { LoginFormComponent } from './login/login-form.component';
import { ArtistListComponent } from './artist/artist-list.component';
import { ArtistDetailComponent } from './artist/artist-detail.component';
import { AlbumListComponent } from './album/album-list.component';
import { AlbumDetailComponent } from './album/album-detail.component';

export const routing = RouterModule.forRoot([
  {
    path: 'login',
    component: LoginFormComponent
  },
  {
    path: 'artists',
    component: ArtistListComponent
  },
  {
    path: 'artists/:artistId',
    component: ArtistDetailComponent
  },
  {
    path: 'albums',
    component: AlbumListComponent
  },
  {
    path: 'albums/:albumId',
    component: AlbumDetailComponent   },
  {
    path: '',
    redirectTo: '/artists',
    pathMatch: 'full'
  }
]);
```

Importer le fichier « app.routing.ts » dans le fichier « app.module.ts » :

```
import { routing } from './app.routing';
```

Et après, il faut l'ajouter dans le module « NgModuel »

```
@NgModule({
  imports: [BrowserModule, FormsModule,routing],
  declarations: [
    AppComponent,LoginFormComponent,ArtistListComponent,
    ArtistDetailComponent,AlbumListComponent,AlbumDetailComponent
  ],
  providers: [
    ArtistService,AlbumService,LoginService,
  ],
  bootstrap: [AppComponent]
})
```

Maintenant, on va ajouter la directive « router-outlet » dans le Template du composant principale « app.component.ts »

```
import { Component } from '@angular/core';

@Component({
  selector: 'angular-tunes',
  template: `
    <div class="container">
 <router-outlet></router-outlet>
    </div>
  `
})
export class AppComponent { }
```

Il nous reste seulement de définir les liens de chaque page à l'aide de la directive « routerlink » dans les pages suivantes :

- **artist-detail.component.html**

```
<h1>{{artist.name}}</h1>
<div class="row">
  <div class="col-md-4">
    <div class="thumbnail">
      <img [src]="artist.image">
      <div class="caption">
        <table class="table">
          <tr *ngFor="let album of artist.albums">
            <td>
              <a [routerLink]="['/albums', album.id]">{{album.title}}</a>
            </td>
            <td>{{album.year}}</td>
          </tr>
        </table>
        <div [innerHTML]="artist.description"></div>
      </div>
    </div>
  </div>
</div>
```

- **artist-list.component.html**

```
<h1>Artists</h1>
<table class="table">
  <tr *ngFor="let artist of artists">
    <td>
      <a [routerLink]="['/artists', artist.id]">{{artist.name}}</a>
    </td>
  </tr>
</table>
```

- **album-detail.component.html**

```
<h1>{{album.title}}</h1>
<p>
  <a [routerLink]="['/artists', album.artist.id]">{{album.artist.name}}</a>,
  {{album.year}}
</p>
<div class="row">
  <div class="col-md-4">
    <div class="thumbnail">
      <img [src]="album.image">
      <div class="caption">
        <div [innerHTML]="album.description"></div>
      </div>
    </div>
  </div>
</div>
```

- **album-list.component.html**

```
<h1>Albums</h1>
<table class="table">
  <tr *ngFor="let album of albums">
    <td>
      <a [routerLink]="['/albums', album.id]">{{album.title}}</a>
    </td>
    <td>{{album.year}}</td>
    <td>{{album.artist}}</td>
  </tr>
</table>
```

En fin, on doit configurer le chemin de base Href :

1. Solution HTML : dans la page index.html, il faut ajouter la balise « base » dans la balise « head » :

```
<base href="/">
```

2. Solution Type Script : dans le ficher « app.module.ts » :

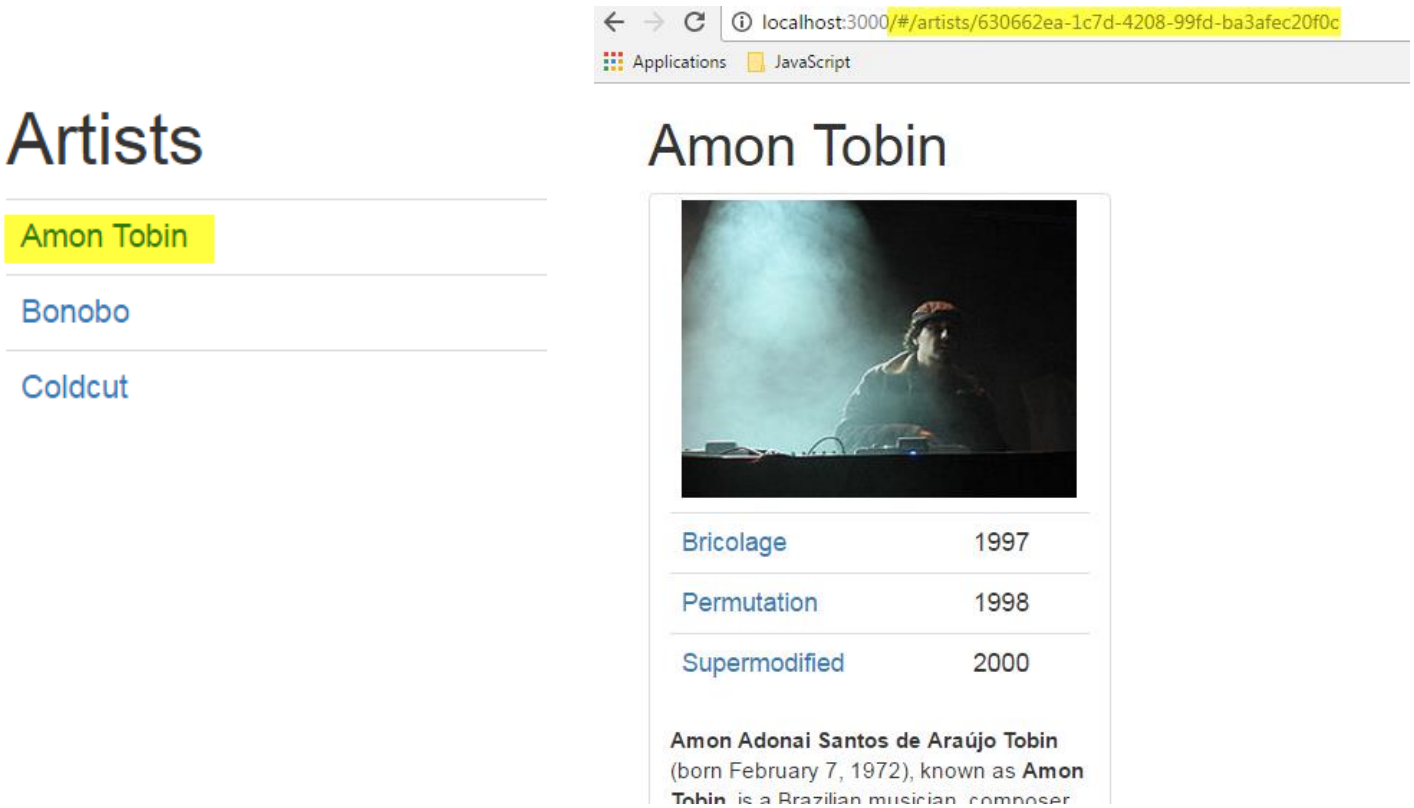
    - Importer « LocationStrategy » et « HashLocationStrategy » à partie de « @angular/common » :

```
import { LocationStrategy, HashLocationStrategy } from '@angular/common';
```

- Ajouter la configuration suivante dans « providers » de NgModule :

```
providers: [
  ArtistService,
  AlbumService,
  LoginService,
  {provide: LocationStrategy, useClass: HashLocationStrategy}
]
```

Maintenant vous pouvez naviguer entre les différents liens du site.



4- Paramètre Router :

Il peut être intéressant d'avoir des URL propres tout en transmettant des paramètres.

Et pour le récupérer, on utilise le service « ActivatedRoute » qui contient tous les paramètres récupérés dans l'URL. On va modifier les deux composants suivants :

- **Composant « artist-detail.component.ts » :**

```
import { Subscription } from 'rxjs/Subscription';
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { ArtistService } from './artist.service';

@Component({
  selector: 'artist-detail',
   templateUrl:  '../app/artist/artist-detail.component.html'
})
export class ArtistDetailComponent implements OnInit {

  artist;
  paramsSubscription: Subscription;

  constructor(private route: ActivatedRoute,
              private artistService: ArtistService) { }

  ngOnInit() {
      let artistId = this.route.snapshot.params["artistId"];
      this.artist = this.artistService.getArtist(artistId);
  }
}
```

- **Composant « album-detail.component » :**

```
import { Subscription } from 'rxjs/Subscription';
import { Component, OnInit, OnDestroy } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { AlbumService } from './album.service';

@Component({
  selector: 'album-detail',
  templateUrl:  '../app/album/album-detail.component.html'
})
export class AlbumDetailComponent implements OnInit, OnDestroy {

  album;
  paramsSubscription: Subscription;

  constructor(private route: ActivatedRoute,
              private albumService: AlbumService) { }

  ngOnInit() {
    this.paramsSubscription = this.route.params.subscribe(params => {
      this.album = this.albumService.getAlbum(params['albumId']);
    });
  }

  ngOnDestroy() {
    this.paramsSubscription.unsubscribe();
  }

}
```

5- Les gardes de route :

Pour contrôler si l'utilisateur peut naviguer dans les différents liens du site, on va utiliser les gardes de route « Guards ».

On va créer une nouvelle classe dans le fichier « logged.guard.ts » qui implémente « CanActivate »:

```
import { Injectable } from '@angular/core';
import { CanActivate, Router,
         ActivatedRouteSnapshot, RouterStateSnapshot } from '@angular/router';
import { LoginService } from './login.service';

@Injectable()
export class LoggedInGuard implements CanActivate {

  constructor(private router: Router,
              private loginService: LoginService) { }

  canActivate(route: ActivatedRouteSnapshot, state: RouterStateSnapshot) {
    if (!this.loginService.loggedIn) {
      this.router.navigate(['/login']);
      return false;
    }
    return true;
  }

}
```

Importer le nouveau fichier dans le module :

```
import { LoggedInGuard } from './login/logged.guard';
```

Et aussi dans le NgModule :

```
@NgModule({
  imports: [BrowserModule, FormsModule,routing],
  declarations: [
    AppComponent,LoginFormComponent,ArtistListComponent,
    ArtistDetailComponent,AlbumListComponent,AlbumDetailComponent
  ],
  providers: [
    ArtistService,AlbumService,LoginService,
     {provide: LocationStrategy, useClass: HashLocationStrategy},
    LoggedInGuard
  ],
  bootstrap: [AppComponent]
})
```

Dans notre route config, on va ajouter le nouveau garde pour les deux règles de routage « ArtistDetailComponent » et « AlbumDetailComponent »:

```
import { RouterModule } from '@angular/router';
import { LoginFormComponent } from './login/login-form.component';
import { ArtistListComponent } from './artist/artist-list.component';
import { ArtistDetailComponent } from './artist/artist-detail.component';
import { AlbumListComponent } from './album/album-list.component';
import { AlbumDetailComponent } from './album/album-detail.component';
import { LoggedInGuard } from './login/logged.guard';
export const routing = RouterModule.forRoot([
    {
        path: 'login',component: LoginFormComponent
    },
    {
        path: 'artists',component: ArtistListComponent
    },
    {
        path: 'artists/:artistId',component: ArtistDetailComponent,
        canActivate: [LoggedInGuard]
    },
    {
        path: 'albums',component: AlbumListComponent
    },
    {
        path: 'albums/:albumId',component: AlbumDetailComponent   ,
        canActivate: [LoggedInGuard]
    },
    {
        path: '',redirectTo: '/artists', pathMatch: 'full'
    }
]);
```

Maintenant, vous n'êtes pas autorisé à consulter le détail de l'artiste ou de l'album si vous n'êtes pas authentifié. Vous serez redirigé automatiquement vers la page d'authentification.