

5 - Ajouter la navigation dans l'application avec le routage

Etape 1

1 - Dans le dossier src/app créer le fichier app-routing.module.ts et ajoutez-y le code suivant :

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { HeroesComponent } from '../heroes/heroes.component';

const routes: Routes = [
  { path: 'heroes', component: HeroesComponent }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})

export class AppRoutingModule { }
```

2 - Ouvrez le modèle AppComponent (app.component.html) et remplacez l'élément <app-heroes> par un élément <router-outlet>

3 - Ajouter un lien de navigation (routerLink) qui vers le composant heroe

4 Nous allons ajouter une vue de tableau de bord. pour cela créez un nouveau composant nommé "dashboard" et remplacer le contenu des fichiers par le code suivant:

src / app / dashboard / dashboard.component.html

```
<h3>Top Heroes</h3>
<div class="grid grid-pad">
  <a *ngFor="let hero of heroes" class="col-1-4">
    <div class="module hero">
      <h4>{{hero.name}}</h4>
    </div>
  </a>
</div>
```

src / app / dashboard / dashboard.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Hero } from '../hero';
import { HeroService } from '../hero.service';

@Component({
  selector: 'app-dashboard',
  templateUrl: './dashboard.component.html',
  styleUrls: [ './dashboard.component.css' ]
})

export class DashboardComponent implements OnInit {
  heroes: any;

  constructor(private heroService: HeroService) {
    this.heroes = this.heroService.heroes.slice(1, 5)
  }

  ngOnInit() {}
}
```

src / app / dashboard / dashboard.component.css

```
/* DashboardComponent's private CSS styles */

[class*='col-'] {
  float: left;
  padding-right: 20px;
  padding-bottom: 20px;
}

[class*='col-']:last-of-type {
  padding-right: 0;
}
```

```

a {
  text-decoration: none;
}
*, *:after, *:before {
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
  box-sizing: border-box;
}
h3 {
  text-align: center;
  margin-bottom: 0;
}
h4 {
  position: relative;
}
.grid {
  margin: 0;
}
.col-1-4 {
  width: 25%;
}
.module {
  padding: 20px;
  text-align: center;
  color: #eee;
  max-height: 120px;
  min-width: 120px;
  background-color: #3f525c;
  border-radius: 2px;
}
.module:hover {
  background-color: #eee;
  cursor: pointer;
  color: #607d8b;
}
.grid-pad {
  padding: 10px 0;
}
.grid-pad > [class*='col-']:last-of-type {
  padding-right: 20px;
}
@media (max-width: 600px) {
  .module {
    font-size: 10px;
    max-height: 75px; }
}
@media (max-width: 1024px) {
  .grid {
    margin: 0;
  }
  .module {
    min-width: 60px;
  }
}

```

5 - Ajouter Dashboard à la barre de navigation

6 - Ajouter un itinéraire par défaut pour que l'application accède automatiquement au tableau de bord à son démarrage

Etape 2

1 - Créer une navigation vers les détails du héros

L'utilisateur doit pouvoir accéder à ces détails de trois manières.

- En cliquant sur un héros dans le tableau de bord.
- En cliquant sur un héros dans la liste des héros.
- En collant une URL "lien profond" dans la barre d'adresse du navigateur qui identifie le héros à afficher.

2 - Vous ajouterez un bouton "Back" sur la page des détails-héros qui permet de revenir à la vue d'avant

Résultat :

