

Övningsuppgift 11

I den här övningsuppgiften skall vi lära oss mer om navigering (routing) i Angular. Vi kommer att skapa flera olika "Components" och navigera mellan dem - jag rekommenderar därför att arbetet sker i en ny Angular Applikation.

Du kommer att skapa en applikation som kan navigera mellan fyra komponenter samt att den har en navigations (meny) komponent. En av komponenterna kommer dessutom att skicka vidare information till en annan komponent med hjälp av routern.

Innehållet i de olika komponenterna kommer att vara begränsat eftersom övningen går ut på att bli mer van vid att arbeta med routing.

1. Skapa en ny applikation - när du skapar välja att den skall ha "Angular Routing".
 1. Kontrollera i index.html att den har en "base href"
 2. Undersök filen app-routing.module.ts -- hur fungerar den? Var kommer du att skriva dina routes?
 3. Undersök filen app.module.ts — hur är routing tillagt till din applikation?
 4. Du har nu tittat på allt som alternativet "Angular Routing" skapar när appen skapades. Detta kan skapas manuellt om du vill lägga till Routing till en befintlig applikation.
2. Nu skall vi lägga till bootstrap till applikationen
 1. I mappen som skapades, kör

```
ng add @ng-bootstrap/ng-bootstrap
```

```
npm install bootstrap-icons --save
```
 2. Om du redan har startat "ng serve" behöver du stänga av det och starta det på nytt.
 3. Din applikation har nu tillgång till Bootstrap och du kan style dina komponenter.
3. Dags att skapa lite komponenter. Använd "ng g c" för att skapa dem - om du vill använda "inline" template och style kan du använda -t -s flaggorna till "ng"-kommandot.
 1. Skapa "Home" - kommer att vara din "hemskärm"
 2. Skapa "About" - kommer att vara din sida där du kan skriva text om appen / projektet.
 3. Skapa "NotFound" - en sidan som hanterar icke befintliga routes.
 4. Skapa "Navigation" - en komponent som innehåller din meny.
 5. Skapa "Rating" - En komponent vi använder för att visa rating (här kommer vi att experimentera med route parametrar).
4. Uppdatera din "app-root" komponent - låt den skriva ut din "Navigation"-komponent samt din "router-outlet". Nu kommer din applikation alltid visa menyn över ditt innehåll.
5. Dags att skapa routes. I din app-routing.module.ts - uppdatera din "routes"-konstant.
 1. Innehålla en tom route som kopplar oss till "Home"-komponenten
 2. Innehålla en "about" route som kopplar oss till "About"-komponenten.

3. Innehålla en default route (**) som kopplar oss till "NotFound"-komponenten.
4. I din app skall du nu se innehållet från din "Navigations"-komponenten över innehållet från din "Home"-komponent.
5. Om du ändrar din adress - lägg till t.ex. /angular - då skall du nu se "Navigations"-komponenten över din "NotFound"-komponent.
6. Dags att skapa en meny. Gå till din "Navigation"-komponent. Skapa en meny. Du kan lägga lite eller mycket tid på style. Upp till dig! Det viktiga är att du kan navigera mellan Home- och About-komponenten.
 1. Skapa en länk (a-tag) med en tom "routerLink" - när du testar skall denna öppna din "Home"-komponent.
 2. Skapa en länk (a-tag) sätt "routerLink" så att den öppnar din "About"-komponent.
7. Nu skall vi navigera utan att använda "taggar" (imperativ navigation). I din "About"-komponent:
 1. Skapa en knapp - sätt texten på knappen till "Home" - koppla (click) på knappen till en metod i komponentens klass.
 2. Använd "dependency injection" för att ge komponenten tillgång till routern.
 3. Använd "navigate"-metoden på routern för att öppna "Home" när knappen trycks ner.
8. Nu skall vi testa att skicka parametrar med hjälp av routern. Gå till app-routing.module.ts filen. Skapa en route för "rating/:rate" som tar oss till Rating-komponenten.
9. Uppdatera din "Home"-komponent. Den skall ha en lista med taggar (Rate 1 — 5) - de skall ta användaren till "Rating" och skicka med det motsvarande talet som en parameter. Du kan göra detta genom att skapa 5 taggar med olika routerLink-attribut. Om du trycker på de olika länkarna nu kommer du att se att motsvarande värde syns i adressfältet på webbläsaren.
10. I "Rating"-komponenten skall vi titta på värdet och skriva ut rätt antal stjärnor beroende på hur man har valt att rösta (vilken länk man tryckte på). Dvs visa en

```
<svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-star-fill" viewBox="0 0 16 16"><path d="M3.612 15.443c-.386.198-.824-.149-.746-.592l.83-4.73L173 6.765c-.329-.314-.158-.888.283-.951l4.898-.696L753.792c.197-.39.73-.39.927 0l2.184 4.327 4.898.696c.441.062.612.636.282.951l-3.522 3.356.83 4.73c.078.443-.367.746-.592L13.187l-4.389 2.256z"/></svg>
```

för varje tal.

1. Börja med att skapa ett nytt klass-attribut - "rating" - detta är ett tal.
2. Skapa en "ActivatedRoute" med hjälp av dependency injection.
3. Importera "OnInit" se till att din komponent-klass implementerar den samt skapa metoden ngOnInit.
4. I ngOnInit - skapa en "subscription" för params på din ActivatedRoute. Hämta "rate" parametern och stoppa in den i ditt "rating" klass-attribut.
5. Skapa "markup" för att skriva ut rätt antal "stjärnor". Du kan för enkelhets skull också skriva ut ditt rating klass-attribut för att se att du får värdet från routern.
6. Du skall nu se olika många stjärnor beroende på vilken länk du trycker på i Home-komponenten.

11. Gå tillbaka "About"-komponenten. Skapa en ny knapp som navigerar dig till "Rating"-komponenten och ger dig 5-stjärnor.(Imperativ)