15.1 Layout

Lav et nyt MVC projekt.

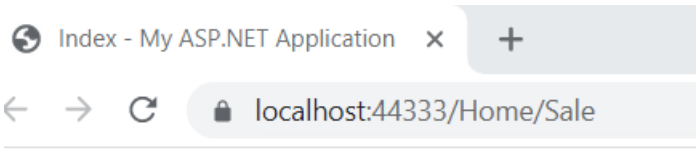
Lav en ny controller. Kald den bare HomeController.

Lav en action der hedder Sale og et tilhørende View. Lad View'et anvende default layout (dvs. check i Use a layout page, men intet layout angivet).

Kør programmet og se, at det virker. Hvor kommer al teksten fra?

Tilret _Layout-filen (under Views/Shared) og se, at ændringerne slår igennem. Du kan bare rette noget af teksten.

Kig på fanen i browseren. Der står muligvis Index - My ASP.NET Application:



Eller der kan stå Sale - My ASP.NET Application

Hvor kommer dette fra? _Layout eller dit view? Eller begge steder?

Tilret i view, så der står noget andet end Index (eller sale)

Lav tilføjelse til Sale-view'et: lav en section "SaleSection". Der skal bare stå en tekst f.eks.

```
<div>
        Denne tekst forklarer, hvordan Sale fungerer
</div>
```

Brug denne (RenderSection) i _Layout før RenderBody.

Kør det og se, at denne section kommer med som det første, uanset hvor du har placeret den i Sale-view'et.

Lav en ny Action (MoreSales) i Home og et tilhørende view. Lav også i dette view en section SaleSection med en anden tekst. Kør denne og se, at teksten kommer i resultatet.

Lav en ny Action Buy i Home og et tilhørende view. I _Layout laves en RenderSection på BuySection og i Buy-view'et laves denne section.

Prøv at køre alle dine views. Hvis du har lavet dine RenderSection med en Required på true vil de fejle. Hvorfor? Løs dette.

Efter løsningen skal dine sale-views gerne vise deres sale-section mens buy-view'et skal vise deres buy-section.

Lav en ny Action BuyAndSell i Home og tilhørende view. I dette view laves både en SaleSection og en BuySection. Kør dette view. Nu skulle begge sections gerne vises. Det kan godt være, du skal lave lidt plads mellem dem. Det kan ske i _Layout eller i dit view.

Til sidst: lav om i _Layout, så dine sections vises efter body. Kør alle views og se, at ændringen slår igennem i alle views.

15.2 Partial view og child-action

Lav et nyt MVC projekt.

Lav en simpel model med en Person klasse (eller noget andet).

Lav en controller (bare Home).

Lav en action VisAlle, der laver en hardcoded liste af Personer og sender denne med til et view. Du kan f.eks. lave en metode private List<Person> AllePersoner(), der returnerer den. Vi får brug for den senere.

Lav et view til denne action, med model en liste af Personer, og en løkke, der udskriver personoplysningerne.
Hint: det kan drille lidt med at konkatenere strings i Razor. Prøv dette:

```
@(p.Fornavn + " " + p.Efternavn)
```

Eller lav noget mere fancy HTML.

Lav en ny action VisEn i samme controller, der blot viser et view, der modtager en enkelt Person.

Send f.eks. en af personerne i AllePersoner med.

Dette view skal have Person som model og blot vise den ene person på samme måde som i VisAlle.

Test begge views.

Hvad er problemet med denne løsning? Hvor mange steder skal rettes, hvis personer skal vises på en anden måde (f.eks. en ekstra property)?

Lav nu et partial view VisEnPerson, der indeholder kode til at vise en person (og Person som model). Du kan bare copy-paste fra det du har lavet. Husk fluebenet, når du laver view'et.

Erstat den eksisterende kode, både i VisAlle og i VisEn, så den inkluderer dette view i stedet vha. Partial eller RenderPartial. Husk at sende modellen med (en person).

Nu skal vi anvende lidt mere kompliceret kode til at vise personer. Man skal forestille sig en visning, der kræver avanceret logik. For at undgå at kode for meget i Razor laver vi en Child action:

Lav en ny action i controlleren, kald den VisEnPerson. Den skal markeres med ChildActionOnly.

Denne action skal have en parameter af type Person.

Controlleren skal først modificere personen. Det kan være noget med at ændre en dato eller vælge mellem to forskellige personer ud fra et eller andet kriterie eller bytte om på fornavn og efternavn. Eller måske skal personen anonymiseres hvis vedkommende er under 18 år. Derefter skal den vise view'et VisEnPerson med personen som model. Skal dette gøres med View() eller PartialView() ?

I views VisEn og VisAlle skal nu i stedet kaldes denne action (Html.Action eller Html.RenderAction)

15.3 Gennemgående opgave.

Fortsæt på din gennemgående opgave. Hvis du ikke har en kørende tre lags arkitektur, hvor du kan hente fra databasen, skal du have det til at fungere først.

Find steder i din applikation, hvor du kan anvende

- Layout, burde ikke være så svært. Lav et fornuftigt layout. Du kan evt. tage udgangspunkt i default layout og så rette i det.
- Partial views, det kan godt være svært, hvis du kun har et enkelt view. Så kan du evt. overveje, hvor og hvad det kunne være.
- Child actions, det er nok endnu sværere end partial views, af samme grund (du er ikke kommet ret langt endnu)
- Evt. lav et nyt area.

For ovenstående gælder, at hvis din applikation ikke er avanceret nok til at anvende dem endnu, så bygger du videre på din applikation med henblik på at kunne anvende dem senere.