

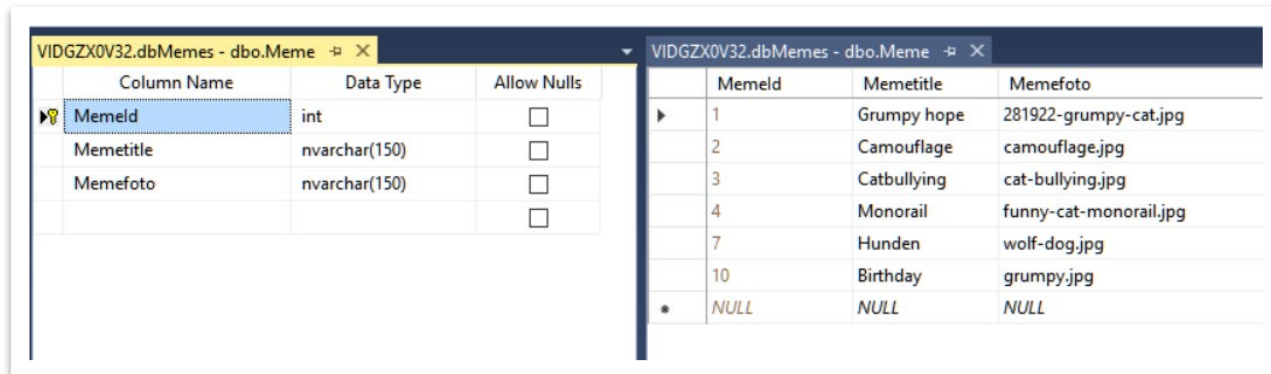
Memes

Opret, slet og ret - simpel database og webprojekt med image

Projektet her er et mini-projekt, hvor der er et image med. Det giver et simpelt overblik over, hvordan man kan håndtere billeder ved opret, slet og ret data (her er det memes) og et image.

Det særlige ved et billede er, at du udover at skulle **gemme navnet (= en stump tekst) i database-tabellen** (med mulighed for at rette i det og slette det) OGSÅ skal **uploade selve filen (jpg, gif, png osv.)** til serveren - til din image-mappe.

Databasen



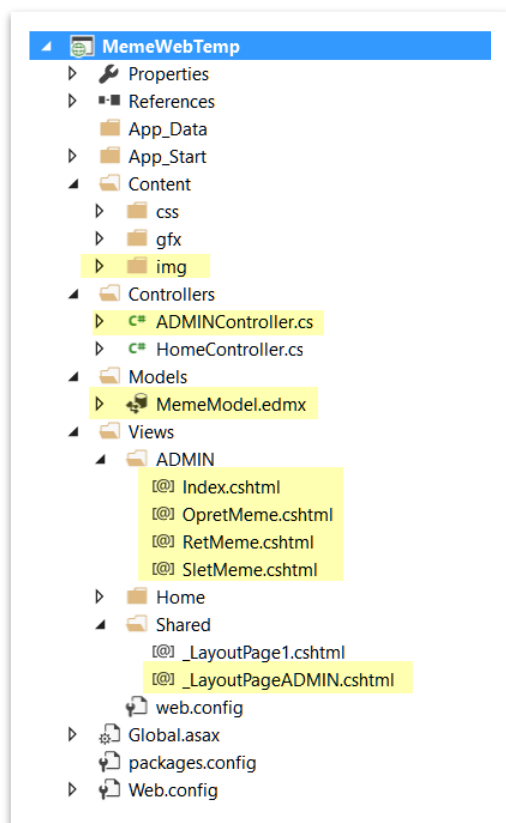
The screenshot shows two panes of SQL Server Enterprise Manager. The left pane displays the database schema for 'VIDGZX0V32.dbMemes - dbo.Meme'. The right pane displays the data for the same table.

Column Name	Data Type	Allow Nulls
Memeld	int	<input type="checkbox"/>
Memetitle	nvarchar(150)	<input type="checkbox"/>
Memefoto	nvarchar(150)	<input type="checkbox"/>

Memeld	Memetitle	Memefoto
1	Grumpy hope	281922-grumpy-cat.jpg
2	Camouflage	camouflage.jpg
3	Catbullying	cat-bullying.jpg
4	Monorail	funny-cat-monorail.jpg
7	Hunden	wolf-dog.jpg
10	Birthday	grumpy.jpg
• NULL	NULL	NULL

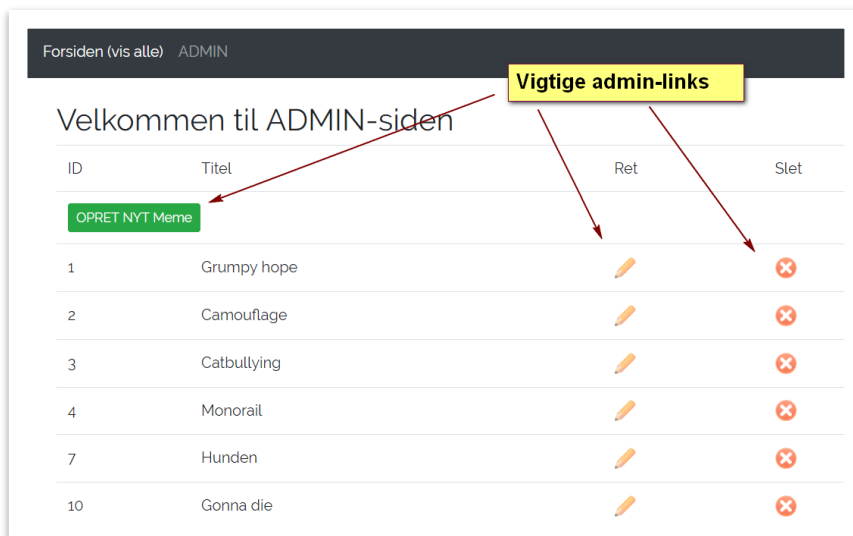
Webprojektet

Mapper og vigtige filer/klasser:



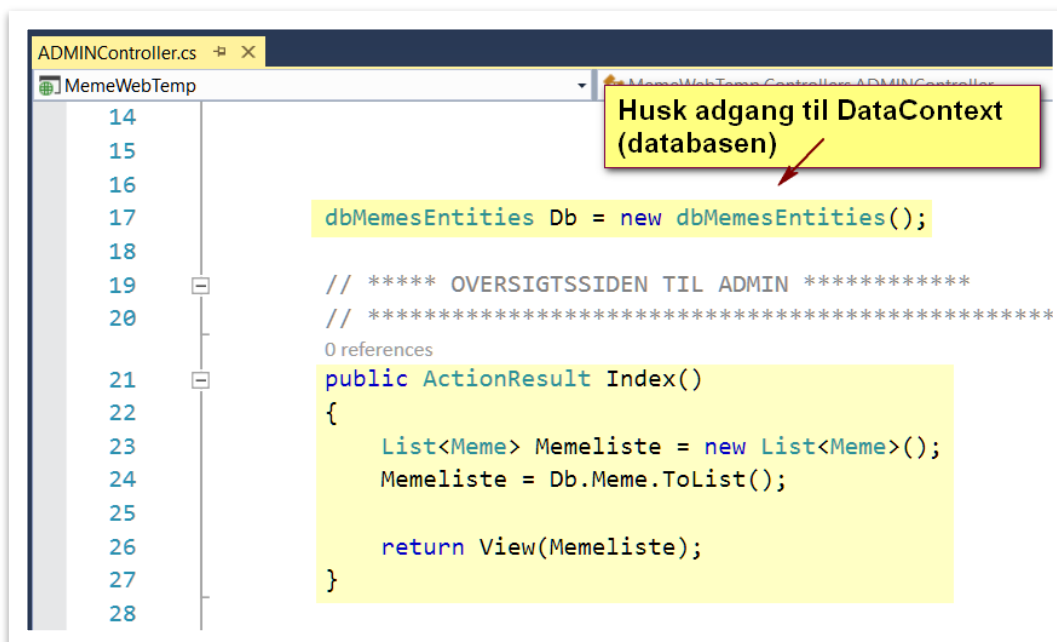
Startsiden til opret, slet og ret

Det kan håndteres på et hav af måder, men denne her startside til opret, slet og ret for fx Memes fungerer fint:



Action-metoden til oversigten - i ADMINController:

... behøver ikke hedde Index, men det er et godt navn til en startside 😊



Viewet (Index.cshtml – i ADMIN-mappen)

Ignorer evt. de mange "class" – de kommer fra Bootstrap og er kun "pynt". Og brug gerne div og grid i stedet for tables ...

```
@model List<MemewebTemp.Models.Meme>
```

```
@{
```

```
    ViewBag.Title = "Index";  
    Layout = "~/Views/Shared/_LayoutPageADMIN.cshtml";
```

```
    if (TempData["Besked"] != null)  
    {  
        ViewBag.Besked = TempData["Besked"];  
    }  
}
```

```
<h2>Velkommen til ADMIN-siden</h2>
```

```
<table class="table">
```

```
    <tr>
```

```
        <td>ID</td>  
        <td>Titel</td>  
        <td>Ret</td>  
        <td>Slet</td>  
    </tr>
```

```
    <tr>
```

```
        <td colspan="2">  
            <a href="/Admin/OpretMeme" class="btn btn-success btn-sm">OPRET NYT Meme</a>  
        </td>  
        <td colspan="2">  
            <span class="text-danger display-4">@ViewBag.Besked</span>  
        </td>  
    </tr>
```

```
    @foreach (var item in Model)
```

```
    {
```

```
        <tr>
```

```
            <td>@item.MemeId</td>  
            <td>@item.Memetitle</td>
```

```
            <td>
```

```
                <a href="/Admin/RetMeme/@item.MemeId">  
                      
                </a>  
            </td>
```

```
            <td>
```

```
                <a href="/Admin/SletMeme/@item.MemeId">  
                      
                </a>  
            </td>
```

```
        </tr>
```

```
    }
```

```
</table>
```

Opret ny meme

Opret, ret og slet har alle brug for 2 action-metoder: En til det, der skal vises, når siden loader. Og en til at håndtere det "post" som kommer fra formularen, når der klikkes på submit-button ... den med [HttpPost]

Action-metoderne til OPRET - i ADMINController

```
// ***** OPRET NY *****  
// *****  
0 references  
public ActionResult OpretMeme()  
{  
    Meme NyMeme = new Meme();  
    return View(NyMeme);  
}
```

```
[HttpPost]  
0 references  
public ActionResult OpretMeme(Meme NyMeme, HttpPostedFileBase Memefoto)  
{  
    if (!ModelState.IsValid || Memefoto == null)  
    {  
        ViewBag.Besked = "Noget gik galt - prøv igen";  
        return View(NyMeme); // Retur til formularen - send det allerede-udfyldte med retur  
    }  
  
    // Formularen er korrekt udfyldt - data gemmes i DB og fil uploades i korrekt mappe i webprojekt  
  
    // Håndter filen (snup filnavn til DB + gem filen i img-mappe)  
    string imgnavn = System.IO.Path.GetFileName(Memefoto.FileName);  
    string imgsti = System.IO.Path.Combine(Server.MapPath("~/Content/img/"), imgnavn);  
    // Gem filen:  
    Memefoto.SaveAs(imgsti);  
  
    // Gem i DB  
    NyMeme.Memefoto = imgnavn; // Fx hest.jpg  
  
    Db.Meme.Add(NyMeme);  
    Db.SaveChanges();  
  
    TempData["Besked"] = "Meme er oprettet!!!";  
  
    return RedirectToAction("Index", "ADMIN");  
}
```

Viewet (OpretMeme.cshtml i ADMIN-mappen)

```
@model MemewebTemp.Models.Meme

@{
    ViewBag.Title = "OpretMeme";
    Layout = "~/Views/Shared/_LayoutPageADMIN.cshtml";
}

<h1>Opret ny meme</h1>

<form action="~/ADMIN/OpretMeme" method="post" enctype="multipart/form-data">

    <div class="text-danger">@Html.ValidationSummary()</div>

    @* *** Meme-Overskrift *** *@
    <div class="form-group">
        <label for="Memetitle">Overskrift:</label>
        <input type="text" name="Memetitle" id="Memetitle" value="@Model.Memetitle" placeholder="Skriv en titel" class="form-control" />
        <span class="text-danger">@Html.ValidationMessageFor(m => m.Memetitle)</span>
    </div>

    @* *** Meme-foto *** *@
    <div class="form-group">
        <label for="Memefoto">Vælg et foto</label>
        <input type="file" name="Memefoto" id="Memefoto" class="form-control" />
        <span class="text-danger">@Html.ValidationMessageFor(m => m.Memefoto)</span>
    </div>

    @* *** SUBMIT *** *@
    <button type="submit" class="btn btn-primary">OPRET NY</button>

</form>
```

Slet meme

Opret, ret og slet har alle brug for **2 action-metoder**: En til det, der skal vises, når siden loader. Og en til at håndtere det "post" som kommer fra formularen, når der klikkes på submit-button ... den med [HttpPost]

Action-metoderne til SLET - i ADMINController

```
// ***** SLET *****  
// *****  
0 references  
public ActionResult SletMeme(int? Id)  
{  
    // Hvis der ikke er en ID med, så spark tilbage til index-siden  
    if (Id == null)  
        return RedirectToAction("Index");  
  
    // Hent meme som måske skal slettes - og send det med til viewet  
    Meme MemeSlettes = Db.Meme.Find(Id);  
  
    if (MemeSlettes == null) // Hvis der ikke er et meme som matcher id  
        return RedirectToAction("Index");  
  
    return View(MemeSlettes);  
}
```

```
[HttpPost]  
0 references  
public ActionResult SletMeme(Meme MemeSlettes)  
{  
    // Hent meme som måske skal slettes - og send det med til viewet  
    // Meme har kun ID med - så resten skal lige slås op med "Find"  
    MemeSlettes = Db.Meme.Find(MemeSlettes.MemeId);  
  
    if (MemeSlettes == null) // Hvis der ikke er et meme som matcher id  
        return RedirectToAction("Index");  
  
    // Slet fysisk fil/image  
    string imgsti = System.IO.Path.Combine(Server.MapPath("~/Content/img/"), MemeSlettes.Memefoto);  
  
    // HVIS der er et image på den angivne sti  
    if (System.IO.File.Exists(imgsti))  
        System.IO.File.Delete(imgsti); // Filen slettes  
  
    // Slet fra DB  
    Db.Meme.Remove(MemeSlettes);  
    Db.SaveChanges();  
  
    TempData["Besked"] = "Meme er slettet";  
  
    return RedirectToAction("Index", "ADMIN");  
}
```

Viewet (SletMeme.cshtml i ADMIN-mappen)

```
@model MemeWebTemp.Models.Meme

@{
    ViewBag.Title = "SletMeme";
    Layout = "~/Views/Shared/_LayoutPageADMIN.cshtml";
}

<h1>ADVARSEL - SLETNING</h1>

<h4>Er du sikker på, at du vil slette følgende meme: </h4>

<div class="card my-5">
    <div class="card-body">
        <h4>@Model.Memetitle</h4>
    </div>
</div>

<form action="~/ADMIN/SletMeme" method="post">
    <input type="hidden" name="MemeId" id="MemeId" value="@Model.MemeId" />
    <button type="submit" class="btn btn-danger">SLET</button>
    <a href="~/ADMIN/" class="btn btn-success">Fortryd (uden at slette noget)</a>
</form>
```

Ret (rediger/update) meme

Opret, ret og slet har alle brug for 2 action-metoder: En til det, der skal vises, når siden loader. Og en til at håndtere det "post" som kommer fra formularen, når der klikkes på submit-button ... den med [HttpPost]

Action-metoderne til RET – i ADMINController

```
// ***** RET *****
// *****
0 references
public ActionResult RetMeme(int? Id)
{
    // Hvis der ikke er en ID med, så spark tilbage til index-siden
    if (Id == null)
        return RedirectToAction("Index");

    // Hent meme som måske skal slettes - og send det med til viewet
    Meme MemeRettes = Db.Meme.Find(Id);

    if (MemeRettes == null) // Hvis der ikke er et meme som matcher id
        return RedirectToAction("Index");

    return View(MemeRettes);
}
```


[HttpPost]

0 references

public ActionResult RetMeme(Meme MemeRettes, HttpPostedFileBase NytMemefoto)

{

 // Hvis det ikke er udfyldt korrekt

 if (!ModelState.IsValid)

 {

 ViewBag.Besked = "Noget gik galt - prøv igen";

 return View(MemeRettes); // Retur til formularen - send det allerede-udfyldte med retur

 }

 // HVIS der er valgt et nyt image

 // = slet den gamle imagefil + upload det nye image til image-mappen + nyt imgnavn i DB

 if (NytMemefoto != null)

 {

 // Find image som skal slettes

 string imgsti = System.IO.Path.Combine(Server.MapPath("~/Content/img/"), MemeRettes.Memefoto);

 // HVIS der er et image på den angivne sti

 if (System.IO.File.Exists(imgsti))

 System.IO.File.Delete(imgsti); // Filen slettes

 // Gem det nye/rettede foto

 string imgnavn = System.IO.Path.GetFileName(NytMemefoto.FileName);

 imgsti = System.IO.Path.Combine(Server.MapPath("~/Content/img/"), imgnavn);

 NytMemefoto.SaveAs(imgsti); // Gemmer selve filen

 MemeRettes.Memefoto = imgnavn; // Gemmer navnet i modellen - som i næste step bliver sendt til DB

 }

 // Gem i DB

 Db.Meme.AddOrUpdate(MemeRettes);

 Db.SaveChanges();

 TempData["Besked"] = "Meme er rettet";

 return RedirectToAction("Index", "ADMIN");

}

Viewet (RetMeme.cshtml i ADMIN-mappen)

```
@model MemWebTemp.Models.Meme

@{
    ViewBag.Title = "RetMeme";
    Layout = "~/Views/Shared/_LayoutPageADMIN.cshtml";
}

<h1>Ret meme</h1>

<form action="~/ADMIN/RetMeme" method="post" enctype="multipart/form-data">

    <div class="text-danger">@Html.ValidationSummary()</div>

    @* Citat-ID *@
    <div class="form-group">
        <label for="MemeId">ID:</label>
        <input type="text" name="MemeId" id="MemeId" value="@Model.MemeId" readonly class="form-control" />
    </div>

    @* *** Meme-Overskrift *** *@
    <div class="form-group">
        <label for="Memetitle">Overskrift:</label>
        <input type="text" name="Memetitle" id="Memetitle" value="@Model.Memetitle" placeholder="Skriv en titel" class="form-control" />
        <span class="text-danger">@Html.ValidationMessageFor(m => m.Memetitle)</span>
    </div>

    @* *** Meme-foto *** *@
    <div class="form-group">

        Nuværende billede:
        

        @* Hvis brugeren vil beholde det eksisterende billede *@
        <input type="hidden" id="Memefoto" name="Memefoto" value="@Model.Memefoto" />

        @* Hvis brugeren vil have et nyt foto *@
        <label for="NytMemefoto">Vælg evt. et nyt foto</label>
        <input type="file" name="NytMemefoto" id="NytMemefoto" class="form-control" />

    </div>

    @* SUBMIT *@
    <button type="submit" class="btn btn-warning">GEM RETTELSE</button>
    <a href="~/ADMIN/" class="btn btn-success">Fortryd (uden at slette noget)</a>

</form>
```