Dagens program

Valideringsteknikker

- Explicit Model Validation
- Visning af Validation Error Messages

- Property Validation Attributes
- Model Validation Attributes
- Regular Expressions

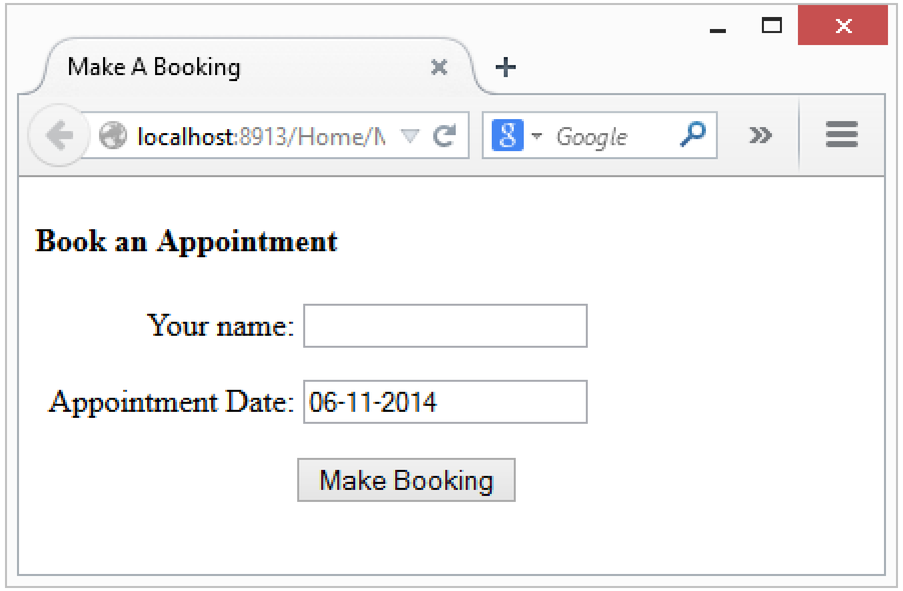
Avancerede emner

- Custom Property Validation Attribute
- Self-Validating Models
- Brug af Client Side Validation

Opgaver

Explicit Model Validation

Explicit Model Validation



Book an Appointment

Your name:

Appointment Date: 06-11-2014

Strongly typed **`Appointment`** view

```
public class Appointment {
        public string FullName { get; set; }

        [DataType(DataType.Date)]
        public DateTime Date { get; set; }
}
```

`Web.config`

```
<system.web>
    <globalization culture="da-DK" uiCulture="da-DK"/>
</system.web>
```

Sproget har en betydning for om default error beskeder kommer på engelsk (standard) eller dansk.

Explicit Model Validation (i Action metoden)
Fordele og ulemper?

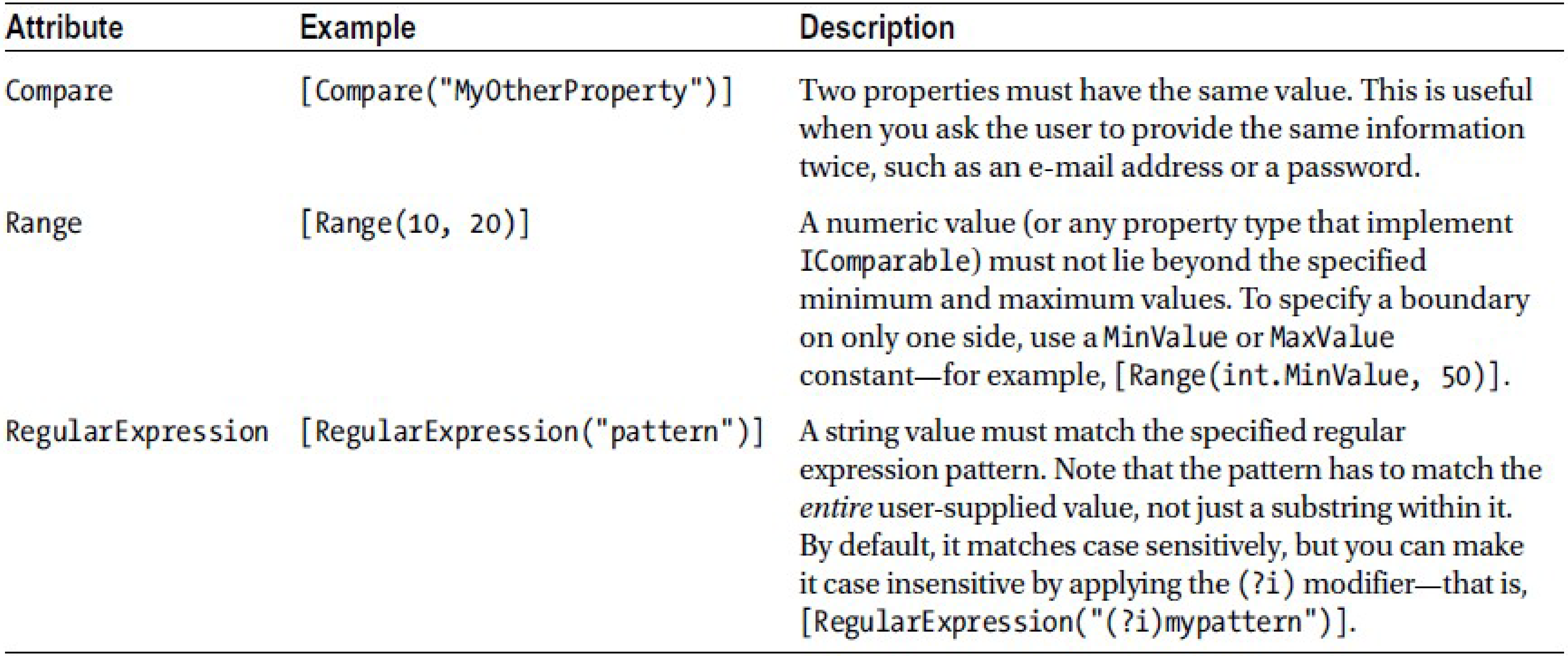
```
[HttpPost]
public ViewResult MakeBooking(Appointment appt) {
    if (string.IsNullOrEmpty(appt.FullName)) {
        ModelState.AddModelError("FullName", "Please enter your name"); // name of prop.
    }
    if (ModelState.IsValidField("Date") && DateTime.Now > appt.Date) {
        ModelState.AddModelError("Date", "Please enter a date in the future");
    }
    if (ModelState.IsValid) {
        // store new Appointment in repository
        return View("Completed", appt);
    }
    else {
        return View();
    }
}
```

Property Validation Attributer

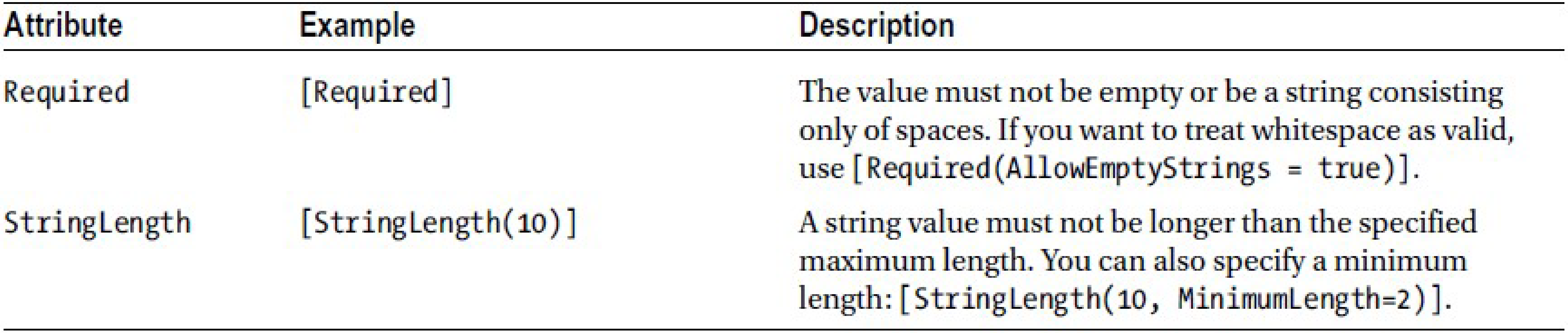
Fordel ved at bruge metadata til validering

- Validation attributter angivet i modellen bruges ALLE steder, hvor den model anvendes og ikke bare i en konkret Action Metode
- Giver også mening, at modellen specificerer valideringsreglerne samme sted som datatyperne defineres.
- Så kommer der en diskussion om, hvilken model, det skal stå i? EF-modellen, en DTO-model eller en viewmodel?

Udvalgte indbyggede Validation Attributer

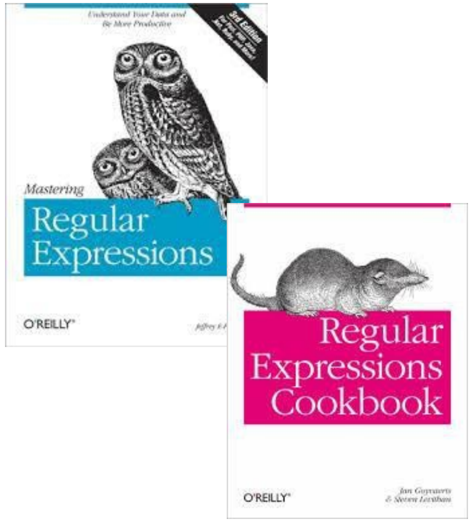


Attribute	Example	Description
`Compare`	`[Compare("MyOtherProperty")]`	Two properties must have the same value. This is useful when you ask the user to provide the same information twice, such as an e-mail address or a password.
`Range`	`[Range(10, 20)]`	A numeric value (or any property type that implement `IComparable`) must not lie beyond the specified minimum and maximum values. To specify a boundary on only one side, use a `MinValue` or `MaxValue` constant—for example, `[Range(int.MinValue, 50)]`.
`RegularExpression`	`[RegularExpression("pattern")]`	A string value must match the specified regular expression pattern. Note that the pattern has to match the *entire* user-supplied value, not just a substring within it. By default, it matches case sensitively, but you can make it case insensitive by applying the `(?i)` modifier—that is, `[RegularExpression("(?i)mypattern")]`.



Attribute	Example	Description
`Required`	`[Required]`	The value must not be empty or be a string consisting only of spaces. If you want to treat whitespace as valid, use `[Required(AllowEmptyStrings = true)]`.
`StringLength`	`[StringLength(10)]`	A string value must not be longer than the specified maximum length. You can also specify a minimum length: `[StringLength(10, MinimumLength=2)]`.

Regulære udtryk (engelsk: regular expressions)


Understand Your Data and Be More Productive
3rd Edition
Mastering
Regular Expressions
O'REILLY
Regular Expressions Cookbook
O'REILLY
Jan Goyvaerts & Steven Levithan

Validering med regular expressions

- Regulære udtryk
  - Bruges til mønster genkendelse
- Består af:
  - **Almindelige tegn** (literals), som repræsentere hvad de er – dvs. a betyder a og £ betyder £
  - **Meta tegn**, som har en anden betydning – f.eks. betyder . (punktum) ”any character” og **`\d`** betyder "any number".

Meta characters i regular expressions

Meta character defined		Exp	Maches
*	The asterisk is used to match 0 or more occurrences of the preceding character.	ab*c	ac, abc, abbc, abbbc, ...
+	1 or more of previous character og expression.	ab+c	abc, abbc, abbbc, ...
()	Logical grouping of part of an expression.	(32)+	32, 3232, 323232, 32323232; But not 32322 or 32323
{n,m}	Explicit quantifier notation	A{2,4}	AA, AAA, AAAA
|	Alternation. One of the two characters or sub expressions must match.	A|B	A, B
[]	Explicit set of characters to match.	[A-Ca-c]	A, B, C, a, b eller c.
[^]	Matches any single character not in the specified set of characters.	[^A-C]	All characters that are nor A, B or C

Flere meta characters

Meta character defined		Exp	Maches
.	Matches any character except \n (new line).	.at	hat, cat, rat, bat, fat, vat
\s	Matches any white-space character like space, tab, new line	ab\sc	ab c
\S	Matches any non-white-space character.		
\d	Matches any number		
\D	Matches any character wich are not numbers		
\w	Matches any word character like letter, number and underscore	\w	sand, Cars 2, see_you_soon
\W	Matches any non-word character	\W	£, $, ;. *, :, ?

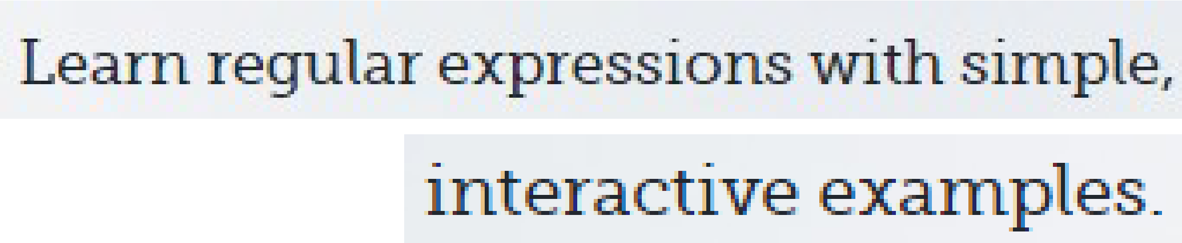
Eksempler på regulære udtryk

Field		
Email address	\S+@\S+\.\S{2,3}	Check for @ and . and allow only nonwhitespace characters
Password	\w+	Allow only word characters (letters, numbers and underscore)
Password	\w{6,12}	The same as previous but min. 6, max. 12 chatacters
Password	[a-zA-Z]\w*\d+\w*	Stats with a-z or A-Z contains word characters and one number at least
Phone	\+{0,1}[\d\s]{8,14}	Might start with +. 8 -14 numbers and whitespaces are allowed

Validering med regulære udtryk

```
[RegularExpression(@"\S+@(\S+\.)+\w{2,4}",
ErrorMessage = "There is a problem with the email")]
[Required(ErrorMessage = "You must enter email")]
public string Email { get; set; }
```

RegexOne



Learn regular expressions with simple, interactive examples.

http://regexone.com

Custom
Property Validation Attribute

Eksempel: FutureDateAttribute class

```
public class FutureDateAttribute : ValidationAttribute {
    public override bool IsValid(object value) {
        DateTime dt;
        return (DateTime.TryParse(value.ToString(), out dt) &&

                        dt > DateTime.Now);
    }
}
```

```
public class Appointment {
       ...
        [DataType(DataType.Date)]
        [FutureDate(ErrorMessage="Please enter a date in the future")]
        public DateTime Date { get; set; }
}
```

Custom
Model Validation Attribute

Vores Custom Model Validation Attribute

Læg den i `infrastructure` folder. Hvad tjekkes i koden her?

```
public class ManufacturerColorAttribute : ValidationAttribute {
     public ManufacturerColorAttribute() {
          ErrorMessage = "The Manufacturer does not support this color";
     }
     public override bool IsValid(object value) {
          CarRent carRent = value as CarRent;
          if (carRent == null || string.IsNullOrEmpty(carRent.Manufacturer)
                              || string.IsNullOrEmpty(carRent.Color)) {
               return false; // I don't have a model with the properties I
               require
          } else {
               if (carRent.Manufacturer == "Ford") {return carRent.Color == "Blue"; }
               else { return false; }
          }
     }
}
```

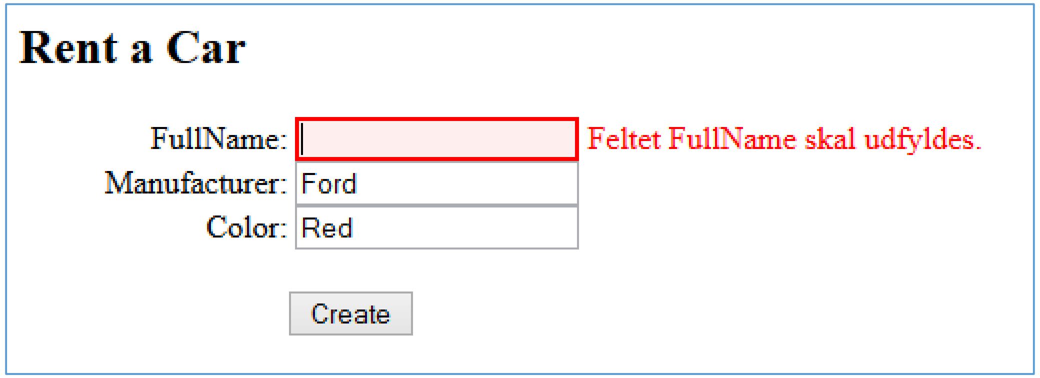
Brug af custom model validation attribute.

```
[ManufacturerColor]
public class CarRent {
    [Required]
    public string FullName{ get; set; }
    [Required]
    public string Manufacturer { get; set; }
    [Required]
    public string Color { get; set; }
}
```

UX Noter.

- Model level validation eksekveres EFTER attribute level validation

1.



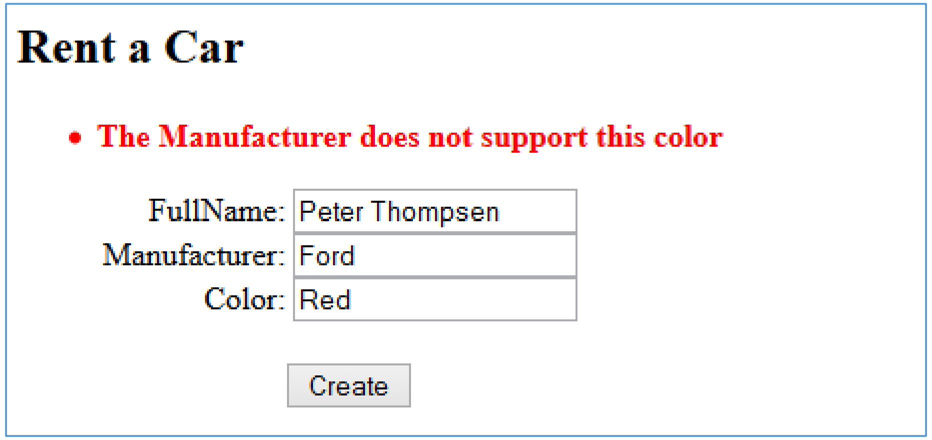
Rent a Car

FullName: Feltet FullName skal udfyldes.

Manufacturer: Ford

Color: Red

2.



Rent a Car

- The Manufacturer does not support this color

FullName: Peter Thompsen

Manufacturer: Ford

Color: Red

Client side vs server side validation

ModelState objekt

- **Gemmer valideringsresultater, incl en liste af fejl.**
- **Centralt er ModelState.IsValid**
- **Alle fejl:**

```
foreach(ModelState s in ModelState.Values)
  {
      foreach(var e in s.Errors)
      {
          Debug.Write(e.ErrorMessage);
      }
  }
```

Serverside validering, eksempel

```
public ActionResult AddEmployee(Employee employee)
       {
           if (employee == null)
           {
               return View("Employee");
           } else {
               if (ModelState.IsValid)
               {
                   EmployeeBLL employeeBLL = new EmployeeBLL();
                   employeeBLL.AddEmployee(employee);
                   return View("EmployeeAdded");
               } else
               {
                   return View("Employee");
               }
           }
       }
```

Client side validation

- **Kræver JavaScript**
- **Dette kan autogeneres af gængse validation attributes**
- **Kræver JQuery**

Client Side Validation

Validering FØR vi trykker på en submit knap - dvs. før Post server kald til controller.

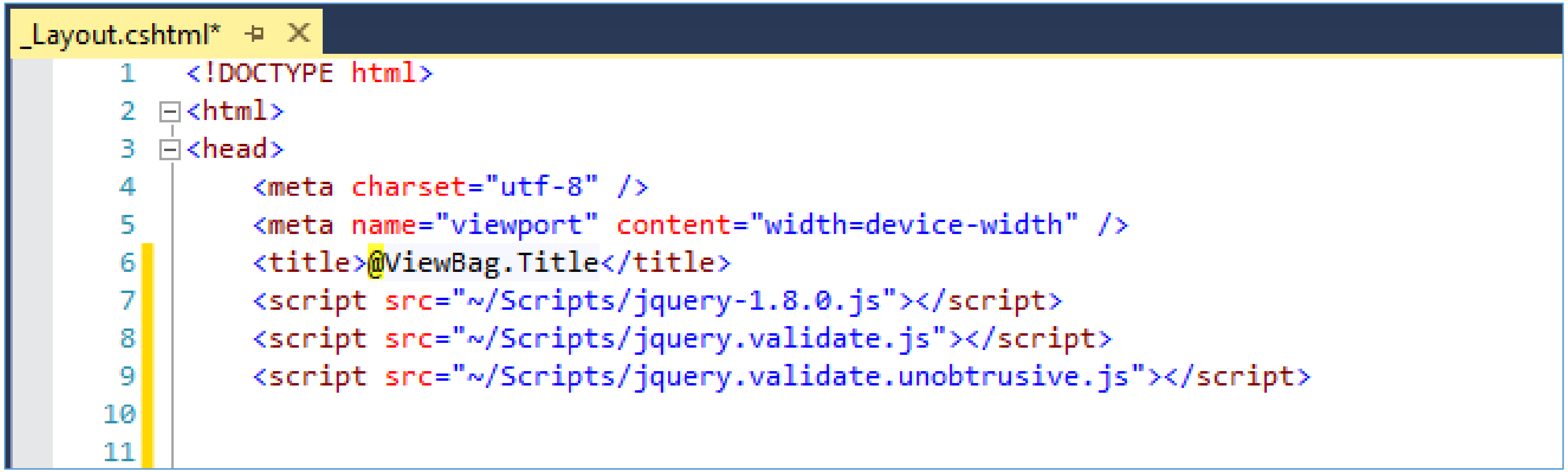
Install JQuery for at få det til at virke.

```
PM> Install-Package jQuery
PM> Install-Package jQuery.Validation
PM> Install-Package Microsoft.jQuery.Unobtrusive.Validation
```

Enable Client-Side validation i `web.config`

```
<appSettings>
  <add key="ClientValidationEnabled" value="true"/>
  <add key="UnobtrusiveJavaScriptEnabled" value="true"/>
</appSettings>
```

Inkluder referencer til JavaScript i master layout

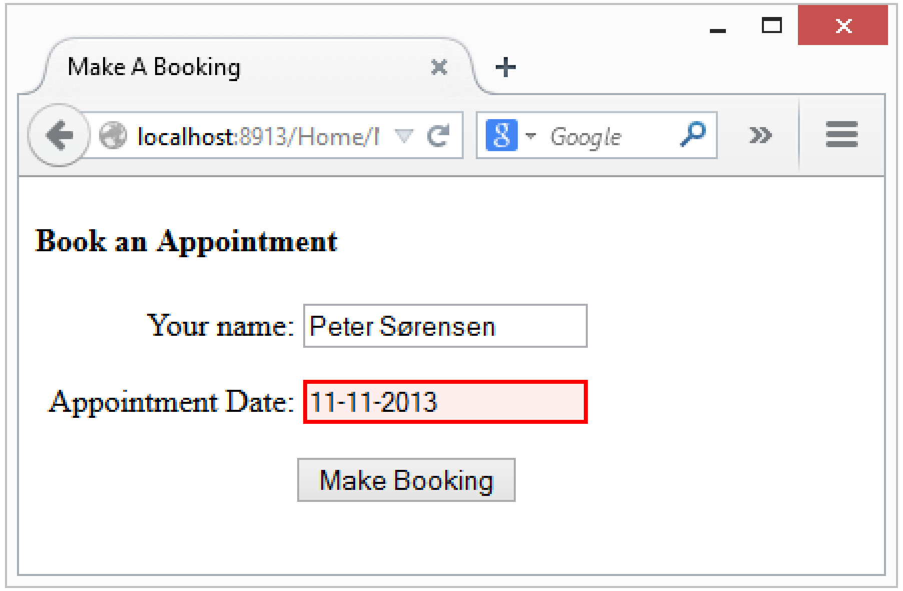


```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <title>@ViewBag.Title</title>
    <script src="~/Scripts/jquery-1.8.0.js"></script>
    <script src="~/Scripts/jquery.validate.js"></script>
    <script src="~/Scripts/jquery.validate.unobtrusive.js"></script>


```

Visning af Validation Error Messages

Visning af validation errors



Book an Appointment

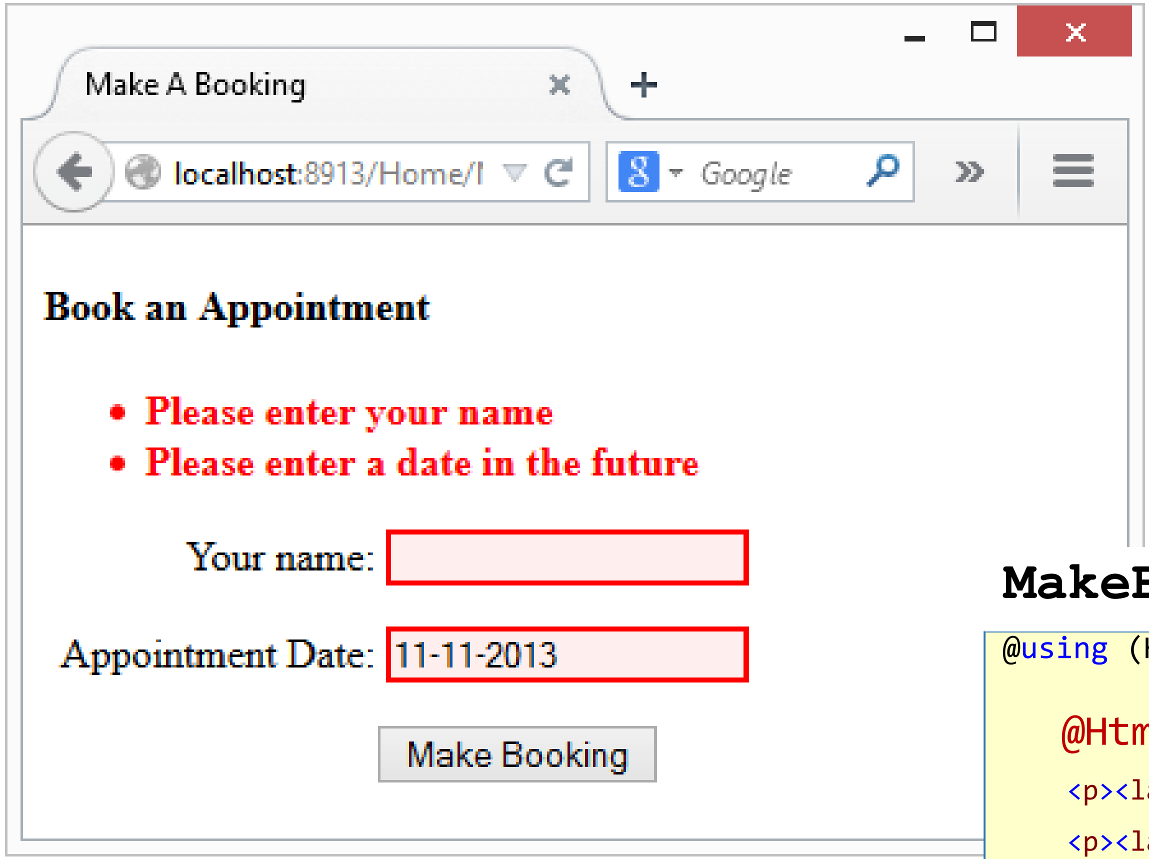
Your name: Peter Sørensen

Appointment Date: 11-11-2013

_Layout.cshtml

```
.input-validation-error {
    border: 2px solid #f00;
    background-color: #fee;
}
```

Visning af validation messages (model-level)



Book an Appointment

- Please enter your name
- Please enter a date in the future

Your name:

Appointment Date: 11-11-2013

Make Booking

_Layout.cshtml

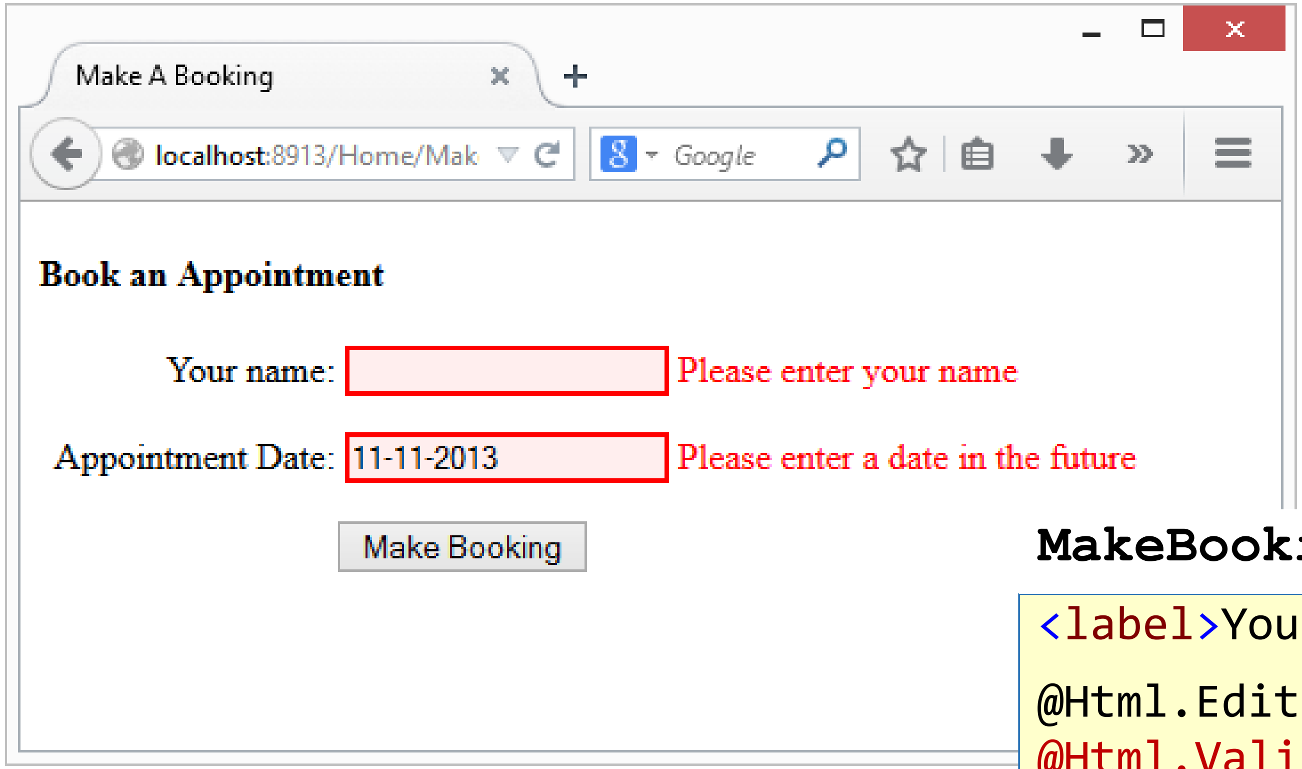
```
.validation-summary-errors {
    color: #f00;
    font-weight: bold;
}
```

MakeBooking.cshtml

```
@using (Html.BeginForm()) {

   @Html.ValidationSummary()
    <p><label>Your name:</label> @Html.EditorFor(m => m.FullName)</p>
    <p><label>Appointment Date:</label> @Html.EditorFor(m => m.Date)</p>
    <p><label> </label><input type="submit" value="Make Booking" /> </p>
}
```

Visning af validation messages (property-level)



Book an Appointment

Your name: Please enter your name

Appointment Date: 11-11-2013 Please enter a date in the future

Make Booking

_Layout.cshtml

```
.field-validation-error {
    color: #f00;
}
```

MakeBooking.cshtml

```
<label>Your name:</label>

@Html.EditorFor(m => m.FullName)
@Html.ValidationMessageFor(m => m.FullName)
```

Enable (eller disable) Client-Side Validation i MVC…på forskellige niveauer

Application Level (`web.config`)

```
<appSettings>
  <add key="ClientValidationEnabled" value="true"/>
  <add key="UnobtrusiveJavaScriptEnabled" value="true"/>
</appSettings>
```

Dette er **det mest almindelige** – så er client side validation sat **til i hele applikationen**.

View Level – muligt, men ikke så ofte.

```
@model MvcApp.Models.Appointment
@{
     ViewBag.Title = "Make A Booking";
     HtmlHelper.ClientValidationEnabled = false;
}
```

Attribute Level – muligt, men ikke ofte.

```
<div class="editor-field">
    @{ Html.EnableClientValidation(false); }
    @Html.TextBoxFor(m => m.BatchId)
    @{ Html.EnableClientValidation(true); }
</div>
```

Custom validation

- **Kører som udgangspunkt ikke på klienten.**
- **Det kan lade sig gøre at få det til at køre på klienten, men det kræver, at man**
  - laver et client side script, der gør det samme
  - kobler dette script op på attributten
- **Der findes opskrifter på dette, søg på nettet, men det er lidt langhåret**

Visning af validation errors serverside

- **Hvis valideringen sker på server-side:**
- **brug** `@Html.ValidationSummary(false, "", new { @class = "text-danger" })`

Self validating models

Et eksempel – bemærk interfacet.

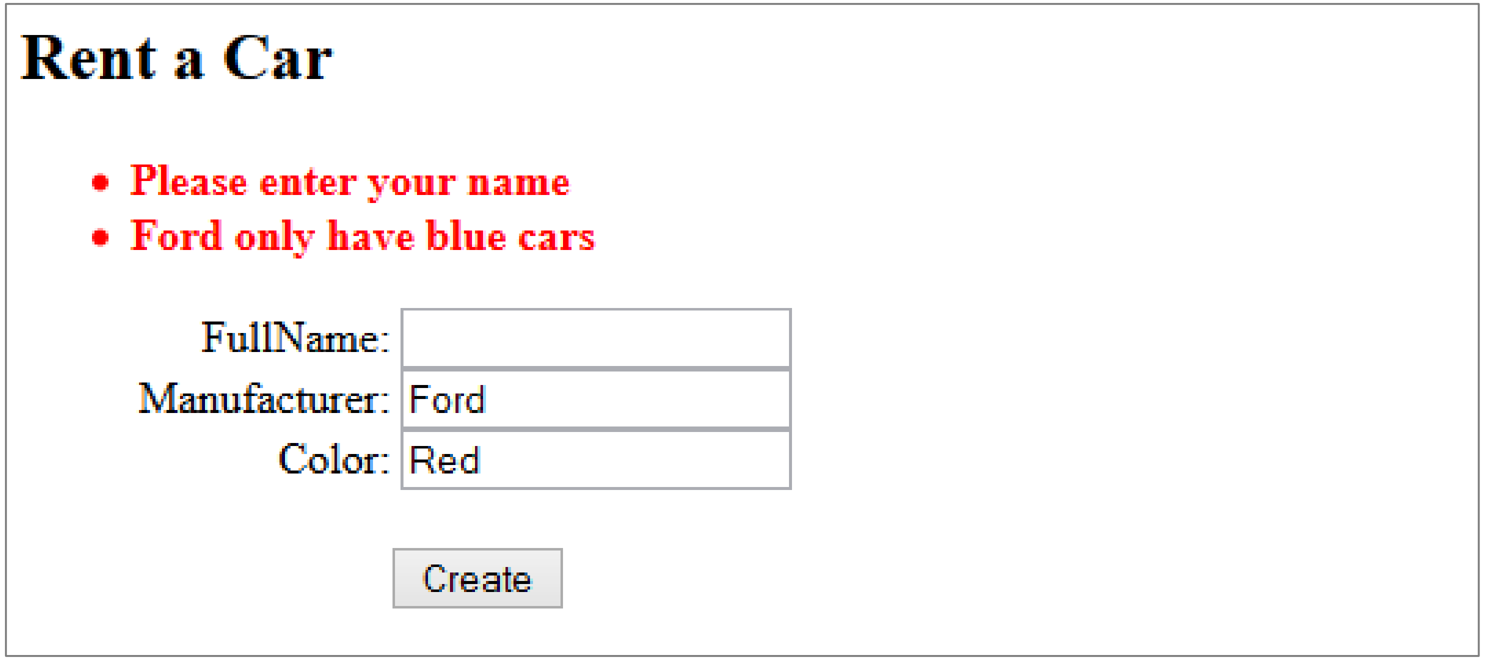
```
public class CarRent2 : IValidatableObject {
     //fields, gettters and setters here…

   public IEnumerable<ValidationResult> Validate(ValidationContext validationContext)
   {
      List<ValidationResult> errors = new
      List<ValidationResult>();
      if (string.IsNullOrEmpty(FullName)) {
         errors.Add(new ValidationResult("Please enter your name"));
      }
      if (string.IsNullOrEmpty(Manufacturer)) {
         errors.Add(new ValidationResult("Please enter a manufacturer"));
      }

      if (Manufacturer == "Ford" && Color != "Blue") {
         errors.Add(new ValidationResult("Ford only have blue cars"));
      }

      return errors;
   }
}
```

Ved self-validating models, så vises model og attribute-level validering sammen.



Rent a Car

- Please enter your name
- Ford only have blue cars

FullName:	
Manufacturer:	Ford
Color:	Red

Opgaver