# URL Routing, View/model Binding og ViewModels

# Intro

- Usability expert Jakob Nielsen (www.useit.com) anbefaler udviklere at have opmærksomhed på URLs og anbefaler følgende guidelindes for høje kvalitets URLs:
  - Korte URLs
  - Nemme at stave URLs
  - URLs som matcher site strukturen
  - Persistent URLs, som ikke ændrer sig
  - Og andre ting…

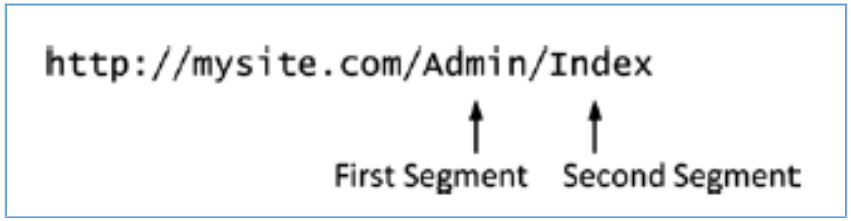
- Traditionelt , i mange web frameworks som JSP, PHP, så repræsenterer URL en fysisk fil på disk – f.eks. :
  - http://example.com/albums/list.aspx
- Disse URLs er ikke altid "pæne":
  - http://example.com/albums/list.aspx?catid=17313&genreid=33723&page=3
  - Eller http://example.com/main/albums/pages/lists/list.aspx?catid=17313

# ASP.NET Routing

- Er en separat API som Asp.NET MVC frameworket bruger til at matche URLs til kode.
- Mapper URLs til classes (of type Controller) and (Action) method kald

# Segmenter - et URL example

- URL med segments (2 vist her):




- Routes transformerer URLs til method calls (default setup):

```
{controller}/{action}
```

# Route eksempler

# Eksempel på URLs som matcher route herunder:

```
public static void RegisterRoutes(RouteCollection routes) {
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

    routes.MapRoute(
        name: "Default", // Route name
        url: "{controller}/{action}/{id}", // URL with
parameters
        defaults: new { controller = "Home", action = "Index",
id = UrlParameter.Optional } // Defaults
    );
}
```

| Request URL | Segment Variables |
|---|---|
| host/ | Controller: Home (default); Action: Index (default) |
| host/Products | Controller: Products; Action: Index (default) |
| host/Products/Books | Controller: Products; Action: Books |
| host/Products/Books/1218 | Controller: Products; Action: Books; Id: 1218 (optional) |
| host/Products/Books/Essays/1218 | No match, too many segments |

# Setup af en Route, hvor id er det 3. required segment?

```
public static void RegisterRoutes(RouteCollection routes) {
    ...
    routes.MapRoute(
        name: "Default", // Route name
        url: "{controller}/{action}/{id}", // URL with
parameters
        defaults: new { controller = "Home", action = "Index"}
// Defaults
    );
}
```

# Setup af en Route, hvor id er et optional segment med "first" som default value?

```
public static void RegisterRoutes(RouteCollection routes) {
    ...
    routes.MapRoute(
        name: "Default", // Route name
        url: "{controller}/{action}/{id}", // URL with
parameters
        defaults: new { controller = "Home", action = "Index",
id = "first" } // Defaults
    );
}
```

# Mere end en route

```
 routes.MapRoute(
                name: "Main",
                url: "None/{action}/{antal}",
                defaults: new { controller = "Main", action = "Index",
antal = UrlParameter.Optional }
            );
            routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}/{antal}",
                defaults: new { controller = "Home", action = "Index",
antal = UrlParameter.Optional}
            );
```

# Mere end en route

Der matches fra starten/øverst, så nedenstående vil give et andet resultat

```
 routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}/{antal}",
                defaults: new { controller = "Home", action = "Index",
antal = UrlParameter.Optional}
            );
            routes.MapRoute(
                name: "Main",
                url: "None/{action}/{antal}",
                defaults: new { controller = "Main", action = "Index",
antal = UrlParameter.Optional }
            );
```

# Setup af en Route, hvor id er 3. optional segment, som kun accepterer integers som værdi?

```
using System.Web.Mvc.Routing.Constraints;
public static void RegisterRoutes(RouteCollection routes) {
    ...
    routes.MapRoute(
        name: "Default", // Route name
        url: "{controller}/{action}/{id}", // URL
        defaults: new { controller = "Home", action = "Index",
id = 12 }, // Defaults
        constraints : new { id = new IntRouteConstraint() } //
Constraints
    );
```

# Route Constraint Classes

| Name | Description | Attribute Constraint |
|---|---|---|
| `AlphaRouteConstraint()` | Matches alphabet characters, irrespective of case (A–Z, a–z) | `alpha` |
| `BoolRouteConstraint()` | Matches a value that can be parsed into a `bool` | `bool` |
| `DateTimeRouteConstraint()` | Matches a value that can be parsed into a `DateTime` | `datetime` |
| `DecimalRouteConstraint()` | Matches a value that can be parsed into a `decimal` | `decimal` |
| `DoubleRouteConstraint()` | Matches a value that can be parsed into a `double` | `double` |
| `FloatRouteConstraint()` | Matches a value that can be parsed into a `float` | `float` |
| `IntRouteConstraint()` | Matches a value that can be parsed into an `int` | `int` |
| `LengthRouteConstraint(len)`<br>`LengthRouteConstraint(min, max)` | Matches a value with the specified number of characters or that is between `min` and `max` characters in length. | `length(len)`<br>`length(min, max)` |
| `LongRouteConstraint()` | Matches a value that can be parsed into a `long` | `long` |
| `MaxRouteConstraint(val)` | Matches an `int` value if the value is less than `val` | `max(val)` |
| `MaxLengthRouteConstraint(len)` | Matches a string with no more than `len` characters | `maxlength(len)` |
| `MinRouteConstraint(val)` | Matches an `int` value if the value is more than `val` | `min(val)` |
| `MinLengthRouteConstraint(len)` | Matches a string with at least `len` characters | `minlength(len)` |
| `RangeRouteConstraint(min, max)` | Matches an `int` value if the value is between `min` and `max` | `range(min, max)` |

Freeman: Pro ASP NET MVC 5, pp. 403-04

# Setup af en Route som bruger “shop” som et statisk (fast) URL segment?

- Matched URL’s
  - *host*/shop/books/
  - *host*/shop/books/programming/pro-asp-net-mvc5
  - *host*/shop/movies/action
  - *host*/shop/movies/drama/the-shining

# Løsningen er simpel :

```
routes.MapRoute(
   // Route name
   name: "Shop",
   // URL with parameters
   url: "shop/{controller}/{category}/{title}",
   // Defaults
   defaults: new { action = "Index",
        category = UrlParameter.Optional,
        title = UrlParameter.Optional }
);
```

ASP.NET MVC har en konfigurationsfil, som søger for at registrere alle de ruter som er i RouteConfig.

- `Global.asax filen:`

```
public class MvcApplication : System.Web.HttpApplication
{
    protected void Application_Start()
    {
        AreaRegistration.RegisterAllAreas();
        RouteConfig.RegisterRoutes(RouteTable.Routes);
    }
}
```

# URLs i Views

# @Html.ActionLink

```
@Html.ActionLink("Same controller and action", "index")

@Html.ActionLink("Same controller and action with URL parameters",  "index",
new { page = 2, sortorder = "Author" })

 @Html.ActionLink("Another Controller”,”index”, ”books”)

 @html.ActionLink(”Another controller wiht URL parameters,
 ”Index”,”books, new {category = ”Fiction”, title =”The
 Idiot”),null) //last parameter is styling of link
```

# External links – full version

```
@Html.ActionLink(
   "External Link",
   "Index", // Action method
   "Home", // Controller
   "https", // Protocol
   "myserver.mydomain.com", // Domain
   "segment", // Html segment
   new { id = "MyId" }, // additional segments
   new { id = "myAnchorID", @class = "myCSSClass" }
   // HTML attributes
)
```

# Model binding
# Binding til Simple Typer i controlleren

```
// incoming URL: /Index1/24
public string Index1(int id) { // Required
      return id.ToString();
}
// incoming URL: /Index2 || /Index2/24 || /Index2?id=24
public string Index2(int? id) { // Nullable
      return id.ToString();
}

public string Index3(int id = 12) { // Default
      return id.ToString();
}
```

# Model binding
# FormCollection

```
[HttpPost]
public ActionResult Create(FormCollection formCollection)
Employee employee = new Employee();
// Retrieve from data using form collection
employee.Name = formCollection["Name"];
...
```

# Model binding
# Simple typer i Action

```
[HttpPost]
public ActionResult CreateFromSimpleTypes(string name, string
gender, string city, decimal Salary, DateTime dateOfBirth)
{
Employee employee = new Employee();
employee.Name = name;
...
```

# Model binding kompleks type

```
[HttpPost]
public ActionResult CreateFromModel(Employee employee)
{
…
```

# Model binding
# View

```
<h2>Create Employee</h2>
@using (Html.BeginForm("Create", "Home"))
{
  @Html.LabelFor(model => model.Name, htmlAttributes: new { @class =
"control-label col-md-2" })
<div class="col-md-10">
  @Html.EditorFor(model => model.Name, new { htmlAttributes = new {
@class = "form-control" } })
...
```

# Model binding

Se mere på
https://dotnettutorials.net/lesson/model-binding-asp-net-mvc/

# What is attribute routing and how do you enable it?

- Since ASP.NET MVC 5, the framework also supports attribute routing.
- Enabling (`RouteConfig.cs`):

```
public class RouteConfig
{
    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
        routes.MapMvcAttributeRoutes();
    }
}
```

# The Route

Setup an attribute route that maps URLs like:

- `/books`
- `/books/1430210079`

```
public class BooksController : Controller {

    [Route("books/{isbn?}")]
    public ActionResult View(string isbn) {
        if (!String.IsNullOrEmpty(isbn)) {
            return View("OneBook", GetBook(isbn));
        }
        return View("AllBooks", GetBooks());
    }
}
```

# Example: Setup Three Attribute Routes

```
public class ReviewsController : Controller {
    // eg: /reviews
    [Route("reviews")]
    public ActionResult Index() { ... }
    // eg: /reviews/5
    [Route("reviews/{reviewId}")]
    public ActionResult Show(int reviewId) { ... }
    // eg: /reviews/5/edit
    [Route("reviews/{reviewId}/edit")]
    public ActionResult Edit(int reviewId) { ... }
}
```

# Q12: Pros and cons of attribute routing?

| | Pro | Cons |
|---|---|---|
| **Convention-based Routing** | • Separation of concerns – controllers have no knowledge or dependency on routing configuration<br>• All routing info in one file | • More work to setup |
| **Attribute Routing** | • Easier to grasp & understand<br>• Easier to route advanced URIs like **/students/1/courses** to get all courses of student with id=1 | • No separation of concerns. You have to go into the code of your controllers to see where the route maps. |

A good link:
https://dotnettutorials.net/lesson/attribute-routing-imvc/

# ViewModels

Problem: Vi kan kun have en @model defineret i et view. Så hvordan kan vi overføre flere typer? (uden brug af ViewBag)?

# Løsning:
# ViewModels (en slags container klasse)

```
using ViewModelsDemo.Models;
using System.Collections.Generic;

//Normalt i ViewModels, og ikke Models folder
namespace ViewModelsDemo.ViewModels
{
    public class CustomerViewModel
    {
        public List<Book> Books { get; set; }
        public List<Customer> Customers { get; set; }
    }
}
```

# I **controlleren** i action metode:

```
//bøger også initialiseret – kode ikke vist her.
var customer = new List<Customer>()
       {
             new Customer {CustomerName = "Zain"},
             new Customer {CustomerName = "Hassan"},
             new Customer {CustomerName = "Syed"}
         };

          var CustomerViewModel = new CustomerViewModel
          {
               Books = books,
               Customers = customer
          };

           return View(CustomerViewModel);
```

# viewet:

```
@model ViewModelsDemo.ViewModels.CustomerViewModel

<h2>List of Books and list of Customers together</h2>

<h4>List of Books</h4>
<ul>
   @foreach (var book in Model.Books)
  {
     <li>
          @book.BookName;
       </li>
   }
</ul>


<h4>List of Customers</h4>
<ul>
 @foreach (var customer in Model.Customers)
 {
  <li>
     @customer.CustomerName;
  </li>
  }
</ul>
```

# Opgaver!