pawel.rajba@gmail.com
http://www.itcourses.eu/

ASP.NET MVC

Agenda

- Wprowadzenie
- Architektura
- Routing
- Cookies
- Sesje
- Razor
- Formularze

Wprowadzenie

- Silnie rozwijany produkt firmy Microsoft
- Jeden w dwóch referencyjnych rozwiązań
  - ASP.NET WebForms
  - ASP.NET MVC
- Bazuje na platformie .NET
- Serwer aplikacji to IIS
- Wsparcie w wielu obszarach
  - np. formularze w walidacją, authN & authZ, dostęp do danych, debugowanie,

Podstawy

- Wzorzec MVC
- Podstawowe elementy
  - Kontrolery, akcje
  - Widoki
- Web.config
- Obiekty Request, Response, Server

Podstawy

- Przekazywanie danych z akcji do widoku
  - ViewData
    - Kontroler
      ViewData["Message"]="Hello World!";
    - Widok
      @ViewData["Message"]
  - ViewBag
    - Kontroler
      ViewBag.Message = "Hello World!";
    - Widok
      @ViewBag.Message
- Obu można stosować wymiennie
  - Opakowują te same dane

Podstawy

- Co może zwracać akcja kontrolera?
  - ContentResult
  - EmptyResult
  - FileResult
  - HttpUnauthorizedResult
  - JavaScriptResult
  - JsonResult
  - RedirectResult
  - RedirectToRouteResult
  - ViewResultBase
- Dla każdego typu jest odpowiednia metoda, np.
  - View(), Json(), File(), ...

Podstawy

- Global.asax
  - Application_Start
  - Application_BeginRequest
  - Application_AuthenticateRequest
  - Session_Start
  - Application_EndRequest
  - Session_End
  - Application_End
  - Application_Error

Routing

- Routing, czyli mapowanie ścieżek na kontrolery i akcje
- Rejestracja ścieżek: RouteTable.Routes.MapRoute
- Przykłady
  - `routes.IgnoreRoute("{resource}.axd/{*pathInfo}");`
  - ```
    routes.MapRoute(
        "Default", // Route name
        "{controller}/{action}/{id}", // URL with parameters
        new { controller = "Home", action = "Index",
              id = UrlParameter.Optional } // Parameter defaults
    );
    ```
  - ```
    routes.MapRoute("CatchAll", "{*values}",
        new { controller = "Default", action = "Dispatch" },
        new { values = @"[a-zA-Z0-9-]*"},
        new string[] { "WebApplication.Frontend.Controllers" });
    ```
- Do poczytania

  http://stephenwalther.com/archive/2009/02/06/chapter-2-understanding-routing.aspx

DEMO

- Oglądamy FirstSample
- Tworzymy
  - Intranet Application
  - Internet Application
    - `\%windir%\Microsoft.NET\Framework\`*`version`*`\Aspnet_regsql.exe`
  - Empty
    - dodajemy niezbędne elementy do uruchomienia
- Oglądamy obiekty Request, Response i Server

Cookies

- Mamy dwie kolekcje
  - Request.Cookies
  - Response.Cookies
- Obiekt HttpCookie
  - Reprezentuje ciacho
  - Ciekawa property
    - HasKeys
    - Values

Sesje

- Obiekt
  - Session
- Konfiguracja
  - ```
    <configuration>
      <system.web>
        <sessionState cookieName="" cookieless="" timeout="" />
      </system.web>
    </configuration>
    ```
- Do poczytania
  - http://msdn.microsoft.com/en-us/library/h6bb9cz9(v=vs.100).aspx
    ```

Razor

- Język szablonów
- Czytelny i oferuje wygodne konstrukcje
  - Blok kodu

```
@{
    string s = "this is string";
}
```

  - Zmienna kodowana i niekodowana

```
@model.Message

@Html.Raw(model.Message)
```

  - Pętla i IF

```
@foreach(var item in items) {
  <span>@item.Prop</span>
}

@if (foo) {
  @:Plain Text is @bar
}
```

Razor

- Szablony
  - Domyślne ustawienia (w tym szablon): _ViewStart.cshtml
  - W widoku można też wskazać szablon

```
@{
    Layout = "~/Views/Shared/_Layout2.cshtml";
}
```

- Helpery
  - Html.*
- Quick reference
  - http://haacked.com/archive/2011/01/06/razor-syntax-quick-reference.aspx

DEMO

- Templates

Formularze

- DEMO
  - FormSimple
  - FormModel
  - FormValidation
  - FormFilter