



ASP.NET



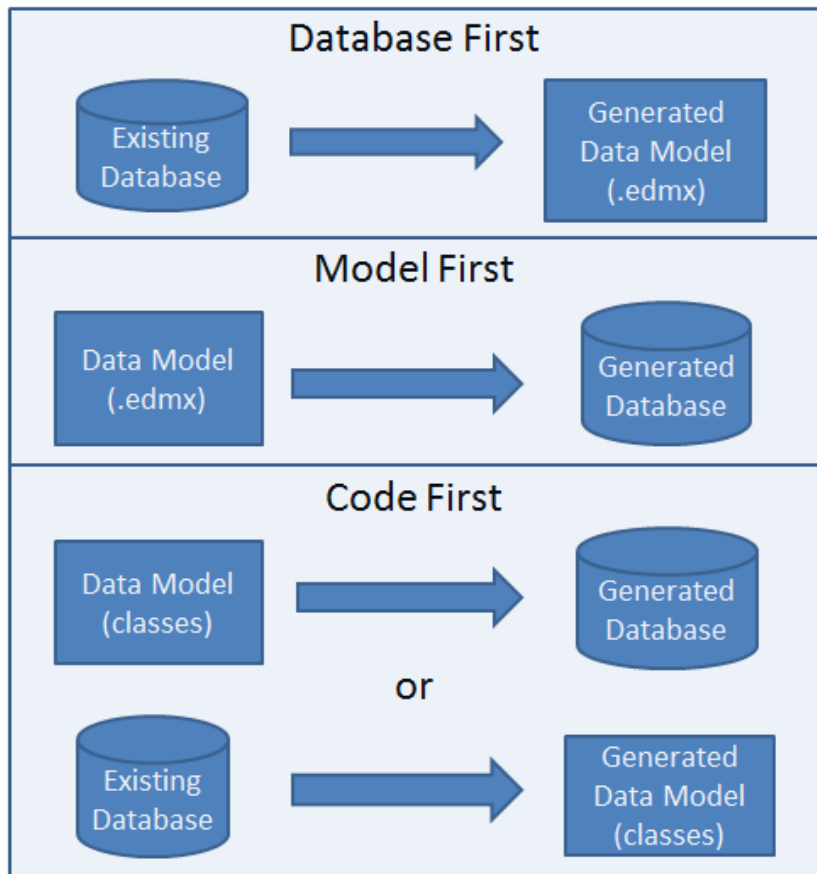
MVC

ASP.NET MVC

- ▶ Model dający największą kontrolę nad aplikacją
- ▶ Najlepiej nadaje się do dużych projektów
- ▶ Separacja warstw
- ▶ Test-Driven Development
- ▶ Zaawansowany mechanizm mapowania adresów url
- ▶ Automatycznie generowane szkielety kontrolerów i widoków
- ▶ Narzędzie – Visual Studio



Model



```
using System.ComponentModel.DataAnnotations;

namespace Wykład_AP.NET.Models
{
    public class Book
    {
        [ScaffoldColumn(false)]
        public int Id { get; set; }

        [Required]
        [StringLength(1000)]
        public string Title { get; set; }

        [Required]
        [StringLength(10)]
        public string Author { get; set; }
    }
}
```

Kontroler

Add Controller

Controller name:
BooksController

Scaffolding options
Template:
MVC controller with read/write actions and views, using Entity Framework

Model class:
Book (Wykład_AP.NET.Models)

Data context class:
Wykład_AP.NET.Models.BooksContext

Views:
Razor (CSHTML) ▼

Adv...

Ac...

```
namespace Wykład_AP.NET.Controllers
{
    public class BooksController : Controller
    {
        private BooksContext db = new BooksContext();

        // GET: /Books/
        public ActionResult Index()
        {
            return View(db.Books.ToList());
        }

        // GET: /Books/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: /Books/Create
        [HttpPost]
        public ActionResult Create(Book book)
        {
            if (ModelState.IsValid)
            {
                db.Books.Add(book);
                db.SaveChanges();
                return RedirectToAction("Index");
            }

            return View(book);
        }
    }
}
```

...

Widok

The image illustrates the process of creating a view in an ASP.NET application. It is divided into three main sections:

- Add View Dialog:** Shows the configuration for a new view. The 'View name' is 'Index', the 'View engine' is 'Razor (CSHTML)', and the 'Scaffold template' is 'List'. The 'Model class' is 'Book (Wykład_AP.NET.Models)'. The 'ContentPlaceHolder ID' is 'MainContent'. The 'Create a strongly-typed view' checkbox is checked.
- Razor View Code:** The code for the 'Index' view, which is a list view. It includes a title, a 'Create New' link, and a table of books. The code is as follows:

```
@model IEnumerable<Wykład_AP.NET.Models.Book>

@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table>
    <tr>
        <th>@Html.DisplayNameFor(model => model.Title)</th>
        <th>@Html.DisplayNameFor(model => model.Author)</th>
        <th></th>
    </tr>
    @foreach (var item in Model)
    {
        <tr>
            <td>@Html.DisplayFor(m => m.Title)</td>
            <td>@Html.DisplayFor(m => m.Author)</td>
            <td>
                @Html.ActionLink("Edit", "Edit", new { id=item.Id }) |
                @Html.ActionLink("Details", "Details", new { id=item.Id }) |
                @Html.ActionLink("Delete", "Delete", new { id=item.Id })
            </td>
        </tr>
    }
</table>
```
- Rendered View:** The output of the view, showing the 'Index' page. It includes a 'Create New' link and a table of books. The table has columns for 'Title', 'Author', and a set of links for each book. The data is as follows:

Title	Author	
The Dark Tower	Stephen King	Edit Details Delete
Doctors	Erich Segal	Edit Details Delete
Scar Tissue	Anthony Kiedis	Edit Details Delete

Przebieg ćwiczenia

- ▶ Podstawy
- ▶ Utworzenie modelu – podejście „Code First”
- ▶ Wygenerowanie kontrolera oraz widoków
- ▶ Dostosowanie aplikacji do własnych potrzeb
- ▶ Proste żądania asynchroniczne



Przydatne materiały

- ▶ <http://www.asp.net/mvc>
- ▶ <http://pluralsight.com/training/Player?author=scott-allen&name=mvc4-building-m1-intro&mode=live&clip=0&course=mvc4-building>
- ▶ <http://code.msdn.microsoft.com/101-LINQ-Samples-3fb9811b>
- ▶ <http://www.asp.net/mvc/tutorials/hands-on-labs/aspnet-mvc-4-entity-framework-scaffolding-and-migrations>

