

ASP.NET MVC: PRAKTICKÝ PŘÍKLAD

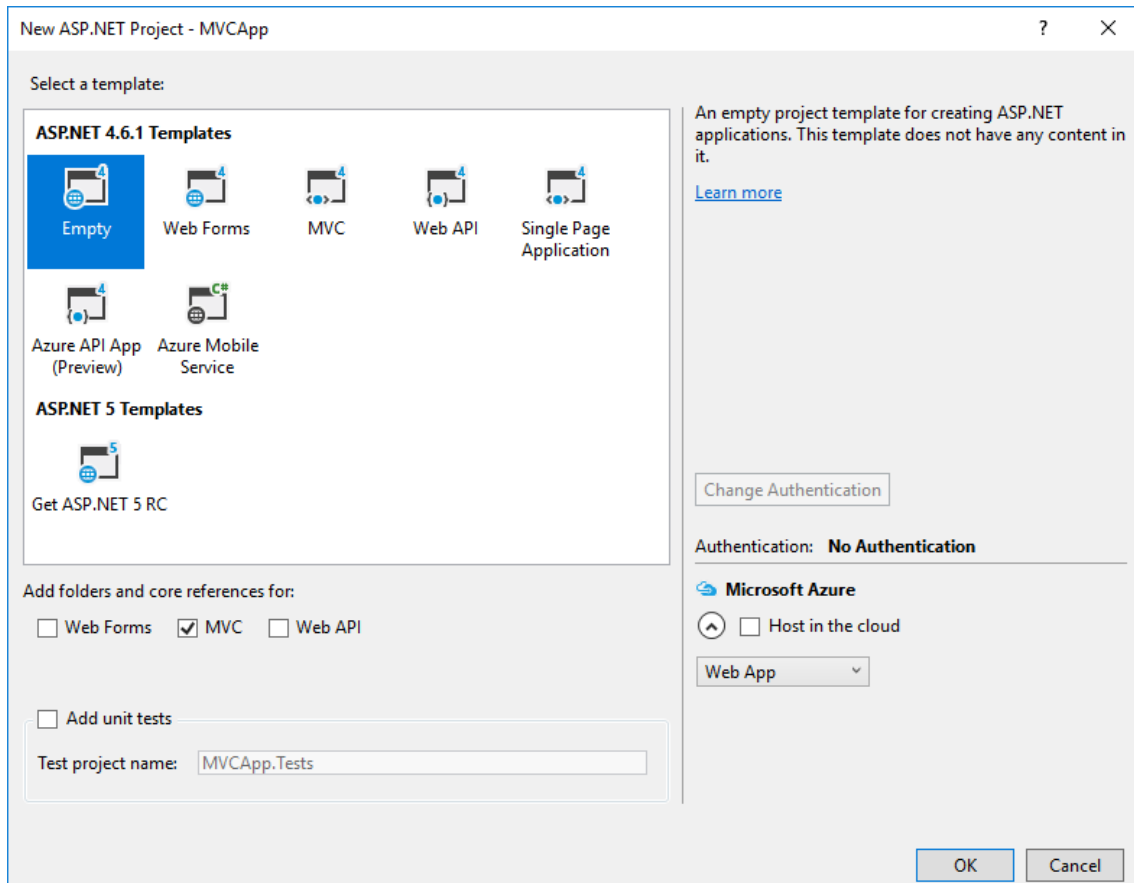
Softwarové inženýrstvo

MARTIN TIMOTHY TIMKO

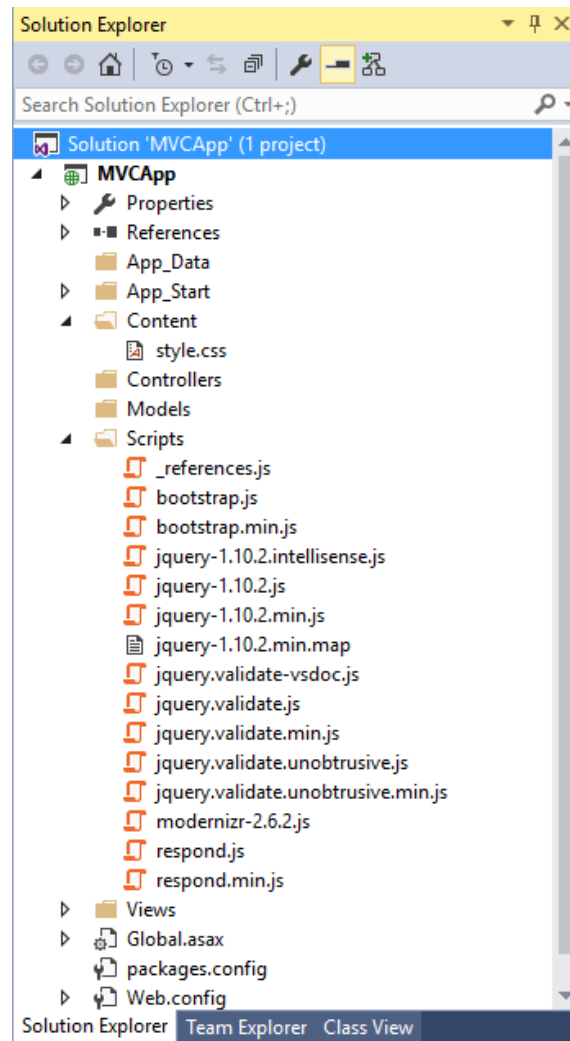
12.1. 2018

1 ÚVOD

Princíp architektonického vzoru MVC je pomerne jednoduchý z hľadiska funkčnosti. Ukážkovým príkladom môže byť jednoduchá aplikácia, ktorá kooperuje medzi jednotlivými komponentmi. Za týmto účelom je možné vytvoriť prázdnu MVC aplikáciu.



Do prázdneho projektu vytvoríme adresáre *Content* a *Scripts*, pričom jednotlivé súbory sú k dispozícii na oficiálnych stránkach *Bootstrap* alebo je možné ich skopírovať z automaticky generovaného projektu MVC.



Jedine súbor kaskádových štýlov je mierne upravený:

```
body,
html {
margin:0;
padding:0;
color:#000;
background:#ddd;
}
#wrap {
width:750px;
margin:0 auto;
background:#808080;
}
#header {
```

```
        padding:5px 10px;
background:#0094ff;
}
h1 {
    margin:0;
}
#nav {
padding:5px 10px;
background:#000;
}
#nav ul {
margin:0;
padding:0;
list-style:none;
}
#nav li {
display:inline;
margin:0;
padding:0;
}
    #nav a {
color:#ffffff;
}
#main {
float:left;
width:480px;
padding:10px;
background:#b5b5b5;
}
h2 {
margin:0 0 1em;
}
#sidebar {
float:right;
width:230px;
padding:10px;
background:#b5b5b5;
}
#footer {
clear:both;
padding:5px 10px;
background:#0094ff;
}
#footer p {
margin:0;
}
* html #footer {
```

```
height:1px;
}
```

Ako je možné vidieť, vzor MVC sa rozprestiera v troch adresároch, a teda model v adresári Models, pohľad v adresári View a ovládač v adresári Controllers.

Model (Model) obsahuje tzv. dátovú vrstvu alebo business vrstvu, ktorá zabezpečuje vo veľkej miere nejaké dáta rôznej povahy, najmä databázovej, ale aj výpočty a podobne. Jedná sa o klasický C# kód, ktorý korešponduje s tzv. *CodeBehind*. Model funguje vo forme nejakého skladu, kde nejakým spôsobom dáta prichádzajú a odchádzajú. Forma, akou tieto dáta navštevujú daný model je zásluhou parametrov metód, teda nejedná sa o žiadne formátovacie výstupy a podobne.

Takže môžeme vytvoriť jednoduchú triedu, ktorá nám bude vracať zoznam kníh. Princiálne nezáleží na forme týchto dát, takže by určite bolo možné vytvoriť nejaký databázový dátový zdroj podobným spôsobom ako v článku *SQL Server: ADO.NET* alebo *SQL Server: Entity Framework*.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace MVCApp.Models
{
    public class Book
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Author { get; set; }

        public static List<Book> GetBooks()
        {
            List<Book> books = new List<Book>();
            books.Add(new Book() { Id = 1, Name = "Nazov_1", Author = "Autor_1" });
            books.Add(new Book() { Id = 2, Name = "Nazov_2", Author = "Autor_2" });
            books.Add(new Book() { Id = 3, Name = "Nazov_3", Author = "Autor_3" });

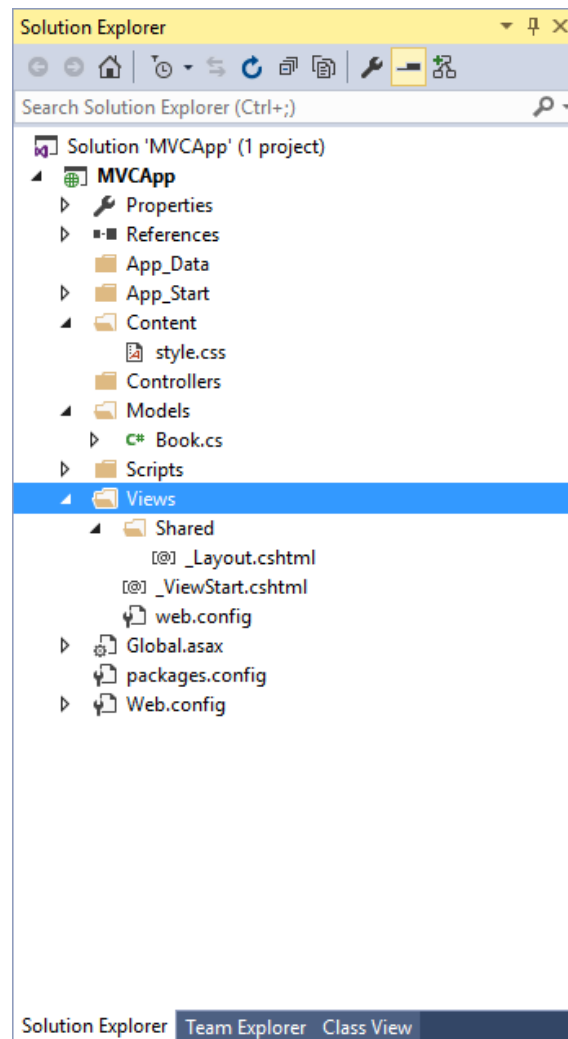
            return books;
        }
    }
}
```

Táto trieda bude tvoriť náš model, jednoduchý dátový zdroj.

Teraz môžeme začať tvoriť pohľady.

Pohľad (View) ako už z názvu vyplýva, bude obsahovať všetko, čo budeme mať v úmysle zobrazit' používateľovi na web stránke. V tomto ohľade sa jedná o hybridné súbory typu *Razor* s príponou *.cshtml, teda o určitú možnosť písania kódu v jazyku C# a zároveň HTML s JavaScript. Takže na jednej stránke je možné sa stretnúť až s tromi jazykmi. Pohľady môžu byť jedinečné, pre každú konkrétnu stránku, alebo spoločné, ktoré sa zobrazujú všetkým stránkam vo forme určitej šablóny. Pohľad podobne ako Model je nezávislý, zobrazuje nejaké dáta používateľovi, ale zdroj týchto dát nepozná.

Môžeme predbežne vytvoriť všetky pohľady. Takže adresárová štruktúra pohľadových súborov je nasledujúca:



Adresár Home obsahuje konkrétne web stránky a adresár Shared obsahuje ako bolo spomínané práve tie zdieľané pohľady. Z ASP.NET WebForms to je možné prirovnať z Master Page, kde bolo možné definovať základnú stránku, ktorá sa zobrazovala pre všetky ostatné stránky. Takže v našom prípade je potrebné vytvoriť preddefinovaný odkaz `_ViewStart.cshtml`, ktorým uvedieme odkiaľ je možné danú zdieľanú šablónu používať.

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

A samotnú zdieľanú šablónu `_Layout.cshtml`:

```
<!DOCTYPE html>
```

```

<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Moj MVC test</title>

  <link href="style.css" rel="stylesheet">
  <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet">
  <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.min.css" rel="stylesheet">
  <link href="@Url.Content("~/Content/style.css")" rel="stylesheet" />

  <script src="@Url.Content("~/Scripts/jquery-1.9.1.min.js")"></script>
  <script src="@Url.Content("~/scripts/bootstrap.min.js")"></script>
</head>
<body>
  <div id="wrap">
    <div id="header"><h1>MVC test</h1></div>
    <div id="nav">
      <ul>
        <li><a href="@Url.Action("Index", "Home")">Domov</a></li>
        <li><a href="@Url.Action("Prva", "Home")">Prvá</a></li>
        <li><a href="@Url.Action("Druha", "Home")">Druhá</a></li>
        <li><a href="@Url.Action("Tretia", "Home")">Tretia</a></li>
        <li><a href="@Url.Action("Stvrta", "Home")">Štvrtá</a></li>
      </ul>
    </div>
    <div id="main" style="background-color: #b5b5b5">

      @RenderBody()

    </div>
    <div id="sidebar">
      <h2>Sidebar</h2>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris vel magna.</p>
      <ul>
        <li><a href="/">Link 1</a></li>
        <li><a href="/">Link 2</a></li>
        <li><a href="/">Link 3</a></li>
        <li><a href="/">Link 4</a></li>
        <li><a href="/">Link 5</a></li>
        <li><a href="/">Link 6</a></li>
        <li><a href="/">Link 7</a></li>
        <li><a href="/">Link 8</a></li>
        <li><a href="/">Link 9</a></li>
        <li><a href="/">Link 10</a></li>
        <li><a href="/">Link 11</a></li>
        <li><a href="/">Link 12</a></li>
      </ul>
    </div>
  </div>

```

```

        <li><a href="/">Link 13</a></li>
        <li><a href="/">Link 14</a></li>
        <li><a href="/">Link 15</a></li>
    </ul>

    @RenderSection("Sidebar", false)

</div>
<div id="footer">
    <p>Footer</p>
</div>
</div>
</body>
</html>

```

Ako je možné vidieť šablóna obsahuje celú webovú stránku s odkazmi na jednotlivé kaskádové štýly ako aj JavaScriptové súbory. Obsah konkrétnych web stránok bude zobraziteľný v sekcii `@RenderBody()`. To znamená, že zobrazí sa zdieľaná šablóna a v mieste bližšie určenom touto funkciou sa zobrazí časť konkrétnej web stránky. Zdieľaná šablóna obashuje aj funkciu `@RenderSection("Sidebar", false)`, ktorá predstavuje podobnú funkcionality. V mieste určenia tejto funkcie sa zobrazí v špecifickej situácii ďalšia určitá časť konkrétnej web stránky. Jednotlivé URL odkazy odkazujú na konkrétny ovládač (Controller).

Ovládač (Controller) je prakticky najdôležitejšia komponenta celého vzoru MVC, pretože prostredníctvom tohto prvku komunikujú všetky komponenty navzájom. Každý ovládač môže obsahovať jeden alebo viac pohľadov. Konkrétny ovládač podľa parametrov zistí akú akciu je potrebné vykonať. Následne sa zavolá model, ktorý vráti nejaké dáta do ovládača, ten ich nejakým spôsobom pripraví a posiela do pohľadu. Názov pohľadu zistíme podľa akcie, ktorú vykonávame. Teda, prvý kontakt používateľa s danou web stránkou sa realizuje prostredníctvom ovládača alebo pomocou JavaScriptu priamo z pohľadu.

Takže kompletný ovládač `HomeController.cs` bude vyzeráť nasledovne:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Data;
using System.Data.SqlClient;
using MVCApp.Models;

namespace mojeMVC.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            string retazec = "retazec";

```



```

        ViewBag.Retazec = retazec;

        return View();
    }

    public ActionResult Prva()
    {
        return View();
    }

    public ActionResult Druha()
    {
        string pozdrav = "Ahoj";
        int cislo = 17;

        ViewBag.Pozdrav = pozdrav;
        ViewBag.Cislo = cislo;

        return View();
    }

    public ActionResult Tretia(int id = 2)
    {
        var b = from Book book
                in Book.GetBooks()
                where book.Id == id
                select book;

        ViewBag.Datum = DateTime.Now;

        return View(b);
    }

    public ActionResult Stvrta()
    {
        List<Book> zoznamKnih = Book.GetBooks();

        ViewBag.ZoznamKnih = zoznamKnih;

        return View();
    }
}
}

```

Jedná sa o podobný súbor akým disponuje model, takže v zásade je to čistý jazyk C# vo forme tzv. *CodeBehind*. Metóda `Index()` sa zavolá v prípade, ak používateľ vyvolá domovskú stránku. Podobne to platí aj pre ostatné metódy, napr. `Prva()`, `Druha()` až `Stvrta()`, s tým, že každý

tento názov špecifikuje daný pohľad. Návratový typ všetkých týchto metód je `ActionResult`, čo predstavuje objekt, ktorý po dokončení požiadavky posielame prehliadaču. V našom prípade posielame šablónu `View()` s určitými parametrami. Samozrejme, je možné posielat' aj zložitejšie dátové typy.

K samotnej manipulácii dát dochádza, okrem uvedeného, aj troma kolekciami, konkrétne:

- `ViewData`: kolekcia vo forme slovníka, kde pridávame jednotlivé premenné pre šablónu pod nejakými textovými kľúčmi
- `ViewBag`: kolekcia, ktorá používa dynamické vlastnosti, kde namiesto kľúčov zapisujeme priamo do vlastností, ktoré sa na strane `ViewBag` vytvoria
- `TempData`: kolekcia, ktorá sa používa pri presmerovaní, pričom jednotlivé dáta sú po dokončení aktuálnej požiadavky následne odstránené

Vzhľadom k uvedenému je najrozšírenejšia kolekcia, `ViewBag`, pretože nám poskytuje väčšie možnosti dátového použitia, teda disponuje bližšie nešpecifikovaným charakterom.

K takto pripravenému obsahu je možné generovať samostatné pohľadové stránky.

2 DOMOV


Takže Index.cshtml bude vyzerat' nasledovne:

```
<h3>Domov</h3>
```

```
<p>Výstup: @ViewBag.Retazec</p>
```

Teda, po stlačení na odkaz Domov sa vykoná nasledovný kód z ovládača:

```
public ActionResult Index()  
{  
    string retazec = "retazec";  
  
    ViewBag.Retazec = retazec;  
  
    return View();  
}
```



The screenshot shows a web application interface. At the top, there is a blue header with the text "MVC test". Below the header is a black navigation bar with the text "Domov Prvá Druhá Tretia Štvrtá". The main content area is divided into two columns. The left column has a grey background and contains the text "Domov" and "Výstup: retazec". The right column has a grey background and contains the text "Sidebar" followed by a paragraph of Lorem Ipsum text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris vel magna." Below the text is a list of 15 links, each labeled "Link 1" through "Link 15". At the bottom of the page, there is a blue footer with the text "Footer".

To znamená, že jednoduchý reťazec predáme do pohľadu prostredníctvom kolekcie ViewBag.

3 PRVÁ STRÁNKA

Prva.cshtml bude vyzerat' nasledovne:

```
<h3>Prvá stránka</h3>
```

```
<h4>Tlačidlá</h4>
```

```
<p>
```

```
  <button type="button" class="btn btn-sm btn-default">Default</button>
```

```
  <button type="button" class="btn btn-sm btn-primary">Primary</button>
```

```
  <button type="button" class="btn btn-sm btn-success">Success</button>
```

```
  <button type="button" class="btn btn-sm btn-info">Info</button>
```

```
  <button type="button" class="btn btn-sm btn-warning">Warning</button>
```

```
  <button type="button" class="btn btn-sm btn-danger">Danger</button>
```

```
  <button type="button" class="btn btn-sm btn-link">Link</button>
```

```
</p>
```

```
<h4>Tabul'ky</h4>
```

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

```
  <thead>
```

```
    <tr>
```

```
      <th>#</th>
```

```
      <th>First Name</th>
```

```
      <th>Last Name</th>
```

```
      <th>Username</th>
```

```
    </tr>
```

```
  </thead>
```

```
  <tbody>
```

```
    <tr>
```

```
      <td>1</td>
```

```
      <td>Mark</td>
```

```
      <td>Otto</td>
```

```
      <td>mdo</td>
```

```
    </tr>
```

```
    <tr>
```

```
      <td>2</td>
```

```
      <td>Jacob</td>
```

```
      <td>Thornton</td>
```

```
      <td>fat</td>
```

```
    </tr>
```

```
    <tr>
```

```
      <td>3</td>
```

```
      <td>Larry</td>
```

```
      <td>the Bird</td>
```

```
      <td>twitter</td>
```

```

        </tr>
    </tbody>
</table>

<h4>Značky</h4>

<h3>
    <span class="label label-default">Default</span>
    <span class="label label-primary">Primary</span>
    <span class="label label-success">Success</span>
    <span class="label label-info">Info</span>
    <span class="label label-warning">Warning</span>
    <span class="label label-danger">Danger</span>
</h3>

<h4>Menu a navigácia</h4>

<div class="dropdown theme-dropdown clearfix">
    <a id="dropdownMenu1" href="#" class="sr-only dropdown-toggle" data-toggle="dropdown" role="button">
    <ul class="dropdown-menu" aria-labelledby="dropdownMenu1">
        <li class="active"><a href="#">Action</a></li>
        <li><a href="#">Another action</a></li>
        <li><a href="#">Something else here</a></li>
        <li role="separator" class="divider"></li>
        <li><a href="#">Separated link</a></li>
    </ul>
</div>

<nav class="navbar navbar-default">
    <div class="container">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar"
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="#">Project name</a>
        </div>
        <div class="navbar-collapse collapse">
            <ul class="nav navbar-nav">
                <li class="active"><a href="#">Home</a></li>
                <li><a href="#about">About</a></li>
                <li><a href="#contact">Contact</a></li>
                <li class="dropdown">
                    <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true"
                    <ul class="dropdown-menu">

```

```

        <li><a href="#">Action</a></li>
        <li><a href="#">Another action</a></li>
        <li><a href="#">Something else here</a></li>
        <li role="separator" class="divider"></li>
        <li class="dropdown-header">Nav header</li>
        <li><a href="#">Separated link</a></li>
        <li><a href="#">One more separated link</a></li>
    </ul>
</li>
</ul>
</div><!--/.nav-collapse -->
</div>
</nav>

<h4>Meradlá</h4>

<div class="progress">
    <div class="progress-bar" role="progressbar" aria-valuenow="60" aria-valuemin="0" aria-valuemax=
</div>
<div class="progress">
    <div class="progress-bar progress-bar-success" role="progressbar" aria-valuenow="40" aria-valuem
</div>
<div class="progress">
    <div class="progress-bar progress-bar-info" role="progressbar" aria-valuenow="20" aria-valuemin=
</div>
<div class="progress">
    <div class="progress-bar progress-bar-warning" role="progressbar" aria-valuenow="60" aria-valuem
</div>
<div class="progress">
    <div class="progress-bar progress-bar-danger" role="progressbar" aria-valuenow="80" aria-valuem
</div>
<div class="progress">
    <div class="progress-bar progress-bar-striped" role="progressbar" aria-valuenow="60" aria-valuem
</div>
<div class="progress">
    <div class="progress-bar progress-bar-success" style="width: 35%"><span class="sr-only">35% Comp
    <div class="progress-bar progress-bar-warning" style="width: 20%"><span class="sr-only">20% Comp
    <div class="progress-bar progress-bar-danger" style="width: 10%"><span class='sr-only'>10% Compl
</div>

```

Po stlačení na odkaz Prvá sa vykoná nasledovný kód z ovládača:

```

public ActionResult Prva()
{
    return View();
}

```

MVC test

Domov Prvá Druhá Tretia Štvrtá

Prvá stránka

Tlačidlá

Default
Primary
Success
Info
Warning
Danger
[Link](#)

Tabuľky

#	First Name	Last Name	Username
1	Mark	Otto	mdo
2	Jacob	Thornton	fat
3	Larry	the Bird	twitter

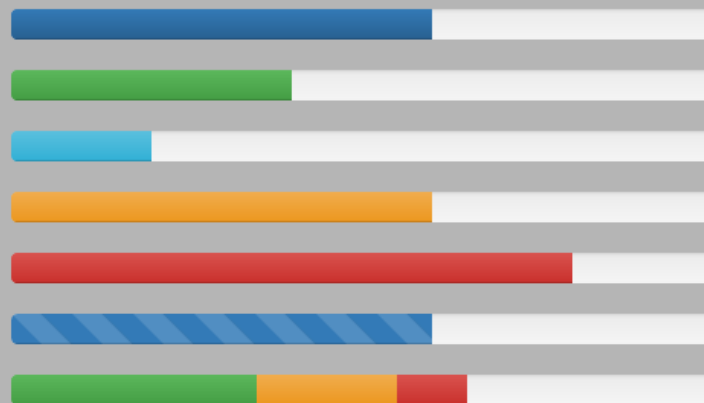
Značky

Default
Primary
Success
Info
Warning
Danger

Menu a navigácia

Project name
Home
About
Contact
Dropdown ▾

Meradlá



Footer

Sidebar

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris vel magna.

- [Link 1](#)
- [Link 2](#)
- [Link 3](#)
- [Link 4](#)
- [Link 5](#)
- [Link 6](#)
- [Link 7](#)
- [Link 8](#)
- [Link 9](#)
- [Link 10](#)
- [Link 11](#)
- [Link 12](#)
- [Link 13](#)
- [Link 14](#)
- [Link 15](#)

Na tomto mieste sme len nechali zobrazit' JavaScriptové prvky z knižnice *Bootstrap*.

4 DRUHÁ STRÁNKA

Druha.cshtml bude vyzerat' nasledovne:

```
<h3>Druhá stránka</h3>

<p>Pozdrav: @ViewBag.Pozdrav</p>
<p>Cislo: @ViewBag.Cislo</p>

@section Sidebar
{
    <p>Špeciálny odkaz:</p>

    <ul>
        <li><a href="/">Link 16</a></li>
    </ul>
}
```

Po stlačení na odkaz Druhá sa vykoná nasledovný kód z ovládača:

```
public ActionResult Druha()
{
    string pozdrav = "Ahoj";
    int cislo = 17;

    ViewBag.Pozdrav = pozdrav;
    ViewBag.Cislo = cislo;

    return View();
}
```


The screenshot shows a web application layout. At the top is a blue header with the text "MVC test". Below it is a black navigation bar with the text "Domov Prvá Druhá Tretia Štvrtá". The main content area is divided into two columns. The left column has a grey background and contains the text "Druhá stránka", "Pozdrav: Ahoj", and "Cislo: 17". The right column has a grey background and contains the text "Sidebar", "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris vel magna.", a list of 15 links labeled "Link 1" through "Link 15", and "Špeciálny odkaz:" followed by a list of one link labeled "Link 16". At the bottom is a blue footer with the text "Footer".

Do kolekcie ViewBag je možné poslať aj viac položiek.

Špeciálne máme na tejto stránke uvedené v časti @section Sidebar nasledovný kód:

```
@section Sidebar
{
    <p>Špeciálny odkaz:</p>

    <ul>
        <li><a href="/">Link 16</a></li>
    </ul>
}
```

Tento úsek kódu zabezpečí zobrazenie doplnkovej informácie na spoločnej šablóne.

5 TRETIA STRÁNKA

Tretia.cshtml bude vyzerat' nasledovne:

```
@using mojeMVC.Models;

<h3>Tretia stránka</h3>

<h4>Výpis kníh</h4>

@foreach (var b in Model)
{
    <p>Id: @b.Id</p>
    <p>Názov: @b.Name</p>
    <p>Autor: @b.Author</p>
}

<p>Dátum: @ViewBag.Datum</p>
<br />
<button type="button" class="btn btn-sm btn-default" onclick="ButtonClick()">Zmena farby pozadia</button>

<script type="text/javascript">

    function ButtonClick()
    {
        $("div").css({ "background-color": "blue" });
        $(document).ready(function () {
            $("p").css({ "color": "#000" });
        });
    }

</script>
```

Teda, po stlačení na odkaz Tretia sa vykoná nasledovný kód z ovládača:

```
public ActionResult Tretia(int id = 2)
{
    var b = from Book book
            in Book.GetBooks()
            where book.Id == id
            select book;

    ViewBag.Datum = DateTime.Now;

    return View(b);
}
```

The screenshot shows a web application with a blue header containing the text "MVC test". Below the header is a black navigation bar with the text "Domov Prvá Druhá Tretia Štvrtá". The main content area is divided into two columns. The left column has a grey background and contains the text "Tretia stránka", "Výpis kníh", "Id: 2", "Názov: Nazov_2", "Autor: Autor_2", and "Dátum: 20-May-18 16:04:36". Below this text is a button labeled "Zmena farby pozadia". The right column has a grey background and contains the text "Sidebar", "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris vel magna.", and a list of 15 links labeled "Link 1" through "Link 15". At the bottom of the page is a blue footer containing the text "Footer".

Na tomto mieste sme požadovali v metóde `Tretia()` parameter s konkrétnym `Id` knihy s hodnotou 2. To je možné napísať ako parameter tejto funkcie a v samotnom tele použijeme túto hodnotu v podmienke LINQ dopytu v uvedenej forme. Bližšie nešpecifikovaná (dynamicky typovaná) premenná `b` bude obsahovať jednu knihu v tvare zoznamu. Obsah tohto zoznamu zobrazíme na pohľadovej stránke nasledovne:

```
@foreach (var b in Model)
{
    <p>Id: @b.Id</p>
    <p>Názov: @b.Name</p>
    <p>Autor: @b.Author</p>
}
```

Metóda je napísaná všeobecne pre výpis všetkých kníh. Túto premennú `b` predáme do pohľadu `View()` ako parameter `View(b)`. Následne je možné súčasne použiť aj pôvodnú kolekciu `ViewBag` na ktorú sa odkazujeme prostredníctvom `@ViewBag.Datum` a pre konkrétny výpis danej položky knihy sa odkazujeme v uvedenom cykle s predtým špecifikovaným using odkazom `@using mojeMVC.Models;`

Na pohľadovej stránke, ako už bolo uvedené, je možné písať JavaScriptový kód v jednej z nasledujúcich podôb:

```
<script type="text/javascript">

    function ButtonClick()
    {
        $("div").css({ "background-color": "blue" });
        $(document).ready(function () {
            $("p").css({ "color": "#000" });
        });
    }

</script>
```

V našom prípade sme vytvorili jednoduchú funkciu na zmeny pozadia spoločnej šablóny.

6 ŠTVRTÁ STRÁNKA

Stvrta.cshtml bude vyzerat' nasledovne:

```
@using mojeMVC.Models;
```

```
<h3>Štvrtá stránka</h3>
```

```
<table class="table table-bordered table-hover table-condensed table-responsive">
  <thead>
    <tr>
      <th>Id</th>
      <th>Názov</th>
      <th>Autor</th>
    </tr>
  </thead>
  <tbody>
    @foreach (Book b in ((List<Book>)ViewBag.ZoznamKnih))
    {
      <tr>
        <td>@b.Id</td>
        <td>@b.Name</td>
        <td>@b.Author</td>
      </tr>
    }
  </tbody>
</table>
```

Teda, po stlačení na odkaz Štvrtá sa vykoná nasledovný kód z ovládača:

```
public ActionResult Stvrta ()
{
    List<Book> zoznamKnih = Book.GetBooks ();

    ViewBag.ZoznamKnih = zoznamKnih;

    return View ();
}
```

MVC test

Domov Prvá Druhá Tretia Štvrtá

Štvrtá stránka

Id	Názov	Autor
1	Nazov_1	Autor_1
2	Nazov_2	Autor_2
3	Nazov_3	Autor_3

Sidebar

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris vel magna.

- [Link 1](#)
- [Link 2](#)
- [Link 3](#)
- [Link 4](#)
- [Link 5](#)
- [Link 6](#)
- [Link 7](#)
- [Link 8](#)
- [Link 9](#)
- [Link 10](#)
- [Link 11](#)
- [Link 12](#)
- [Link 13](#)
- [Link 14](#)
- [Link 15](#)

Footer

Tento spôsob zobrazenia je podobný predchádzajúcemu spôsobu s tým rozdielom, že vypisujeme celý zoznam kníh priamo do HTML tabuľky prostredníctvom kolekcie ViewBack.