

Kreiranje MVC pogleda

MVC pogled

- Kontroler prosleđuje podatke pogledu korišćenjem View metode
- Pogledi pridruženi kontroleru pod nazivom HomeController nalaze se u folderu Views/Home
- Ukoliko želimo da isti pogled pridružimo za dva ili više kontrolera stavljamo ga u folder Views/Shared

ASP.NET Core Web aplikacija

Create a new ASP.NET Core web application

.NET Core ASP.NET Core 3.1

- Empty**
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.
- API**
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.
- Web Application**
A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.
- Web Application (Model-View-Controller)**
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.
- Angular**
A project template for creating an ASP.NET Core application with Angular
- React.js**
A project template for creating an ASP.NET Core application with React.js

[Get additional project templates](#)

Authentication
No Authentication
[Change](#)

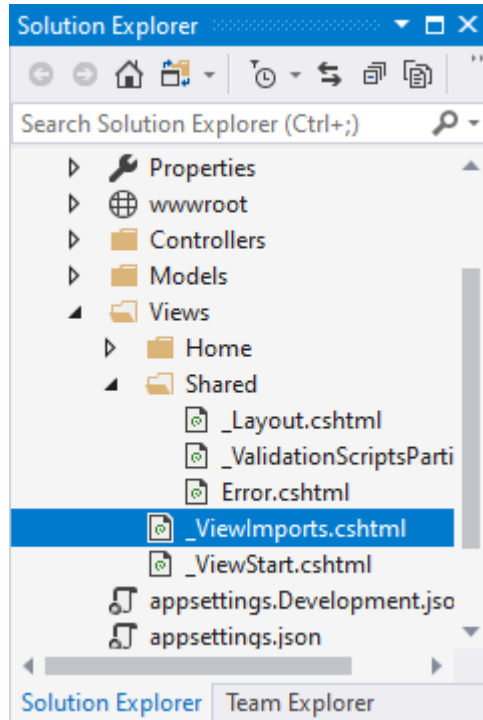
Advanced
 Configure for HTTPS
 Enable Docker Support
(Requires Docker Desktop)

Linux

Author: Microsoft
Source: .NET Core 3.1.1

[Back](#) [Create](#)

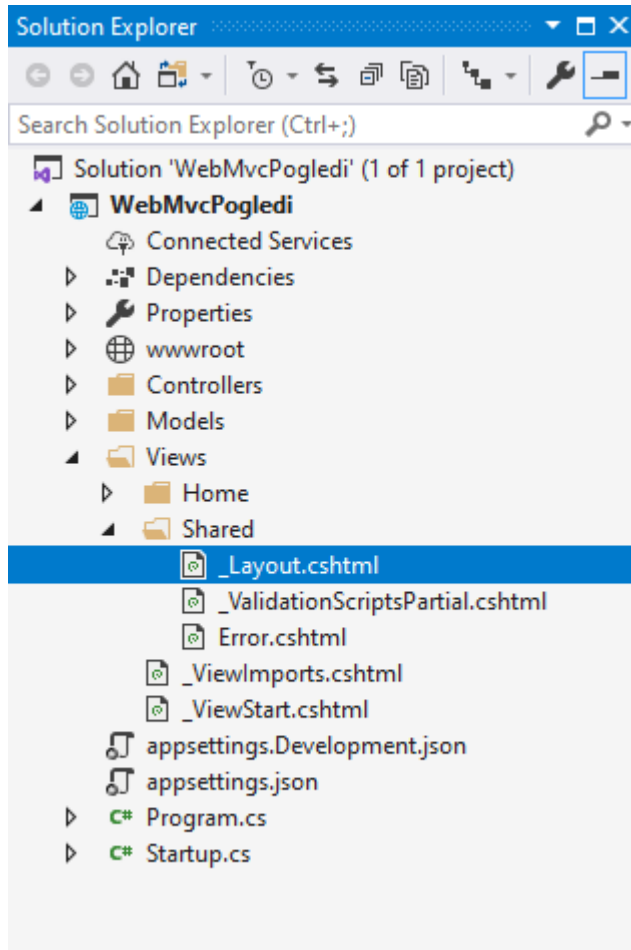
Fajl _ViewImports.cshtml



```
@using WebMvcPogledi  
@using WebMvcPogledi.Models  
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
```

Direktive koje se dele između različitih pogleda smeštaju se unutar fajla _ViewImports.cshtml

Pogled _Layout.cshtml

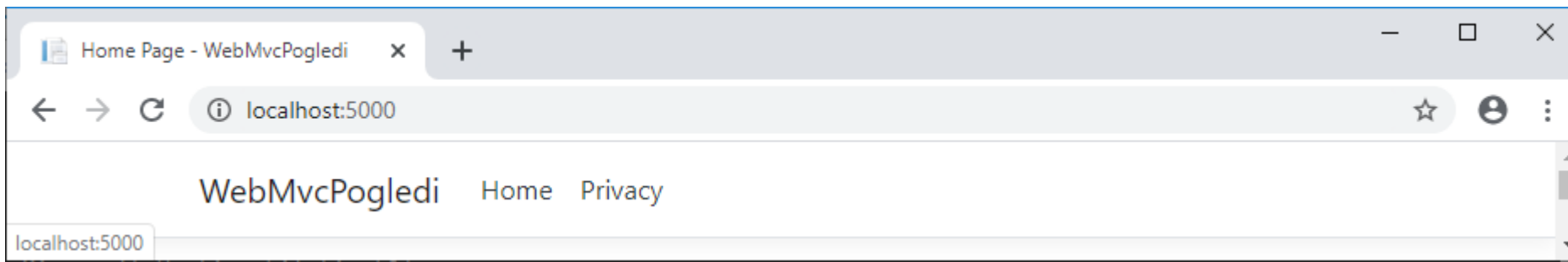


_Layout.cshtml – head deo

```
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>@ViewData["Title"] - WebMvcPogledi</title>
  <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
  <link rel="stylesheet" href="~/css/site.css" />
</head>
```

_Layout.cshtml – body (header)

```
<header>
  <nav class="navbar navbar-expand-sm navbar-togglerable-sm navbar-light bg-white border-bottom box-shadow mb-3">
    <div class="container">
      <a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">WebMvcPogledi</a>
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target=".navbar-collapse" aria-
controls="navbarSupportedContent"
        aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="navbar-collapse collapse d-sm-inline-flex flex-sm-row-reverse">
        <ul class="navbar-nav flex-grow-1">
          <li class="nav-item">
            <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
</header>
```



_Layout.cshtml – glavna oblast strane

- Može biti samo jedna @RenderBody() metoda po layout strani
- Pomoću RenderBody() metode specificira se deo layout strane koji će biti popunjen sadržajem strane koja implementira layout stranu

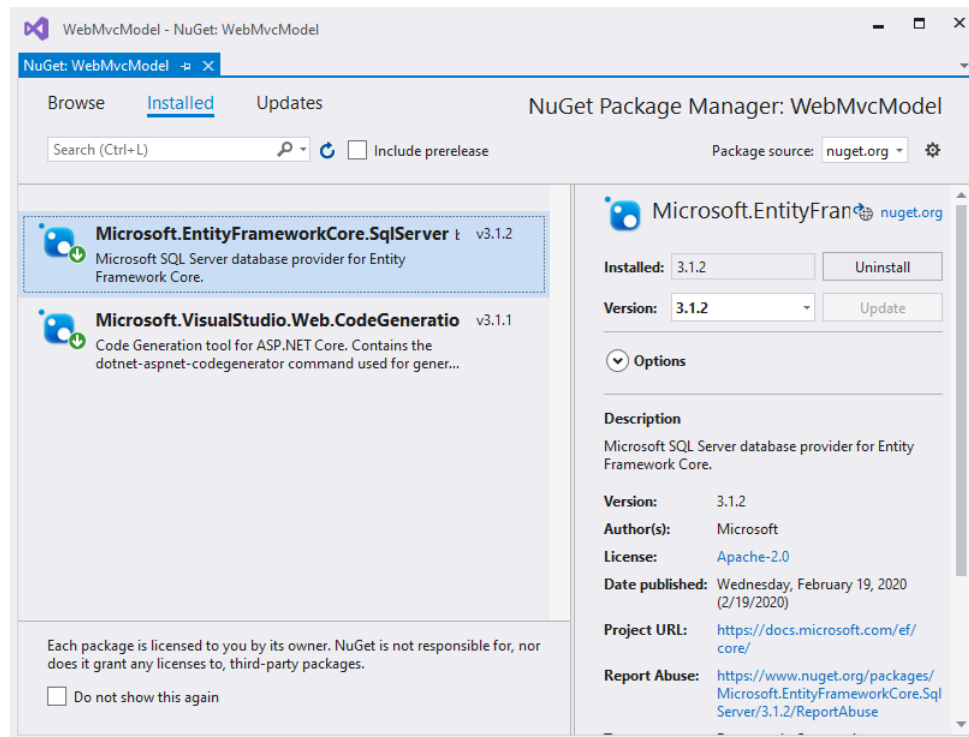
```
<div class="container">  
  <main role="main" class="pb-3">  
    @RenderBody()  
  </main>  
</div>
```


_Layout.cshtml – footer

- Metoda RenderSection() služi za definisanje sekcije na strani koja implementira layout stranu
- Sekciju moramo definisati na strani koja implementira layout osim ako ne napišemo **required: false**

```
<footer class="border-top footer text-muted">
  <div class="container">
    &copy; 2020 - WebMvcPogledi - <a asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>
  </div>
</footer>
<script src="~/lib/jquery/dist/jquery.min.js"></script>
<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
<script src="~/js/site.js" asp-append-version="true"></script>
@RenderSection("Scripts", required: false)
```

Microsoft.EntityFrameworkCore.SqlServer



Microsoft.EntityFrameworkCore.Tools

The screenshot shows the NuGet Package Manager window for a project named 'WebMvcModel'. The search bar contains 'Microsoft.EntityFrameworkCore.Tools'. The package list on the left shows three results, with the first one selected:

Package Name	Author	Downloads	Version
Microsoft.EntityFrameworkCore.Tools	Microsoft	42.5M	v3.1.2
Microsoft.EntityFrameworkCore.Design	Microsoft	51.5M	v3.1.2
Microsoft.EntityFrameworkCore.Tools.DotNet	Microsoft	4.64M	v2.0.3

The details pane on the right for the selected package shows:

- Package Name: Microsoft.EntityFrameworkCore.Tools
- Version: 3.1.2
- Install button
- Options section (expanded)
- Description: Entity Framework Core Tools for the NuGet Package Manager Console in Visual Studio.
- Enables these commonly used commands: Add-Migration, Drop-Database, Get-DbContext

At the bottom of the package list, there is a checkbox for 'Do not show this again'.

Model klasa

```
[Table("Osoba")]
public class Osoba
{
    public int OsobaId { get; set; }

    [Display(Name = "Ime osobe")]
    [Required(ErrorMessage = "Morate uneti ime")]
    [StringLength(30, ErrorMessage = "Maksimalno 30 karaktera")]
    public string Ime { get; set; }

    [Required(ErrorMessage = "Morate uneti prezime")]
    [Display(Name = "Prezime osobe")]
    [StringLength(30, ErrorMessage = "Maksimalno 30 karaktera")]
    public string Prezime { get; set; }

    [Required(ErrorMessage = "Morate uneti adresu")]
    [DataType(DataType.MultilineText)]
    [StringLength(100, ErrorMessage = "Maksimalno 100 karaktera")]
    public string Adresa { get; set; }
}
```

Fajl appsettings.json

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "DefaultConnection": "Server=(local)\\SqlExpress;Database=OsobaDb1603;Trusted_Connection=True"
  }
}
```

DbContext klasa

```
namespace WebMvcPogledi.Models
{
    public class OsobaContext : DbContext
    {
        public virtual DbSet<Osoba> Osobe { get; set; }
    }
}
```

Registrujemo DbContext kao servis

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddDbContext<OsobaContext>(opcije =>
        opcije.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));

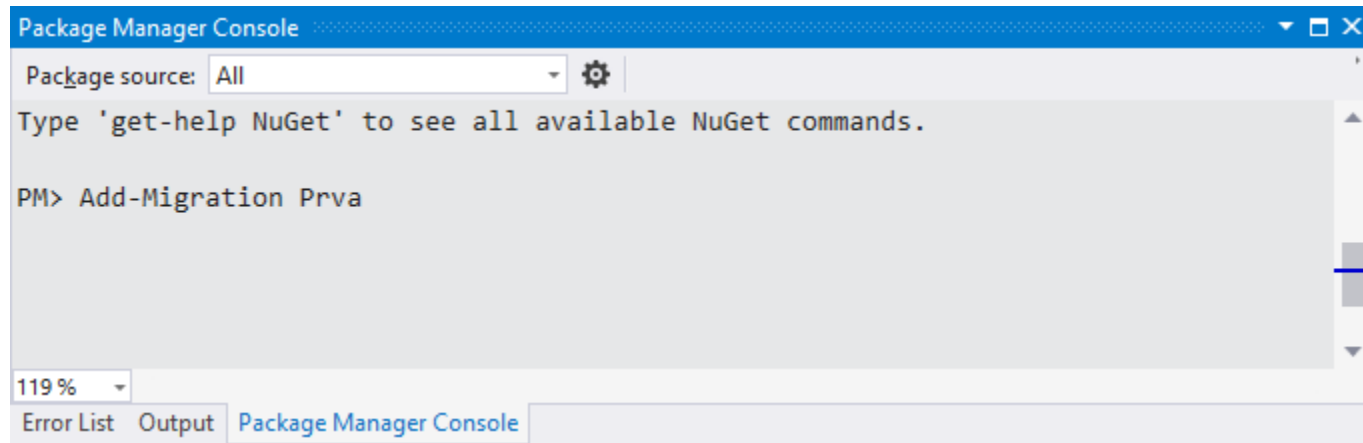
    services.AddControllersWithViews();
}
```

Izmenjena DbContext klasa

```
using Microsoft.EntityFrameworkCore;
```

```
namespace WebMvcKontroler.Models
{
    public class OsobaContext : DbContext
    {
        public OsobaContext(DbContextOptions<OsobaContext> opcije) : base(opcije)
        {
        }
        public DbSet<Osoba> Osobe { get; set; }
    }
}
```


Kreiranje baze



Package Manager Console

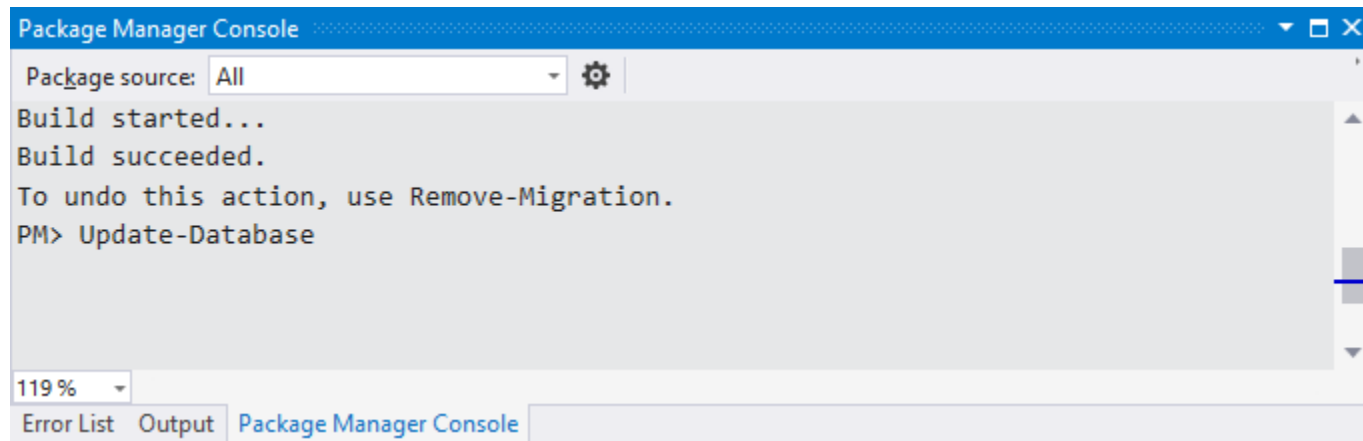
Package source: All

Type 'get-help NuGet' to see all available NuGet commands.

```
PM> Add-Migration Prva
```

119 %

Error List Output Package Manager Console



Package Manager Console

Package source: All

Build started...

Build succeeded.

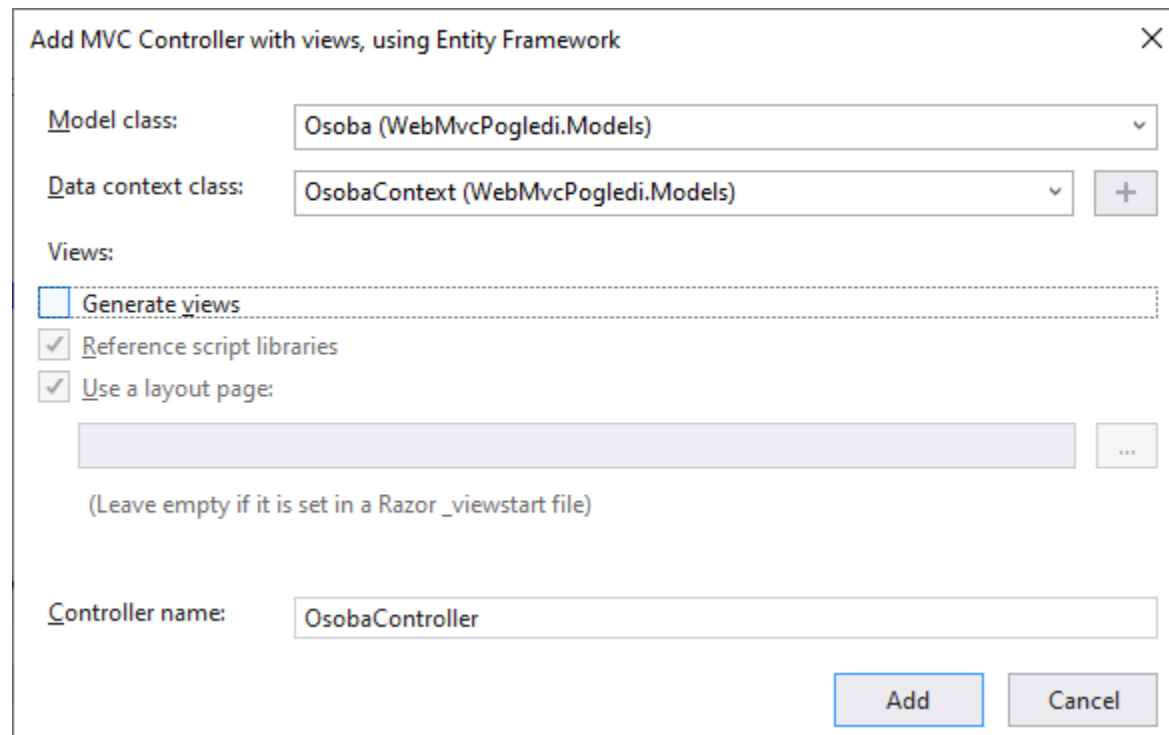
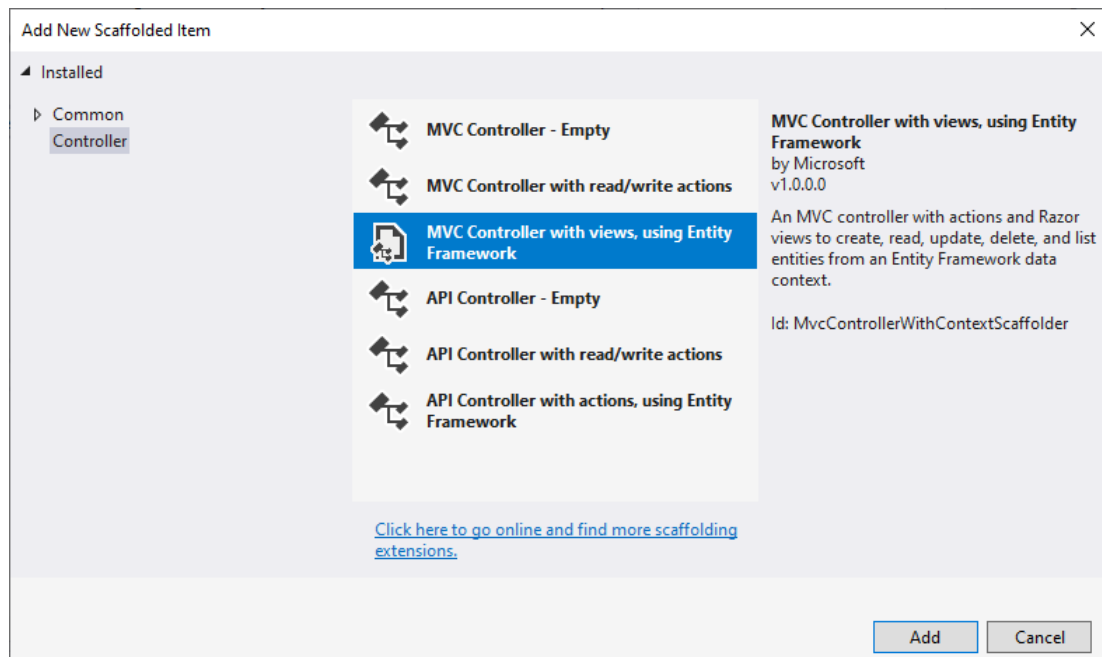
To undo this action, use Remove-Migration.

```
PM> Update-Database
```

119 %

Error List Output Package Manager Console

Kreiranje kontrolera bez pogleda



Kreiranje Index pogleda

Add MVC View ✕

View name:

Template: ▾

Model class: ▾

Data context class: ▾

Options:

Create as a partial view

Reference script libraries

Use a layout page:

...

(Leave empty if it is set in a Razor _viewstart file)

Index.cshtml - 1.deo

```
@model IEnumerable<WebMvcPogledi.Models.Osoba>

@{
    ViewData["Title"] = "Index";
}

<h2>Index</h2>

<p>
    <a asp-action="Create">Create New</a>
</p>
<table class="table">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.Ime)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Prezime)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Adresa)
            </th>
            <th></th>
        </tr>
    </thead>
    <tbody>
```

Index.cshtml - 2.deo

```
@foreach (var item in Model) {  
    <tr>  
        <td>  
            @Html.DisplayFor(modelItem => item.Ime)  
        </td>  
        <td>  
            @Html.DisplayFor(modelItem => item.Prezime)  
        </td>  
        <td>  
            @Html.DisplayFor(modelItem => item.Adresa)  
        </td>  
        <td>  
            <a asp-action="Edit" asp-route-id="@item.OsobaId">Edit</a> |  
            <a asp-action="Details" asp-route-id="@item.OsobaId">Details</a> |  
            <a asp-action="Delete" asp-route-id="@item.OsobaId">Delete</a>  
        </td>  
    </tr>  
}  
</tbody>  
</table>
```

Index pogled

WebMvcPogledi Home Privacy

Index

[Create New](#)

Ime osobe	Prezime osobe	Adresa	
Marko	Markovic	Glavna 5	Edit Details Delete
Ivan	Petrovic	Beogradska 7	Edit Details Delete

© 2020 - WebMvcPogledi - [Privacy](#)

Anchor Tag helper

- Tag helper omogućava korišćenje serverskog koda u kreiranju i renderovanju HTML elemenata u Razor fajlovima
- Većina ugrađenih tag helpera se odnosi na postojeće HTML elemente i obezbeđuju serverske attribute za element
- Možemo kreirati i sopstvene tag helpere
- Anchor tag helper dodaje nove attribute elementu `<a>...`
 - Atribut **asp-controller** specificira kontroler koji se koristi za generisanje URL adrese
 - Ako se ne specificira kontroler, onda se zove podrazumevani kontroler koji poziva pogled
 - Atribut **asp-action** specificira metodu kontrolera koja se koristi za generisanje URL adrese
 - Atribut **asp-route**- bilo koju vrednost koji prosledimo nakon crtice tretira se kao parametar rute, npr. `asp-route-id`

HtmlHelper klasa metoda DisplayNameFor()

- Klasa HtmlHelper koristi se za renderovanje html kontrola u pogledu
- Html property pogleda služi da pristupimo instanci klase HtmlHelper
- Metoda DisplayNameFor() prikazuje ime svojstva odgovarajućeg člana model klase
- Prikazani tekst zavisi od modela, tj. da li se koristi anotacija [Display(Name="")] prilikom definisanja polja klase

```
<th>  
    @Html.DisplayNameFor(model => model.Ime)  
</th>
```

```
<th>  
    Ime osobe  
</th>
```


Metoda DisplayFor() HtmlHelper klase

- Prikazuje vrednost određenog svojstva objekta model klase
- Ako je definisan [DisplayFormat] u zavisnosti od toga će biti prikazan npr. kratki format datuma

```
<td>  
    @Html.DisplayFor(modelItem => modelItem.Adresa)  
</td>
```

```
<td>  
    Glavna 5  
</td>
```

Modifikovani Index pogled

```
@model IEnumerable<Osoba>

@{
    ViewData["Title"] = "Index";
}

<h2>Index</h2>
<p>
    <a asp-action="Create">Nova osoba</a>
</p>
<table class="table">
    <thead>
        <tr>
            <th>
                Ime
            </th>
            <th>
                Prezime
            </th>
            <th>
                Adresa
            </th>
        </tr>
    </thead>
    <tbody>
```

```
@foreach (Osoba os in Model) {
    <tr>
        <td>
            @os.Ime
        </td>
        <td>
            @os.Prezime
        </td>
        <td>
            @os.Adresa
        </td>
        <td>
            <a asp-action="Edit" asp-route-id="@os.OsobaId">Promeni</a> |
            <a asp-action="Details" asp-route-id="@os.OsobaId">Detalji</a> |
            <a asp-action="Delete" asp-route-id="@os.OsobaId">Obrisi</a>
        </td>
    </tr>
}
</tbody>
</table>
</table>
```

Kreiranje pogleda Create

Add MVC View ✕

View name:

Template: ▾

Model class: ▾

Data context class: ▾

Options:

Create as a partial view

Reference script libraries

Use a layout page:

...

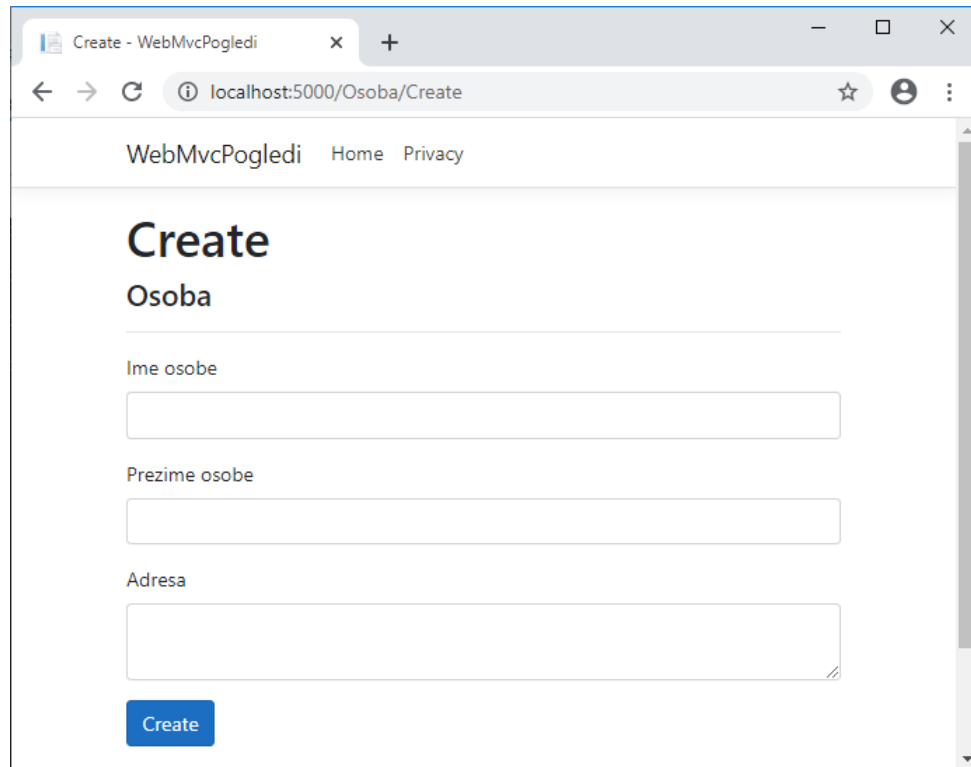
(Leave empty if it is set in a Razor _viewstart file)

Deo pogleda Create.cshtml

@model Osoba

```
<div class="row">
  <div class="col-md-4">
    <form asp-action="Create">
      <div asp-validation-summary="ModelOnly" class="text-danger"></div>
      <div class="form-group">
        <label asp-for="Ime" class="control-label"></label>
        <input asp-for="Ime" class="form-control" />
        <span asp-validation-for="Ime" class="text-danger"></span>
      </div>
      <div class="form-group">
        <label asp-for="Prezime" class="control-label"></label>
        <input asp-for="Prezime" class="form-control" />
        <span asp-validation-for="Prezime" class="text-danger"></span>
      </div>
      <div class="form-group">
        <label asp-for="Adresa" class="control-label"></label>
        <textarea asp-for="Adresa" class="form-control"></textarea>
        <span asp-validation-for="Adresa" class="text-danger"></span>
      </div>
      <div class="form-group">
        <input type="submit" value="Create" class="btn btn-primary" />
      </div>
    </form>
  </div>
</div>
```

Pogled Create



The screenshot shows a web browser window with the following elements:

- Browser Tab:** Create - WebMvcPogledi
- Address Bar:** localhost:5000/Osoba/Create
- Navigation:** Back, Forward, Refresh, Home, Privacy
- Page Title:** WebMvcPogledi
- Section Header:** Create
- Form Title:** Osoba
- Form Fields:**
 - Ime osobe (Text input)
 - Prezime osobe (Text input)
 - Adresa (Text area)
- Action:** Create button

Form tag helper

- Generiše action atribut forme na osnovu imena kontrolera i imena akcione metode
- Generiše skriveni Request Verification Token da bi se sprečilo falsifikovanje zahteva
- POST metoda kontrolera proverava postojanje ovog tokena [ValidateAntiForgeryToken]
- Atribut **asp-controller** specificira kontroler koji se koristi za generisanje URL adrese
- Ako se ne specificira kontroler, onda se zove podrazumevani kontroler koji poziva pogled
- Atribut **asp-action tag** specificira metodu kontrolera koja se koristi za generisanje URL adrese

```
<form asp-action="Create">  
</form>
```

Generisani html od strane form tag helpera

```
<form action="/Osoba/Create" method="post">  
  <input name="__RequestVerificationToken" type="hidden"  
value="CfDJ8HzTg8ZoCT1JsNpvrgaABFy4sGtcAm5iFe8TiOFx3gvFRxMG4uygaeXkuF45zFxH1H-  
pIWSQuc68cELsqvbysvM6zr__V2XQrCmZlbsvIN2091VVM6gMWvP4P_hbIPs2pKVxxEF0jGpBJ6  
Mnikahtc8" />  
</form>
```

Label Tag Helper

- Generiše <label> element za svojstvo modela
- Tag asp-for specificira svojstvo modela
- Generiše vrednost za for atribut

```
<label asp-for="Ime" class="control-label"></label>
```

Generisani HTML:

```
<label class="control-label" for="Ime">Ime osobe</label>
```


Input Tag Helper

- Binduje HTML `<input>` element za polje modela u razor pogledu
- Setuje atribut `type` generisanog input elementa u zavisnosti od tipa polja modela
- Generisani `type` atribut može se prebrisati ako ga ponovo definišemo za input element
- Generiše validacione attribute `data-val-` iz anotacionih atributa za polja klasa modela
 - `data-val`
 - `data-val-required`
 - `data-val-length-max`
 -

```
<input asp-for="Ime" class="form-control" />
```

Generisani html od strane input tag helpera

```
<input class="form-control" type="text" data-val="true" data-val-length="Maksimalno 30 karaktera" data-val-length-max="30" data-val-required="Morate uneti ime" id="Ime" name="Ime" value="" />
```

Ubacivanje validacionih biblioteka u Razor pogled:

```
@section Scripts {  
    @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}  
}
```

_ValidationScriptsPartial.cshtml

```
<script src="~/lib/jquery-validation/dist/jquery.validate.min.js"></script>  
<script src="~/lib/jquery-validation-  
unobtrusive/jquery.validate.unobtrusive.min.js"></script>
```

Atributi za validaciju su definisani unutar biblioteke jquery.validate.unobtrusive.js

Validation Message Tag Helper

- Dodaje data-valmsg-for="svojstvo" atribut u span element
- Ovaj atribut prikazuje validacionu grešku unutar span elementa za određeno svojstvo
- Prikazana greška je rezultat klijentske validacije
- Validacija se takođe izvršava i na serveru

```
<span asp-validation-for="Ime" class="text-danger"></span>
```

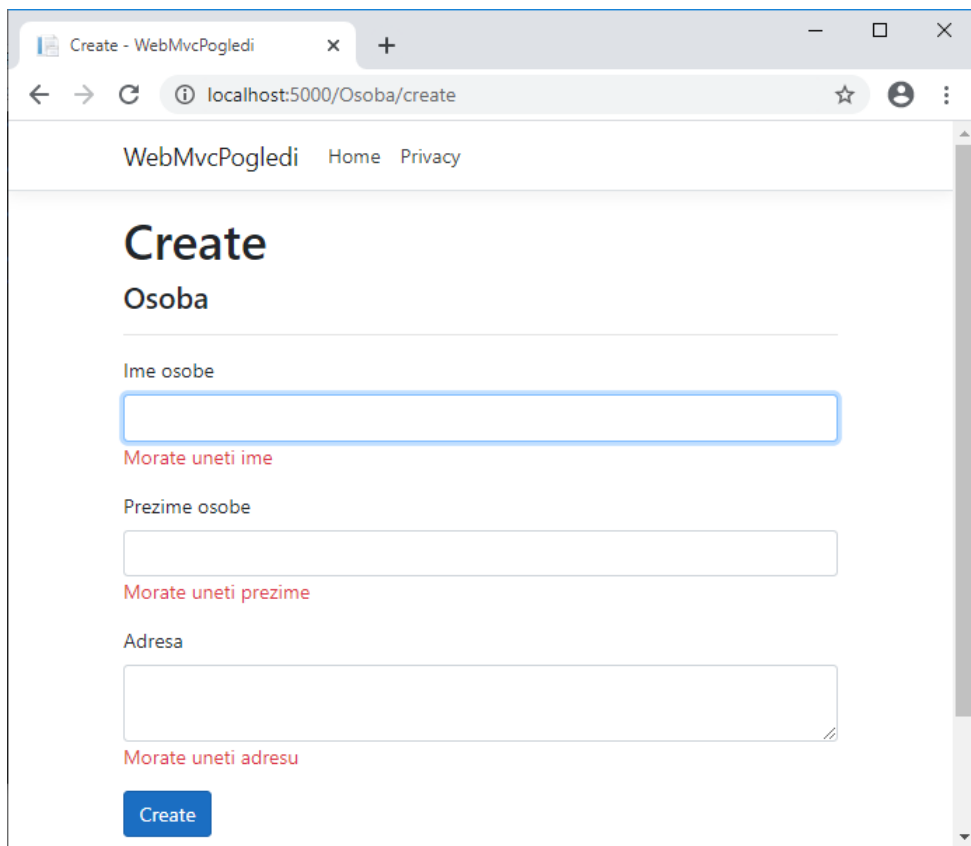
```
<span class="text-danger field-validation-valid" data-valmsg-for="Ime" data-valmsg-replace="true"></span>
```

Validation Summary Tag Helper

- Dodaje atribut `asp-validation-summary` u `div` element
- Vrednost ovog atributa može biti:
 - `ValidationSummary.All` - prikazuju se validacione greške na nivou svojstva i na nivou modela
 - `ValidationSummary.ModelOnly` - prikazuju se validacione greške samo na nivou modela
 - `ValidationSummary.None`

```
<div asp-validation-summary="ModelOnly" class="text-danger"></div>
```

Validacione greške na nivou modela

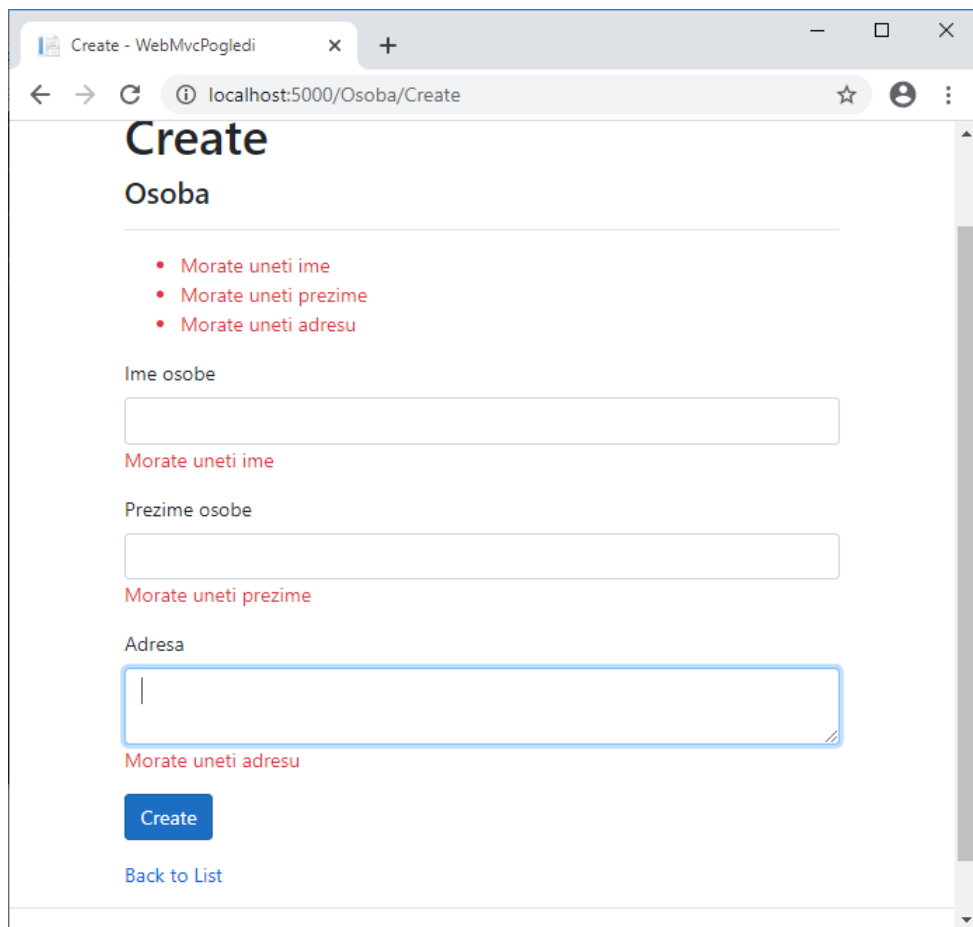


The screenshot shows a web browser window with the URL `localhost:5000/Osoba/create`. The page title is "WebMvcPogledi" and it has navigation links for "Home" and "Privacy". The main heading is "Create Osoba". There are three input fields: "Ime osobe", "Prezime osobe", and "Adresa". Each field has a red error message below it: "Morate uneti ime", "Morate uneti prezime", and "Morate uneti adresu". A blue "Create" button is at the bottom left.

```
<div asp-validation-summary="ModelOnly" class="text-danger"></div>
```

Prikaz svih validacionih grešaka

```
<div asp-validation-summary="All" class="text-danger"></div>
```



The screenshot shows a web browser window with the address bar displaying 'localhost:5000/Osoba/Create'. The page title is 'Create' and the subtitle is 'Osoba'. Below the subtitle, there is a list of validation errors in red text:

- Morate uneti ime
- Morate uneti prezime
- Morate uneti adresu

Below the errors, there are three input fields:

- 'Ime osobe' with an empty text input field and a red error message 'Morate uneti ime' below it.
- 'Prezime osobe' with an empty text input field and a red error message 'Morate uneti prezime' below it.
- 'Adresa' with an empty text input field and a red error message 'Morate uneti adresu' below it.

At the bottom of the form, there is a blue 'Create' button and a blue link 'Back to List'.

Kreiranje Edit pogleda

Add MVC View ✕

View name:

Template: ▾

Model class: ▾

Data context class: ▾

Options:

Create as a partial view

Reference script libraries

Use a layout page:

...

(Leave empty if it is set in a Razor `_viewstart` file)

Deo Edit pogleda

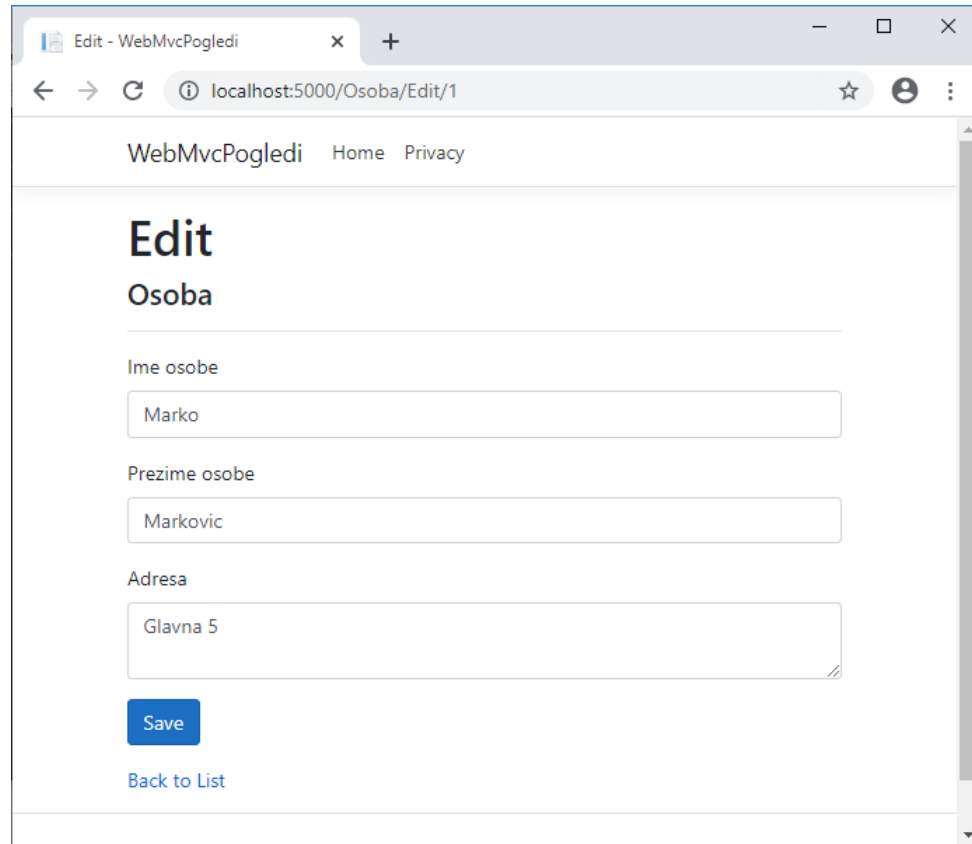
```
<form asp-action="Edit">
    <div asp-validation-summary="ModelOnly" class="text-danger"></div>
    <input type="hidden" asp-for="OsobaId" />

    <div class="form-group">
        <label asp-for="Ime" class="control-label"></label>
        <input asp-for="Ime" class="form-control" />
        <span asp-validation-for="Ime" class="text-danger"></span>
    </div>
```

Kreiranje skrivenog polja kojim se prosleđuje OsobaId polje metodi kontrolera

```
<input type="hidden" data-val="true" data-val-required="The OsobaId field is required." id="OsobaId"
name="OsobaId" value="1" />
```

Izgled strane za editovanje



WebMvcPogledi Home Privacy

Edit

Osoba

Ime osobe

Prezime osobe

Adresa

[Save](#)

[Back to List](#)

Kreiranje Delete pogleda

Add MVC View ×

View name:

Template:

Model class:

Data context class:

Options:

Create as a partial view

Reference script libraries

Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)

Delete.cshtml-1

```
@model WebMvcPogledi.Models.Osoba
```

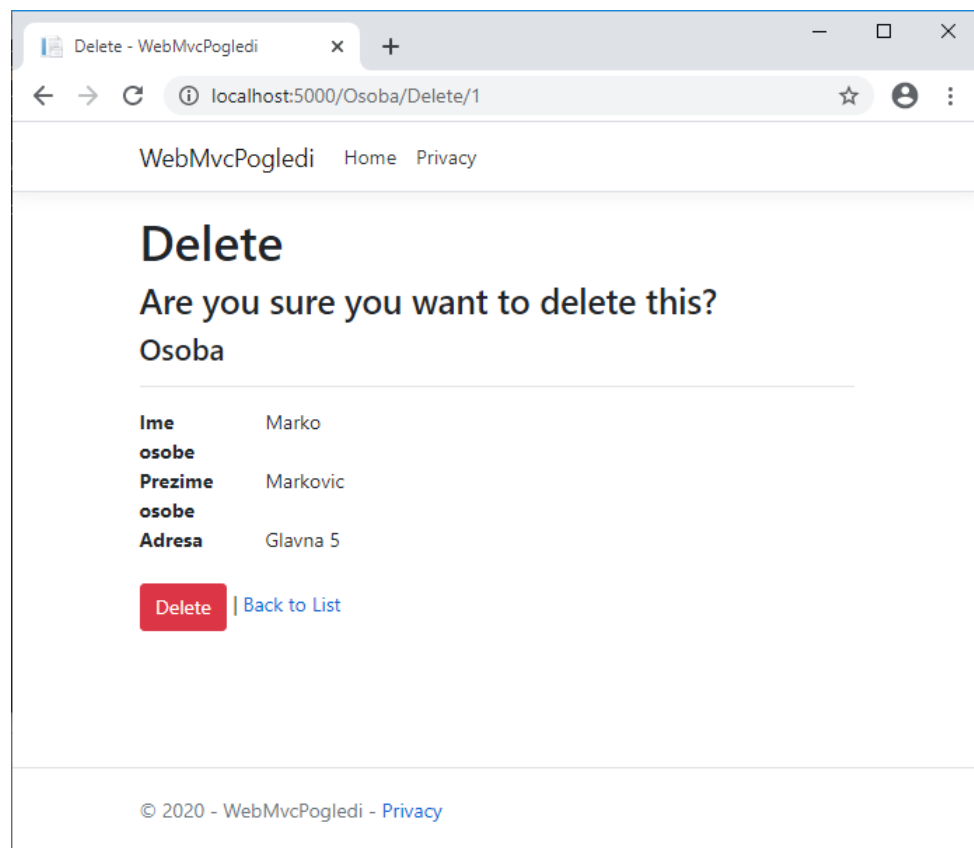
```
@{  
    ViewData["Title"] = "Delete";  
}
```

```
<dl class="row">  
    <dt class = "col-sm-2">  
        @Html.DisplayNameFor(model => model.Ime)  
    </dt>  
    <dd class = "col-sm-10">  
        @Html.DisplayFor(model => model.Ime)  
    </dd>  
    <dt class = "col-sm-2">  
        @Html.DisplayNameFor(model => model.Prezime)  
    </dt>  
    <dd class = "col-sm-10">  
        @Html.DisplayFor(model => model.Prezime)  
    </dd>  
    <dt class = "col-sm-2">  
        @Html.DisplayNameFor(model => model.Adresa)  
    </dt>  
    <dd class = "col-sm-10">  
        @Html.DisplayFor(model => model.Adresa)  
    </dd>  
</dl>
```

Delete.cshtml-2

```
<form asp-action="Delete">
    <input type="hidden" asp-for="OsobaId" />
    <input type="submit" value="Delete" class="btn btn-danger" /> |
    <a asp-action="Index">Back to List</a>
</form>
```

Strana za brisanje



Pitanje 1

Ukoliko želimo da isti pogled pridružimo za dva ili više kontrolera stavljamo ga u folder:

- a. Views/Home
- b. Views/Shared
- c. Views/Index

Odgovor: b

Pitanje 2

Index pogled koji odgovara kontroleru OsobaController je fajl:

- a. `\Views\Osoba\Index.cshtml`
- b. `\Views\OsobaController\Index.cshtml`
- c. `\Views\Index.cshtml`
- d. `\Models\Osoba\Index.cshtml`

Odgovor: a

Pitanje 3

Html helper metoda **DisplayFor()** prikazuje:

- a. Vrednost svojstva model klase za objekat koji trenutno prikazuje
- b. Naziv svojstva iz model klase
- c. generiše element <form>

Odgovor: a

Pitanje 4

Html Helper metoda **DisplayNameFor()** prikazuje:

- a. Vrednost svojstva model klase za objekat koji trenutno prikazuje
- b. Naziv svojstva iz model klase
- c. generiše element <form>

Odgovor: b

Pitanje 5

Direktive koje se dele između različitih pogleda smeštaju se unuta sledećeg fajla Views/Shared foldera

- a. `_ViewImports.cshtml`
- b. `Imports.cshtml`
- c. `_Layout.cshtml`

Odgovor: a

Pitanje 6

Klase koje omogućavaju serverskom kodu da ubacuje serverske attribute u postojeće html elemente i proširi njihovu funkcionalnost nazivaju se:

- a. tag helperi
- b. html helperi
- c. code helperi

Odgovor: a

Pitanje 7

Kako InputTagHelper specificira svojstvo modela za koje se kreira input kontrola

- a. pomoću atributa asp-for
- b. pomoću atributa type
- c. pomoću atributa

Odgovor: a