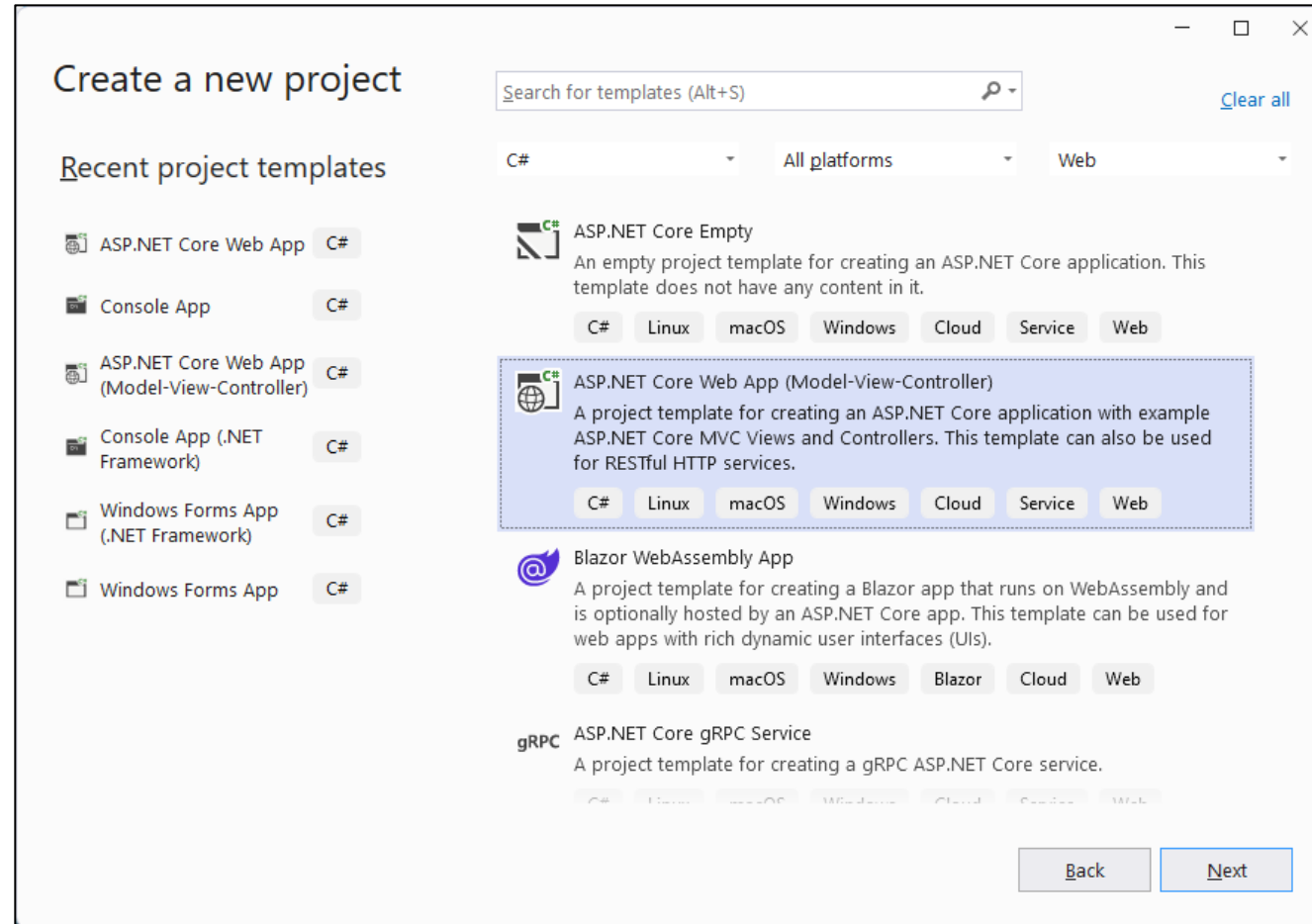


MVC modeli

MVC Model

- Model predstavlja domen aplikacije
- Svrha modela je da da smisao podacima
- Model se može koristiti kao kontejner za podatke koji se prikazuju na web strani
- Najčešće se kao model podataka koristi entitetski model

Kreiranje ASP.NET Core web aplikacije



Klasa kao model – folder Models

```
using System.ComponentModel.DataAnnotations.Schema;
```

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

Interfejs Iskladiste

```
namespace WebApp03.Models
{
    public interface Iskladiste
    {
        IEnumerable<Osoba> Osobe { get; }
    }
}
```

Models folder

```
public class OsobaSkladiste : Iskladiste
{
    public IEnumerable<Osoba> Osobe
    {
        get
        {
            List<Osoba> listaOsoba = new List<Osoba>()
            {
                new Osoba { OsobaId = 1, Ime = "Pera", Prezime = "Peric", Adresa = "Prvomajska 4" },
                new Osoba { OsobaId = 2, Ime = "Mika", Prezime = "Mikic", Adresa = "Glavna 12" },
                new Osoba { OsobaId = 3, Ime = "Laza", Prezime = "Lazic", Adresa = "Marije Bursac 2" }
            };
            return listaOsoba;
        }
    }
}
```

Dodavanje servisa u kontejner

```
using WebApp03.Models;  
  
var builder = WebApplication.CreateBuilder(args);  
  
// Add services to the container.  
builder.Services.AddControllersWithViews();  
builder.Services.AddTransient<Iskladiste, OsobaSkladiste>();  
  
var app = builder.Build();
```

Kada komponenta poput kontrolera ima potrebe za implementacijom `Iskladiste` interfejsa ona će dobiti instancu klase `OsobaSkladiste`. Tranzientni servis se instancira pri svakom zahtevu za implementacijom (posredstvom konstruktora zavisne klase).

Modifikovana klasa HomeController

```
public class HomeController : Controller
{
    private readonly Iskladiste _skladiste;

    public HomeController(Iskladiste skladiste)
    {
        _skladiste = skladiste;
    }

    public IActionResult Index()
    {
        ViewBag.Poruka = "Lista osoba";

        return View(_skladiste.Osobe);
    }
}
```


Ubacivanje zavisnosti u kontroler

- Kada DI kontejner ASP.NET Core aplikacije treba da kreira novu instancu klase HomeController da bi obradio HTTP zahtev on gleda konstruktor i uviđa da on zahteva objekat koji implementira interfejs Iskladiste
- Da bi odredio koju implementacionu klasu treba da koristi DI kontejner konsultuje Program klasu gde su registrovani servisi
- Kontejner servisa zna da za implementaciju interfejsa Iskladiste treba koristiti klasu OsobaSkladiste i da se pri svakom zahtevu treba kreirati nova instanca ove klase
- DI kontejner kreira objekat klase OsobaSkladiste i prosleđuje ga konstruktoru kontrolera
- Proces DI omogućava kontroleru da pristupi skladištu podataka kroz interfejs bez potrebe da zna koja se implementaciona klasa za to koristi

Kreiranje Index pogleda bazirano na modelu

Add Razor View

View name: Index

Template: List

Model class: Osoba (WebApp03.Models)

Options

- Create as a partial view
- Reference script libraries
- Use a layout page

...

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

Add Cancel

Direktiva @model

```
@model IEnumerable<WebApp03.Models.Osoba>
```

```
@{
```

```
    ViewData["Title"] = "Index";
```

```
}
```

Direktiva @model omogućava da se podacima koje je kontroler prosledio pogledu korišćenjem svojstva Model.

Podaci koje prosleđuje kontroler su tipizirani, u ovom slučaju objekti klase Osoba.

Index pogled -1

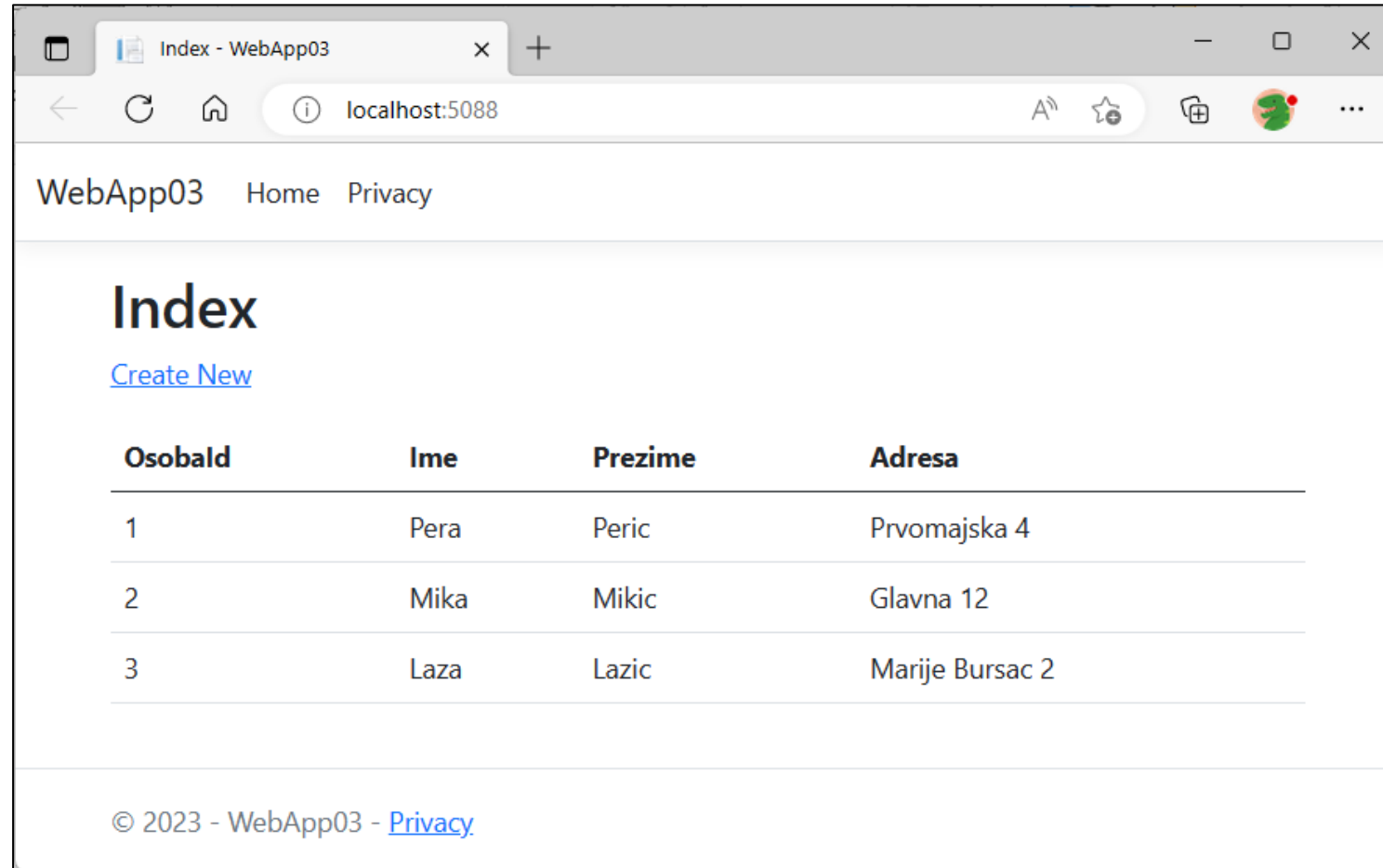
```
<h2>Index</h2>

<table class="table table-bordered table-striped">
  <thead>
    <tr>
      <th>
        Id
      </th>
      <th>
        Ime
      </th>
      <th>
        Prezime
      </th>
      <th>
        Adresa
      </th>
    </tr>
  </thead>
```

Index pogled -2

```
<tbody>
  @foreach (Osoba os in Model) {
    <tr>
      <td>
        @os.OsobaId
      </td>
      <td>
        @os.Ime
      </td>
      <td>
        @os.Prezime
      </td>
      <td>
        @os.Adresa
      </td>
    </tr>
  }
</tbody>
</table>
```

Prikaz Index pogleda



WebApp03 Home Privacy

Index

[Create New](#)

Osobald	Ime	Prezime	Adresa
1	Pera	Peric	Prvomajska 4
2	Mika	Mikic	Glavna 12
3	Laza	Lazic	Marije Bursac 2

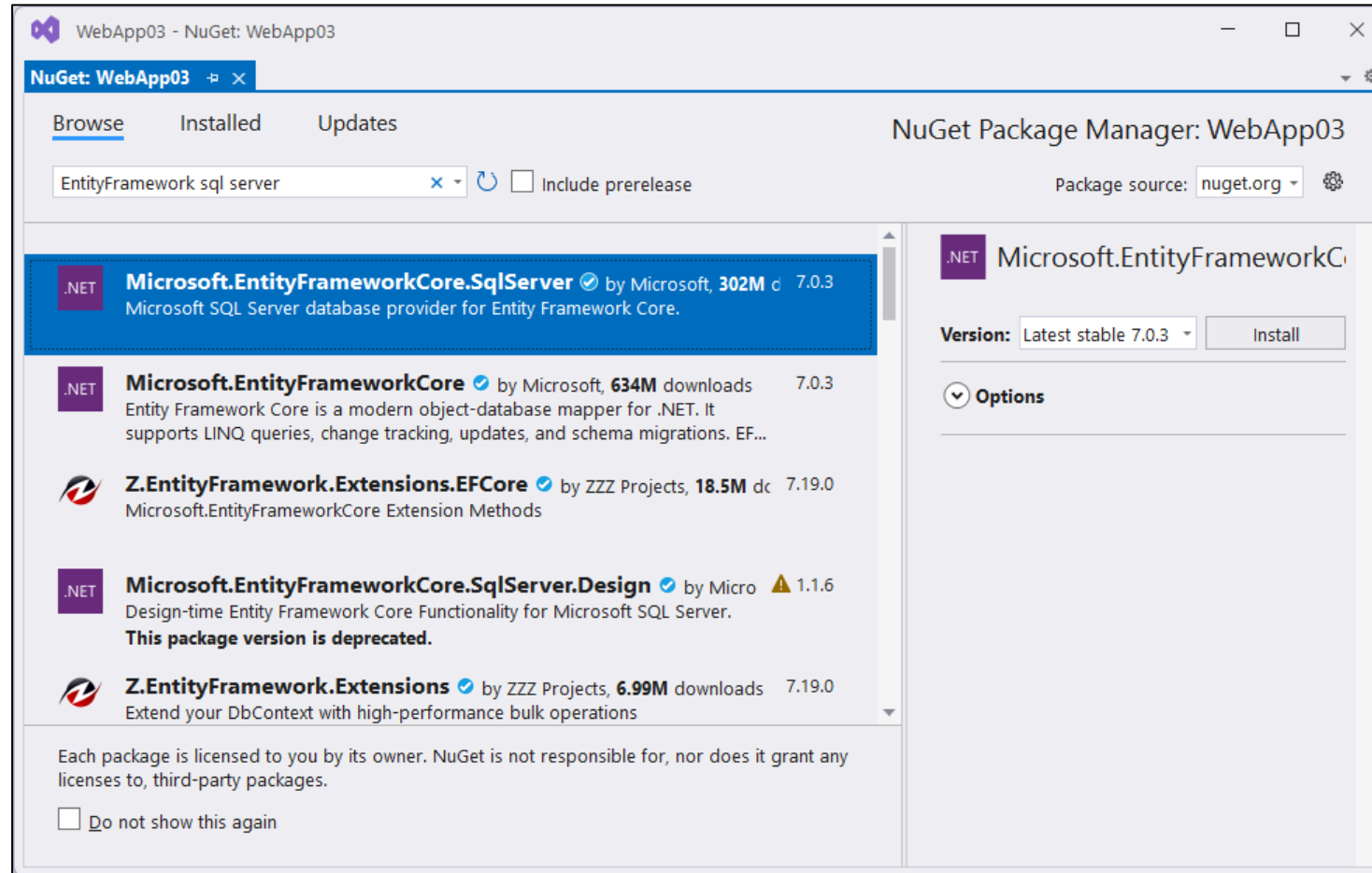
© 2023 - WebApp03 - [Privacy](#)

Kreiranje modela - code first
pristup

Fajl appsettings.json

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "DefaultConnection": "Data Source=GORAN-HP;Initial Catalog=OsobaDb0903;Integrated
Security=True;Encrypt=False"
  }
}
```


Microsoft.EntityFrameworkCore.SqlServer



Klasa OsobaContext

```
using Microsoft.EntityFrameworkCore;
```

```
public class OsobaContext : DbContext  
{  
    public DbSet<Osoba> Osobe { get; set; }  
}
```

Metoda `AddDbContext<TContext>` interfejsa `IServiceCollection`

- Ova metoda omogućava da `DbContext<TContext>` objekat bude raspoloživ za ubacivanje od strane DI kontejnera
- Omogućava da `DbContextOptions<TContext>` objekat bude raspoloživ za ubacivanje od strane DI kontejnera

Registrujemo DbContext kao servis

```
using Microsoft.EntityFrameworkCore;  
using WebApp03.Models;
```

```
// Add services to the container.  
builder.Services.AddControllersWithViews();  
var connectionString =  
builder.Configuration.GetConnectionString("DefaultConnection");  
  
builder.Services.AddDbContext<OsobaContext>(options =>  
    options.UseSqlServer(connectionString));  
  
builder.Services.AddTransient<ISkladiste, OsobaSkladiste>();
```

Metoda AddDbContext registruje **OsobaContext** klasu kao servis. Konfigurira se kontekst da koristi Sql Server bazu podataka. Instanca klase OsobaContext se ubacuje posredstvom DI kontejnera.

Klasa OsobaContext

```
using Microsoft.EntityFrameworkCore;
```

```
public class OsobaContext : DbContext
{
    public OsobaContext(DbContextOptions<OsobaContext> opcije) : base(opcije)
    {
    }

    public DbSet<Osoba> Osobe { get; set; }
}
```

Konstruktor DbContext klase mora imati kao parametar instancu klase DbContextOptions da bi se kreirao objekat DbContext. Ovu instancu ubacuje DI kontejner. Objekat DbContextOptions sadrži opcije za kreiranje DbContext objekta.

Klasa OsobaSkladisteDb

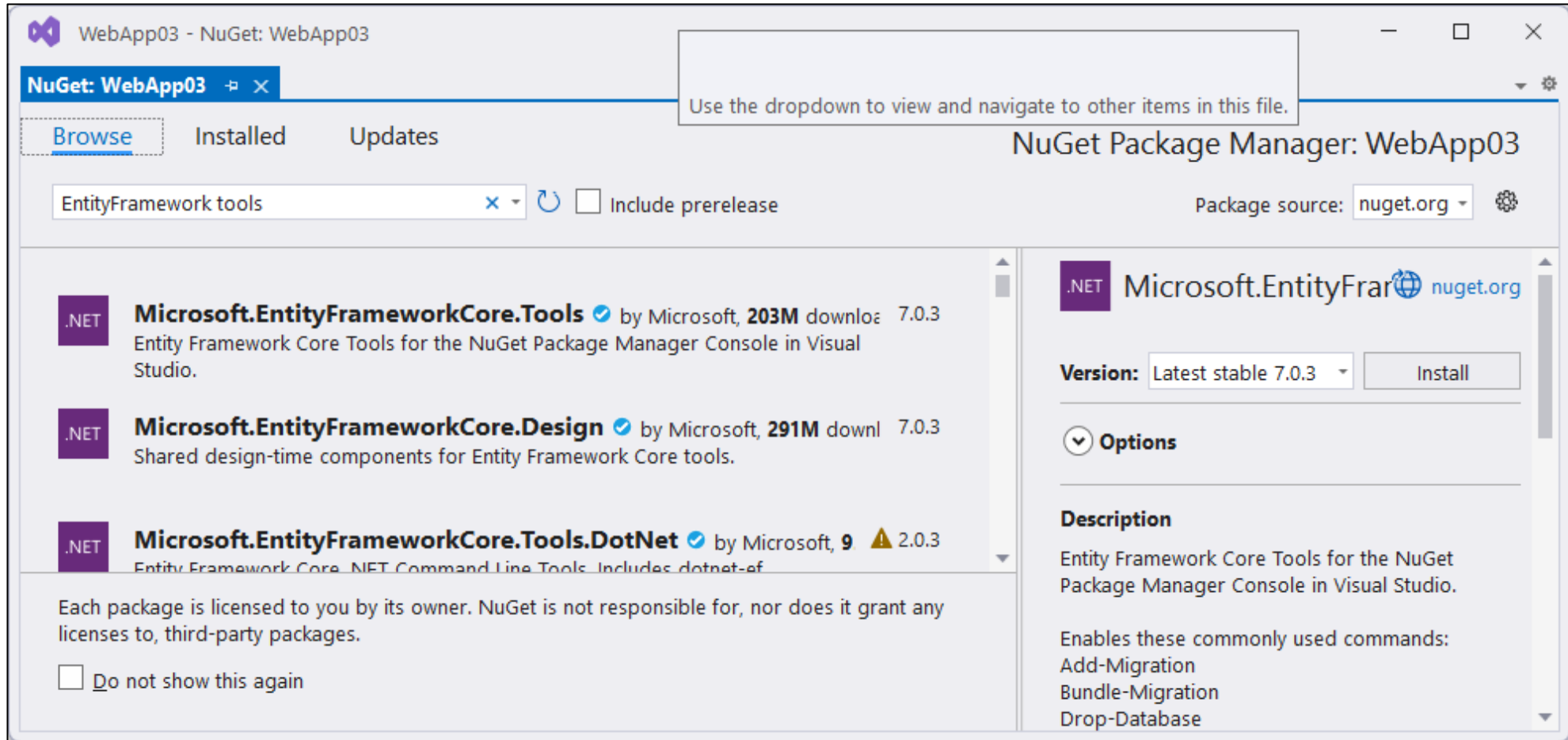
```
public class OsobaSkladisteDb : Iskladiste
{
    private readonly OsobaContext _db;
    public OsobaSkladisteDb(OsobaContext db)
    {
        _db = db;
    }

    public IEnumerable<Osoba> Osobe => _db.Osobe;
}
```

Promena implementacione klase servisa

```
var builder = WebApplication.CreateBuilder(args);  
  
// Add services to the container.  
builder.Services.AddControllersWithViews();  
//builder.Services.AddTransient<Iskladiste, OsobaSkladiste>();  
builder.Services.AddTransient<Iskladiste, OsobaSkladisteDb>();
```

Microsoft.EntityFrameworkCore.Tools

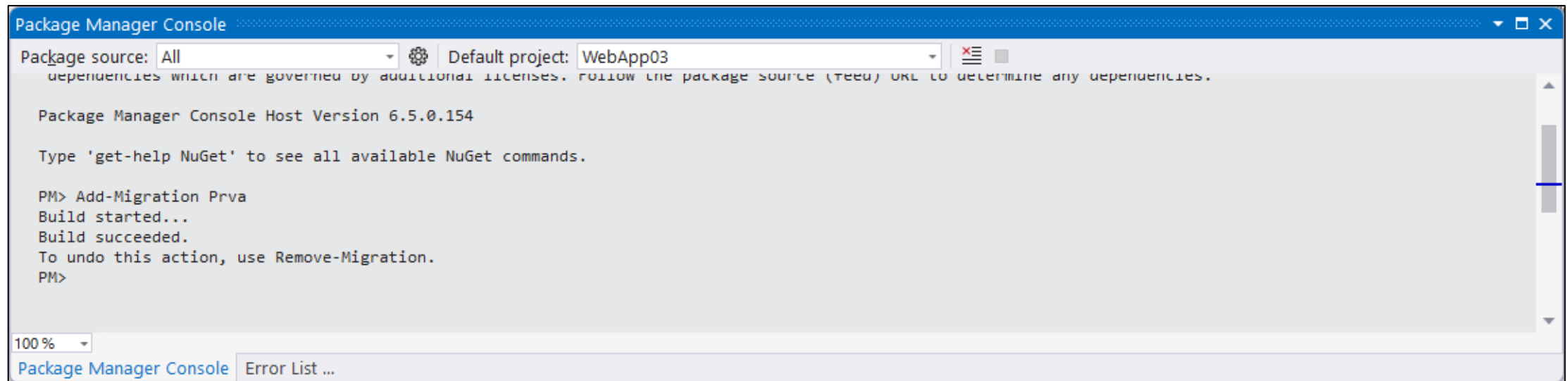


Code first migracije

- Omogućavaju kreiranje baze podataka na osnovu postojećih entitetskih klasa i DbContext klase
- Omogućavaju promenu baze podataka kada se promeni neka od entitetskih klasa i context klasa
- Komanda **Add-Migrations Prva** unutar alata **Package Manager Console** služi da se kreira nova migracija unutar klase Prva
- Nakon dodavanje migracije kreira se folder **Migrations** unutar ASP.NET Core MVC web aplikacije

Dodavanje migracije

- PM> Add-Migration Prva



```
Package Manager Console
Package source: All Default project: WebApp03
dependencies which are governed by additional licenses. Follow the package source (feed) URL to determine any dependencies.

Package Manager Console Host Version 6.5.0.154

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

PM> Add-Migration Prva
Build started...
Build succeeded.
To undo this action, use Remove-Migration.
PM>
```

100 %

Package Manager Console Error List ...

Klasa Prva.cs

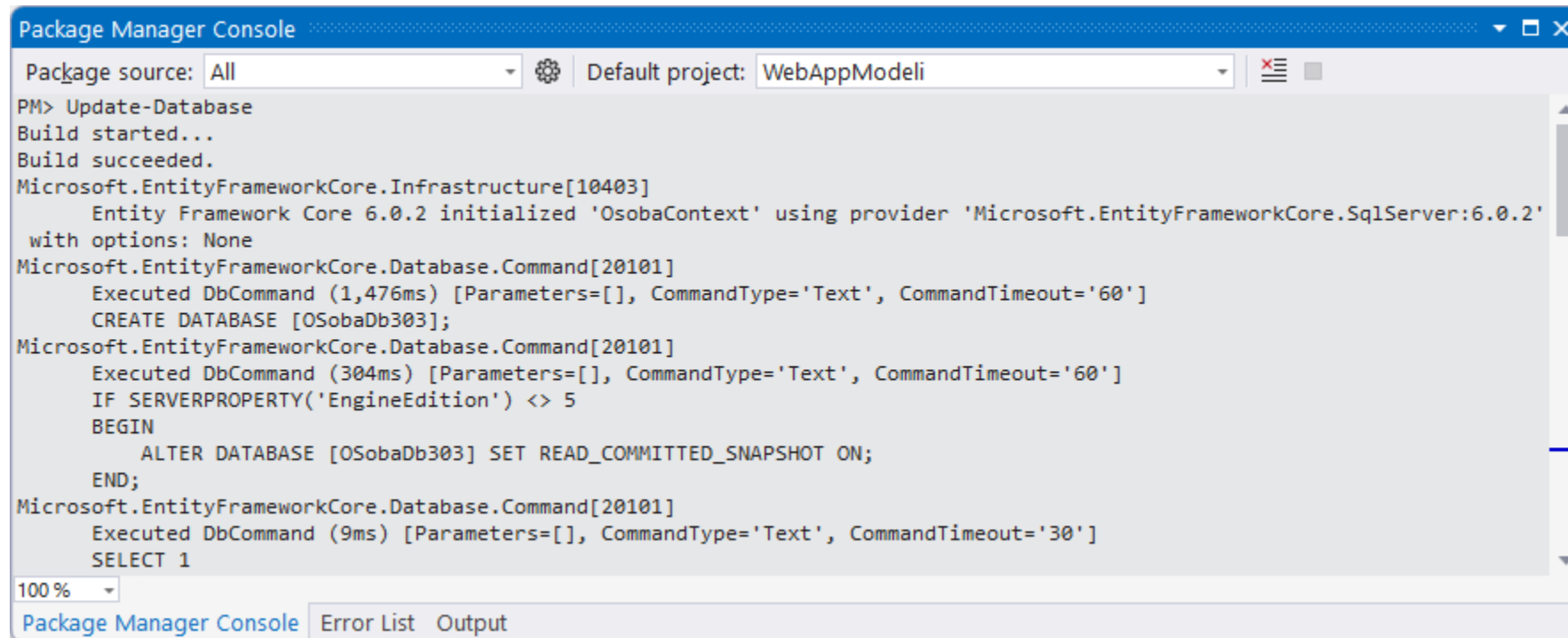
```
public partial class Prva : Migration
{
    protected override void Up(MigrationBuilder migrationBuilder)
    {
        migrationBuilder.CreateTable(
            name: "Osoba",
            columns: table => new
            {
                OsobaId = table.Column<int>(type: "int", nullable: false)
                    .Annotation("SqlServer:Identity", "1, 1"),
                Ime = table.Column<string>(type: "nvarchar(max)", nullable: false),
                Prezime = table.Column<string>(type: "nvarchar(max)", nullable: false),
                Adresa = table.Column<string>(type: "nvarchar(max)", nullable: false)
            },
            constraints: table =>
            {
                table.PrimaryKey("PK_Osoba", x => x.OsobaId);
            });
    }

    protected override void Down(MigrationBuilder migrationBuilder)
    {
        migrationBuilder.DropTable(
            name: "Osoba");
    }
}
```

Kreiranje baze na osnovu modela

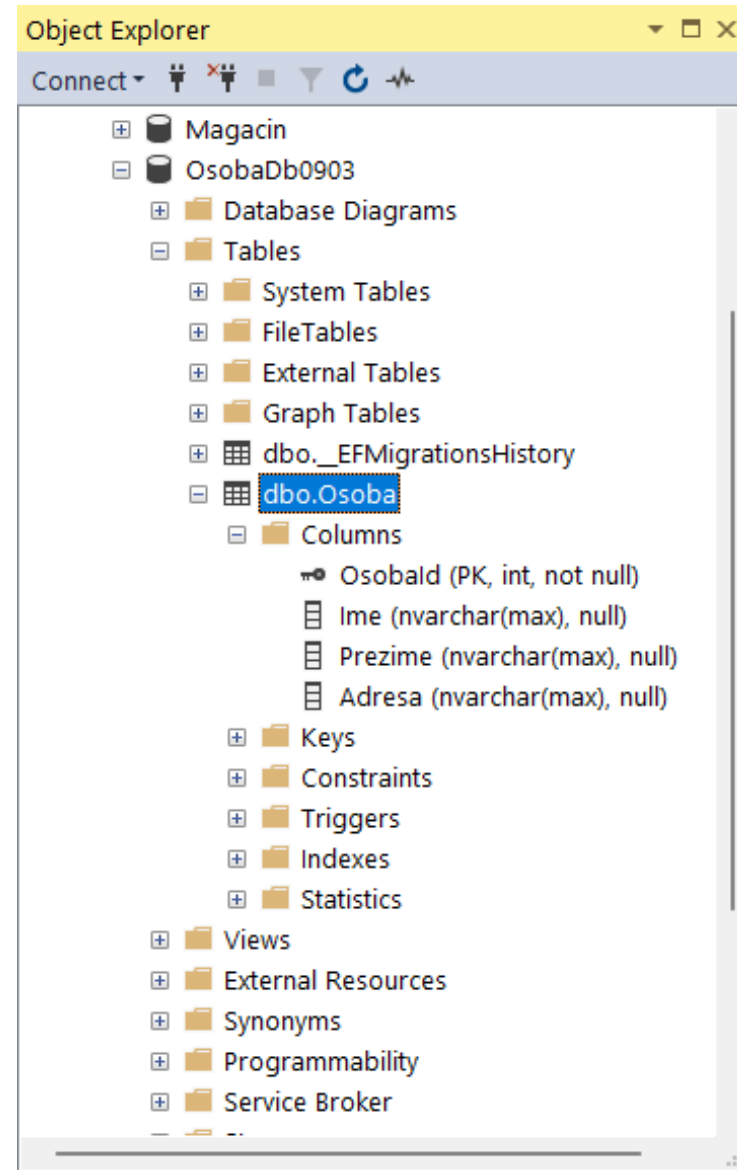
PM> Update-Database

Primena migracije na bazu

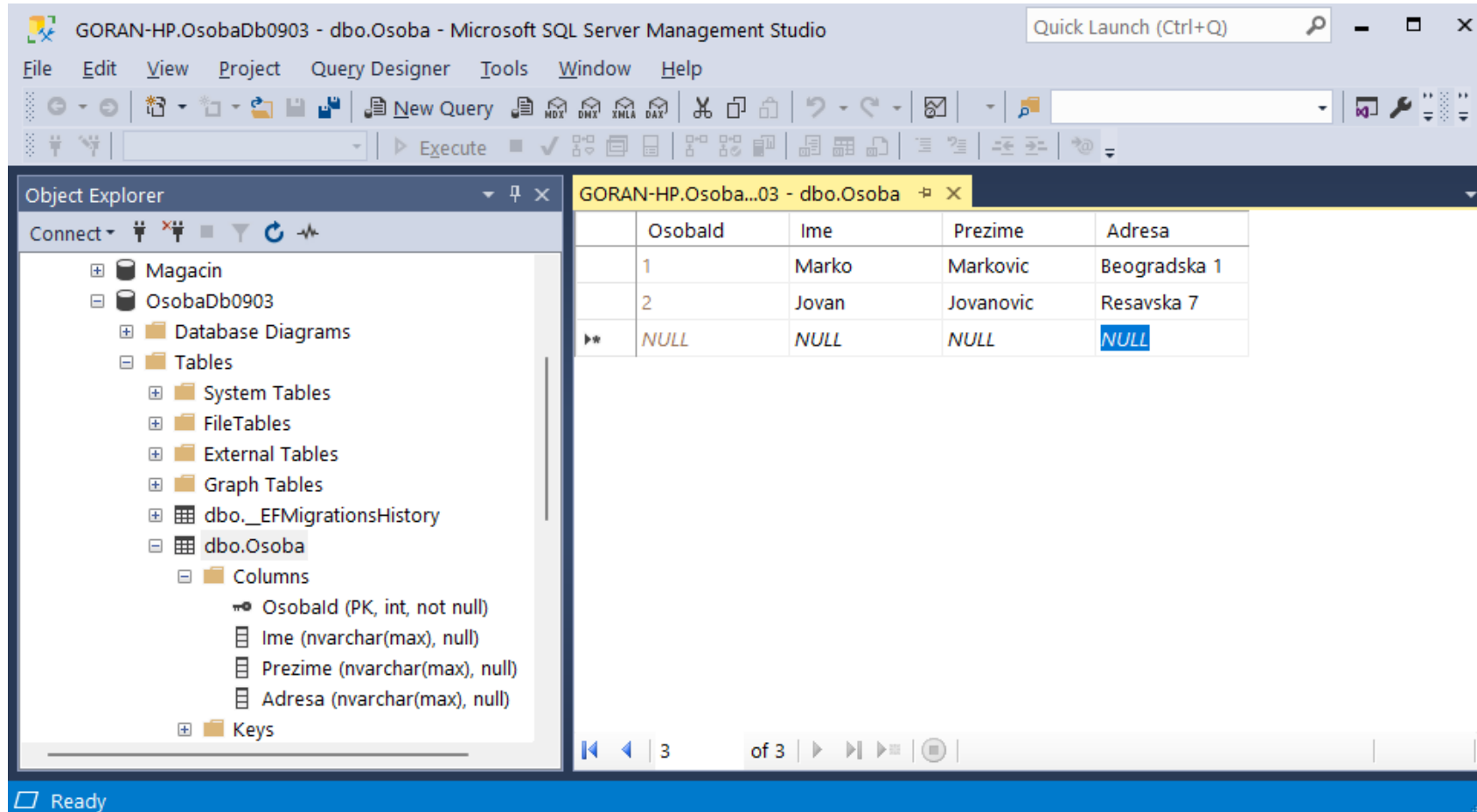


```
Package Manager Console
Package source: All Default project: WebAppModeli
PM> Update-Database
Build started...
Build succeeded.
Microsoft.EntityFrameworkCore.Infrastructure[10403]
    Entity Framework Core 6.0.2 initialized 'OsobaContext' using provider 'Microsoft.EntityFrameworkCore.SqlServer:6.0.2'
    with options: None
Microsoft.EntityFrameworkCore.Database.Command[20101]
    Executed DbCommand (1,476ms) [Parameters=[], CommandType='Text', CommandTimeout='60']
    CREATE DATABASE [OsobaDb303];
Microsoft.EntityFrameworkCore.Database.Command[20101]
    Executed DbCommand (304ms) [Parameters=[], CommandType='Text', CommandTimeout='60']
    IF SERVERPROPERTY('EngineEdition') <> 5
    BEGIN
        ALTER DATABASE [OsobaDb303] SET READ_COMMITTED_SNAPSHOT ON;
    END;
Microsoft.EntityFrameworkCore.Database.Command[20101]
    Executed DbCommand (9ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
    SELECT 1
```

Generisana baza podataka



Unos podataka u bazu

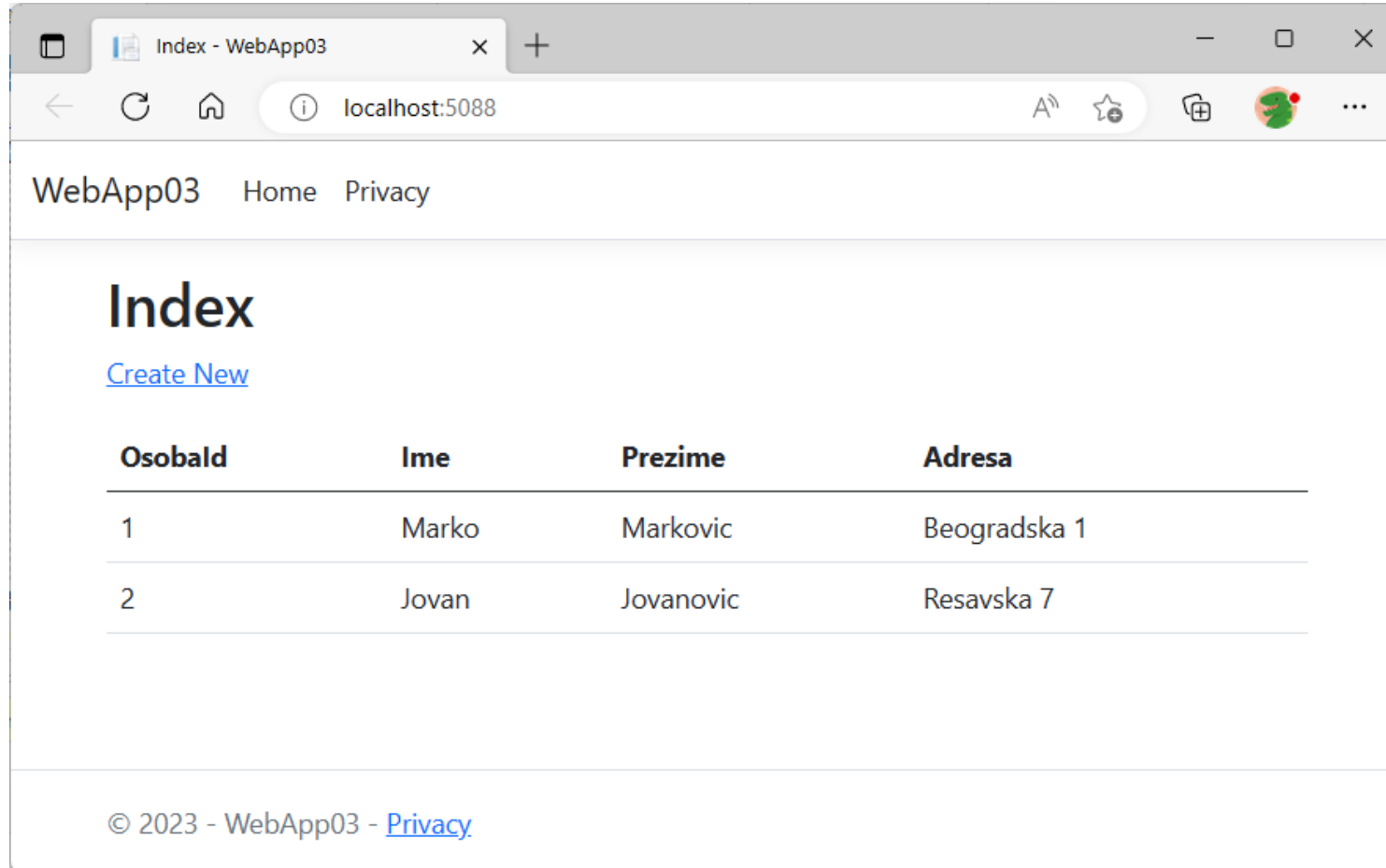


The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the current database is 'GORAN-HP.OsobaDb0903 - dbo.Osoba'. The menu bar includes File, Edit, View, Project, Query Designer, Tools, Window, and Help. The toolbar contains various icons for file operations and database management. The Object Explorer on the left shows the database structure, including the 'dbo.Osoba' table with columns: Osobald (PK, int, not null), Ime (nvarchar(max), null), Prezime (nvarchar(max), null), and Adresa (nvarchar(max), null). The main window displays a table with the following data:

	Osobald	Ime	Prezime	Adresa
	1	Marko	Markovic	Beogradska 1
	2	Jovan	Jovanovic	Resavska 7
▶▶	NULL	NULL	NULL	NULL

The status bar at the bottom indicates 'Ready' and shows navigation controls for the table, including '3 of 3'.

Index strana prikazuje podatke iz baze

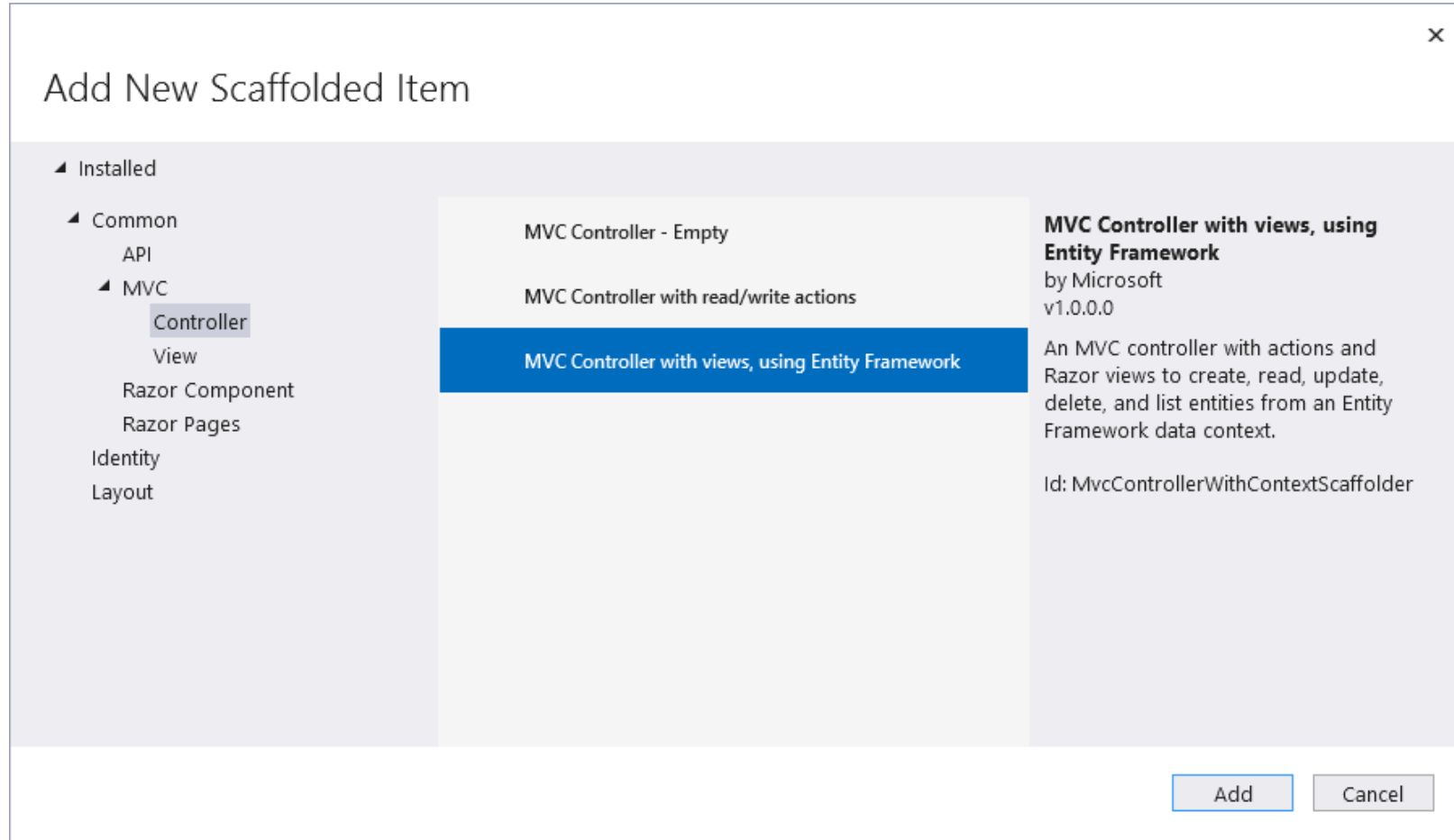


The screenshot shows a web browser window with the following elements:

- Browser tab: Index - WebApp03
- Address bar: localhost:5088
- Page title: WebApp03
- Page navigation: Home, Privacy
- Section title: Index
- Link: [Create New](#)
- Table with 4 columns: Osobald, Ime, Prezime, Adresa
- Table data:

Osobald	Ime	Prezime	Adresa
1	Marko	Markovic	Beogradska 1
2	Jovan	Jovanovic	Resavska 7
- Page footer: © 2023 - WebApp03 - [Privacy](#)

OsobaController



OsobaController

×

Add MVC Controller with views, using Entity Framework

Model class

DbContext class +

Database provider

Views

Generate views

Reference script libraries

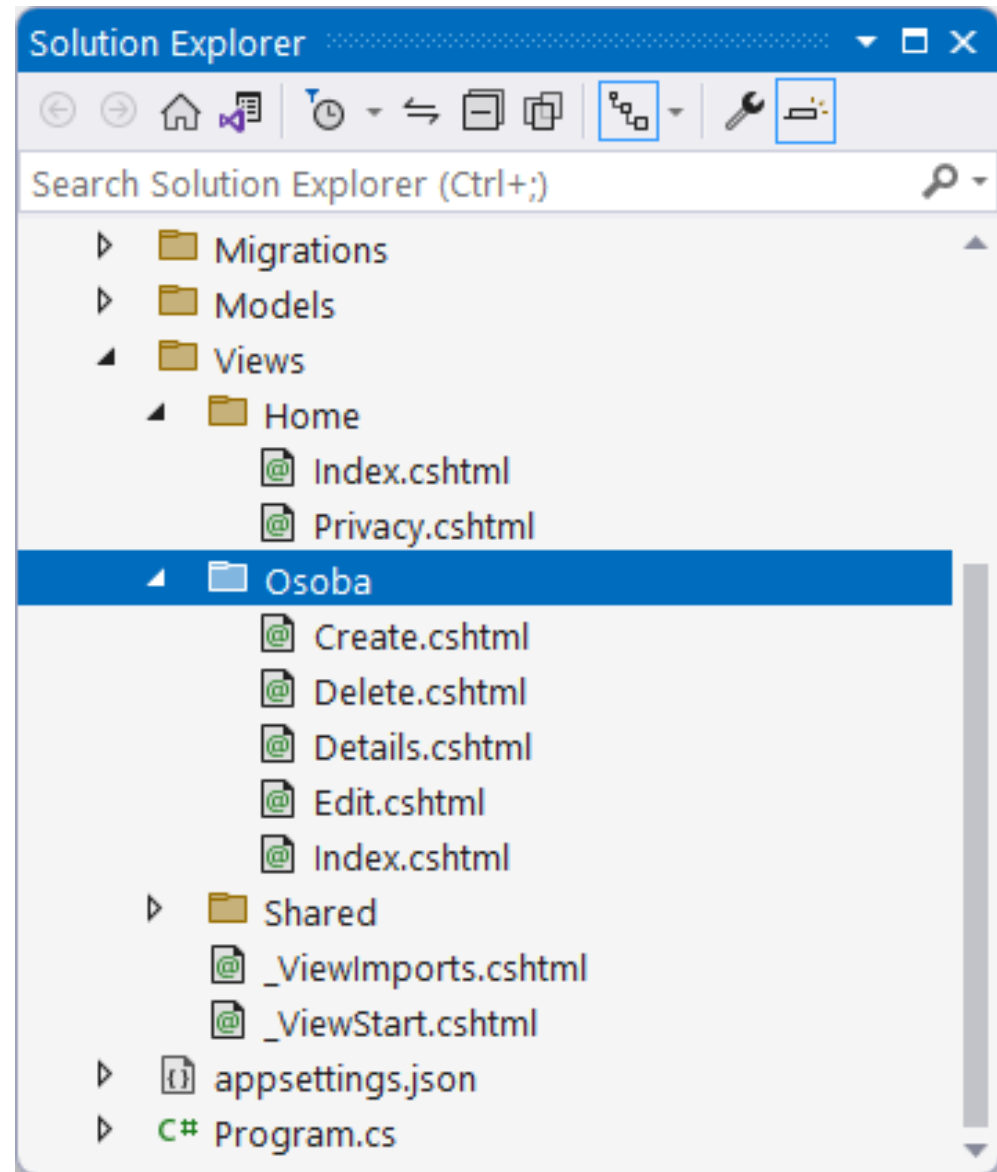
Use a layout page

...

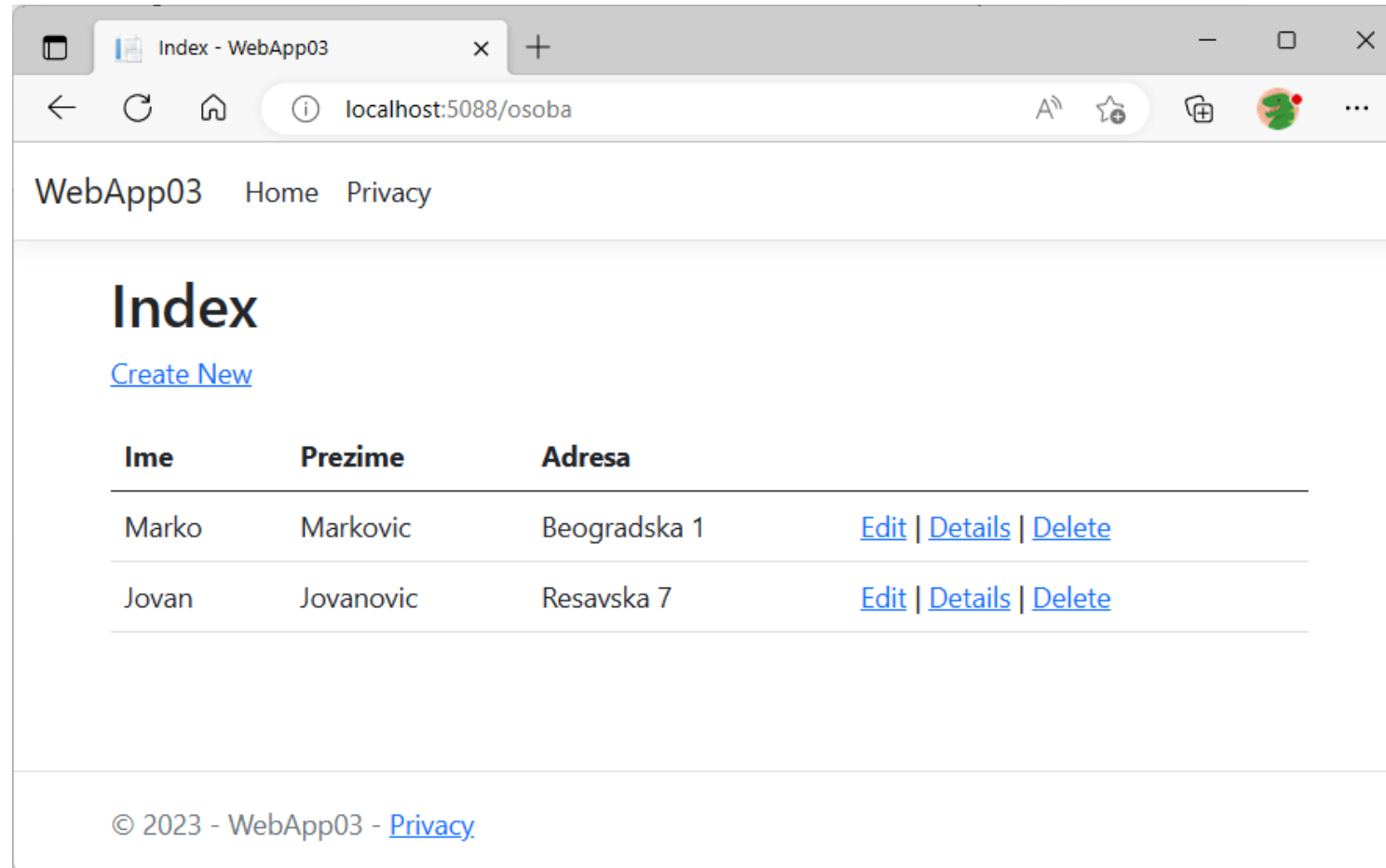
(Leave empty if it is set in a Razor `_viewstart` file)

Controller name

Generisani pogledi



Index pogled klase OsobaController



WebApp03 Home Privacy

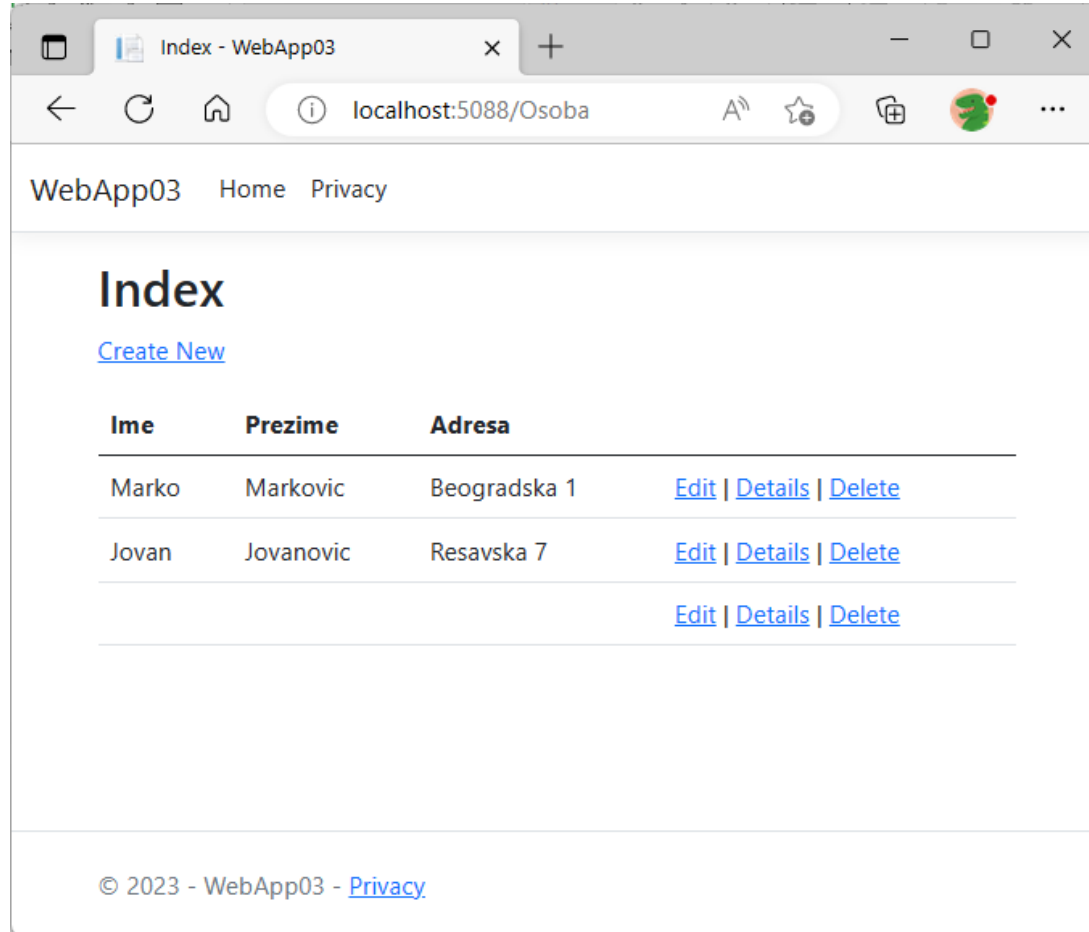
Index

[Create New](#)

Ime	Prezime	Adresa	
Marko	Markovic	Beogradska 1	Edit Details Delete
Jovan	Jovanovic	Resavska 7	Edit Details Delete

© 2023 - WebApp03 - [Privacy](#)

Unos podataka, poziv Create akcione metode



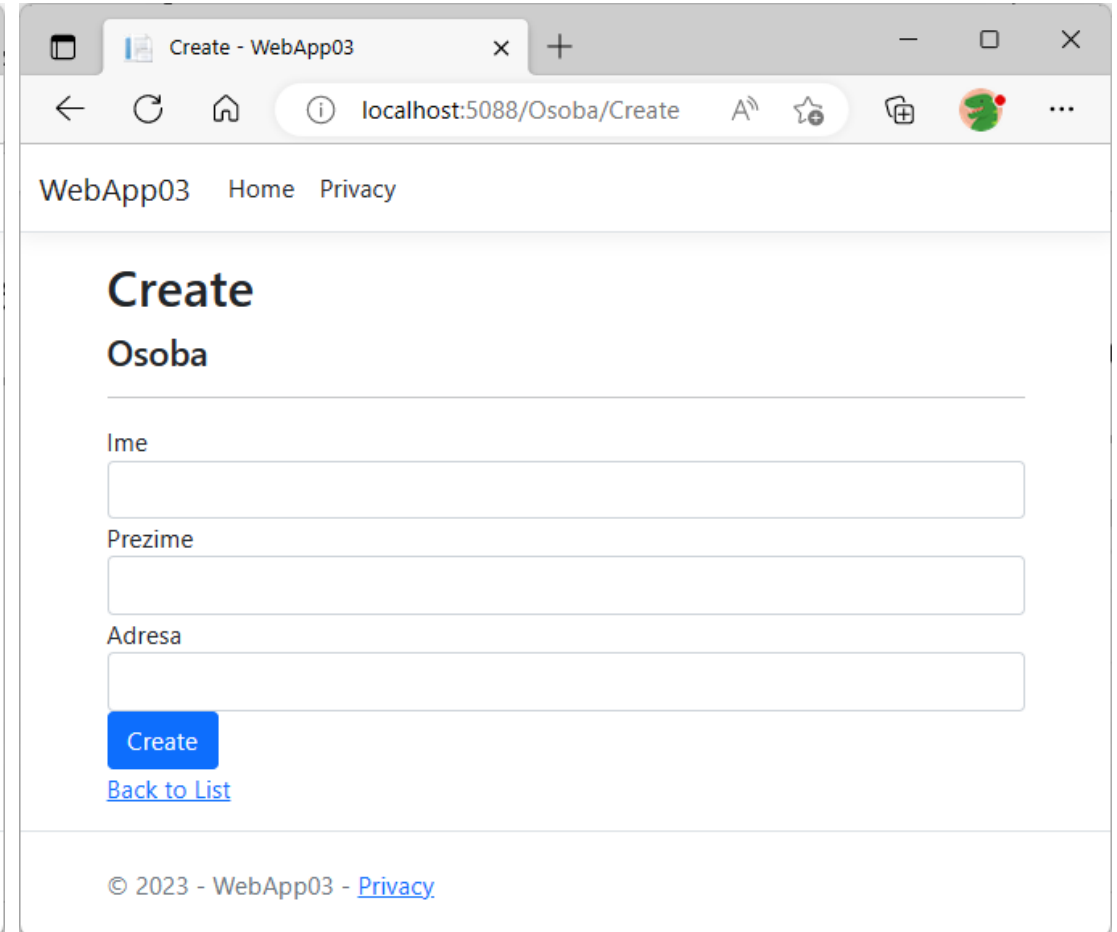
WebApp03 Home Privacy

Index

[Create New](#)

Ime	Prezime	Adresa	
Marko	Markovic	Beogradska 1	Edit Details Delete
Jovan	Jovanovic	Resavska 7	Edit Details Delete
			Edit Details Delete

© 2023 - WebApp03 - [Privacy](#)



WebApp03 Home Privacy

Create

Osoba

Ime

Prezime

Adresa

[Back to List](#)

© 2023 - WebApp03 - [Privacy](#)

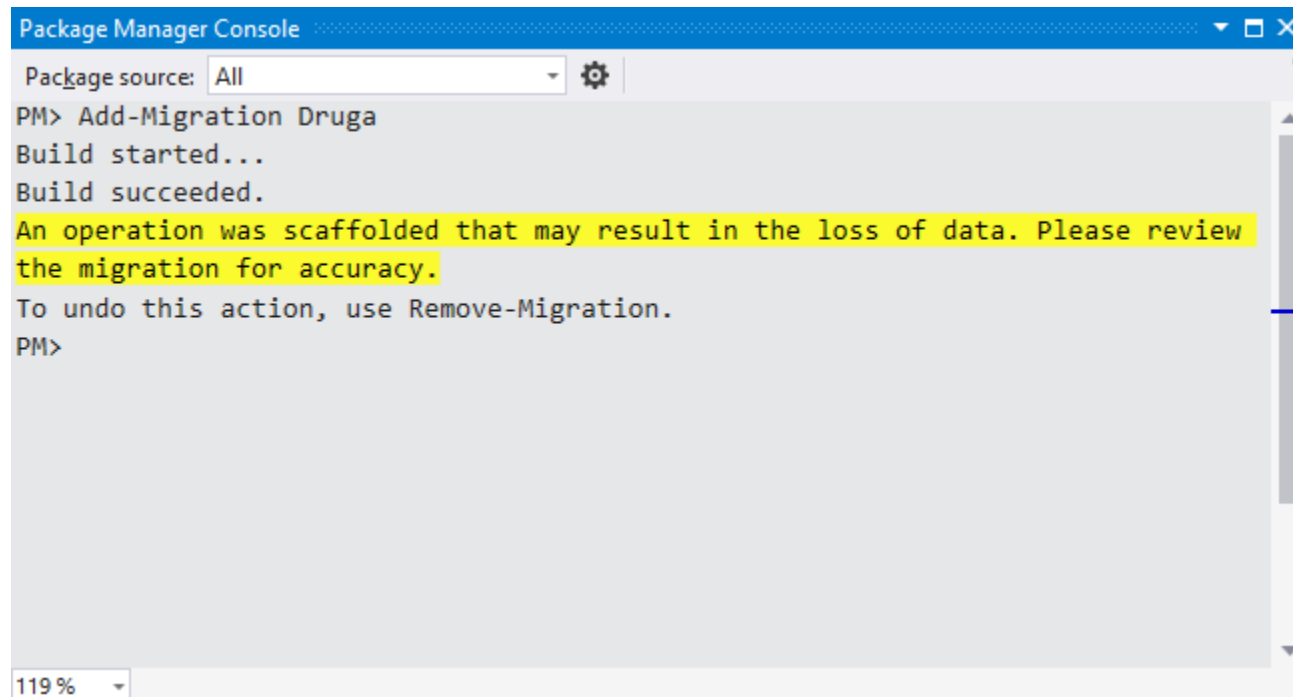
Model klasa sa anotacijama polja

```
using System.ComponentModel.DataAnnotations;
```

```
[Table("Osoba")]  
  
public class Osoba  
{  
    public int OsobaId { get; set; }  
    [Required(ErrorMessage = "Morate uneti ime")]  
    [StringLength(30, ErrorMessage = "Ime je duzine do 30 karaktera")]  
    public string Ime { get; set; }  
  
    [Required(ErrorMessage = "Morate uneti prezime")]  
    [StringLength(30, ErrorMessage = "Prezime je duzine do 30 karaktera")]  
    public string Prezime { get; set; }  
  
    [Required(ErrorMessage = "Morate uneti adresu")]  
    [StringLength(60, ErrorMessage = "Adresa je duzine do 60 karaktera")]  
    public string Adresa { get; set; }  
  
}
```

Dodavanje nove migracije

```
PM> Add-Migration Druga
```



```
Package Manager Console
Package source: All
PM> Add-Migration Druga
Build started...
Build succeeded.
An operation was scaffolded that may result in the loss of data. Please review
the migration for accuracy.
To undo this action, use Remove-Migration.
PM>
```

119 %

Metoda Up() migracije

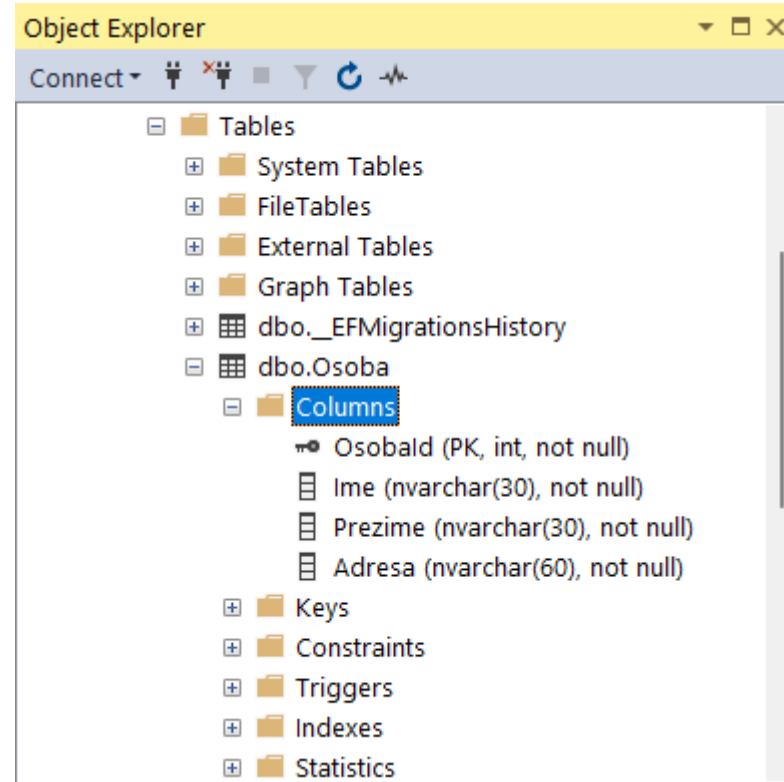
```
public partial class Druga : Migration
{
    protected override void Up(MigrationBuilder migrationBuilder)
    {
        migrationBuilder.AlterColumn<string>(
            name: "Prezime",
            table: "Osoba",
            type: "nvarchar(30)",
            maxLength: 30,
            nullable: false,
            oldClrType: typeof(string),
            oldType: "nvarchar(max)");

        migrationBuilder.AlterColumn<string>(
            name: "Ime",
            table: "Osoba",
            type: "nvarchar(30)",
            maxLength: 30,
            nullable: false,
            oldClrType: typeof(string),
            oldType: "nvarchar(max)");

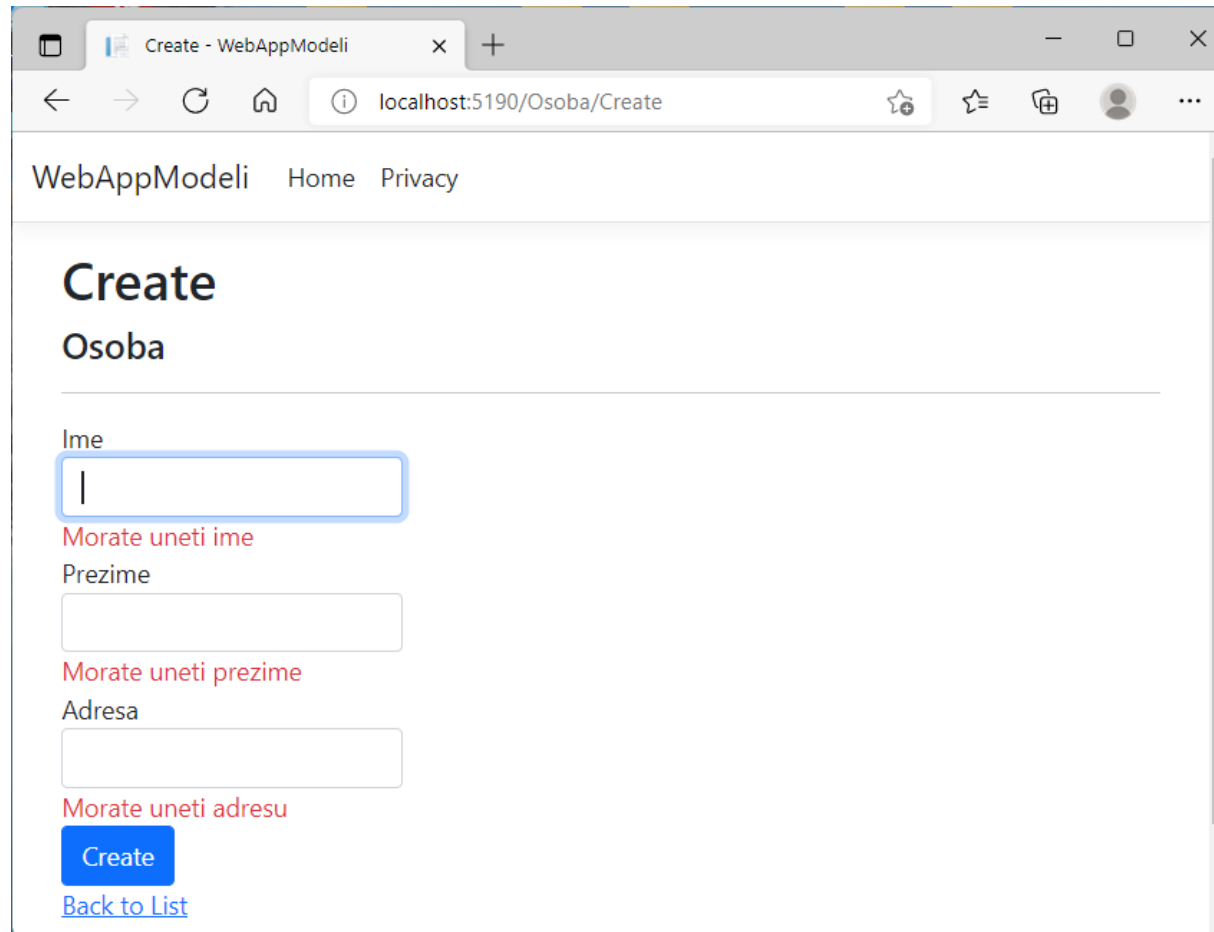
        migrationBuilder.AlterColumn<string>(
            name: "Adresa",
            table: "Osoba",
            type: "nvarchar(60)",
            maxLength: 60,
            nullable: false,
            oldClrType: typeof(string),
            oldType: "nvarchar(max)");
    }
}
```

Primena druge migracije

PM> Update-Database



Validacija podataka



The screenshot shows a web browser window with the address bar displaying 'localhost:5190/Osoba/Create'. The page title is 'WebAppModeli' and it includes navigation links for 'Home' and 'Privacy'. The main heading is 'Create Osoba'. The form contains three input fields: 'Ime', 'Prezime', and 'Adresa'. The 'Ime' field is highlighted with a blue border and contains a red error message: 'Morate uneti ime'. The 'Prezime' field also has a red error message: 'Morate uneti prezime'. The 'Adresa' field has a red error message: 'Morate uneti adresu'. Below the form is a blue 'Create' button and a blue link labeled 'Back to List'.

WebAppModeli Home Privacy

Create

Osoba

Ime
 |
Morate uneti ime

Prezime

Morate uneti prezime

Adresa

Morate uneti adresu

[Create](#)

[Back to List](#)

Unos podataka

WebAppModeli Home Privacy

Index

[Create New](#)

Ime	Prezime	Adresa	
Marko	Marković	Beogradska 1	Edit Details Delete
Mirko	Mirković	Resavska 7	Edit Details Delete
Jovan	Ristić	Bul Mihajla Pupina 15	Edit Details Delete

WebAppModeli Home Privacy

Create Osoba

Ime

Prezime

Adresa

[Create](#)

[Back to List](#)

© 2022 - WebAppModeli - [Privacy](#)

Kreiranje modela- database
first pristup

Kreiranje baze

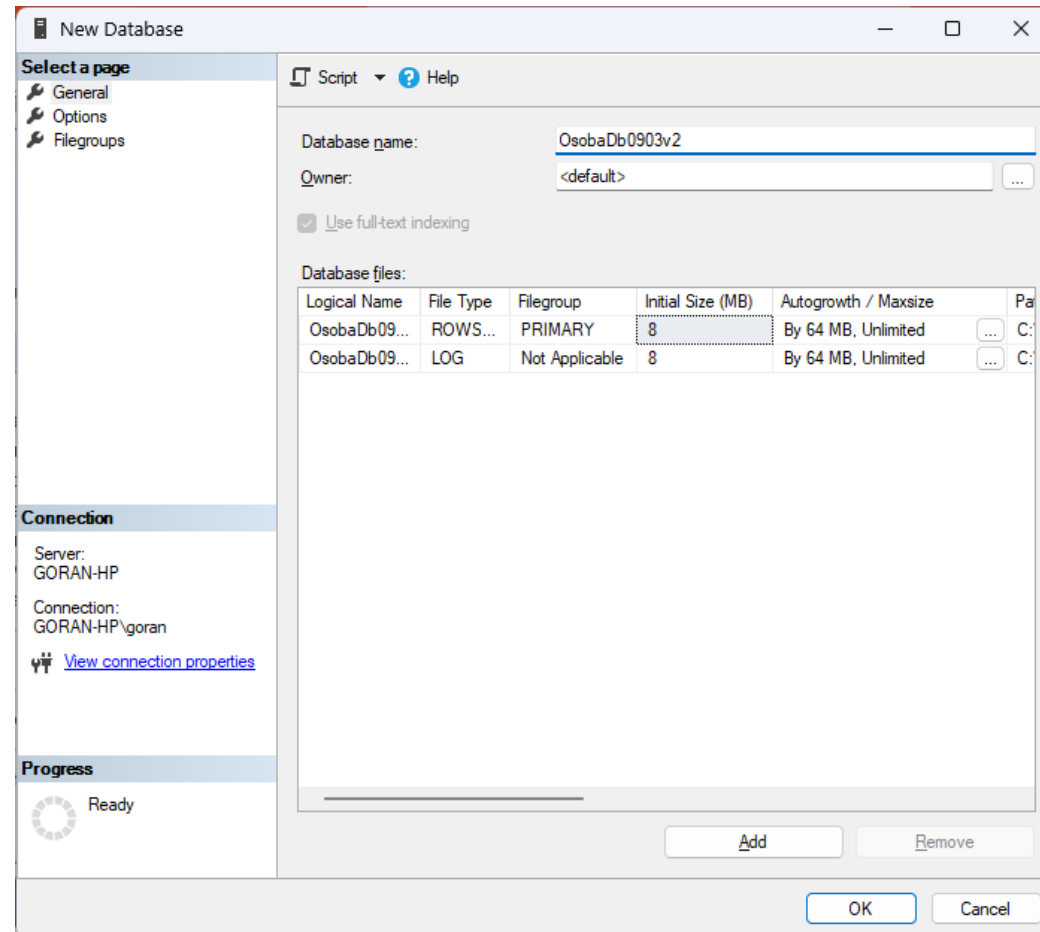
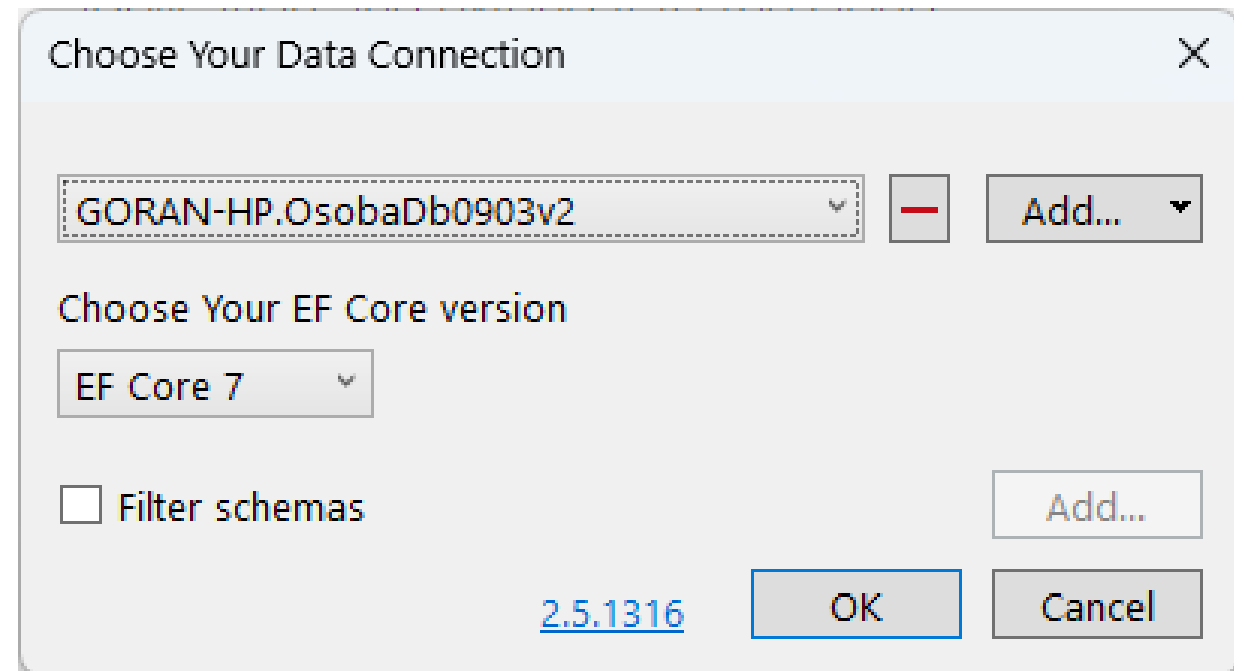
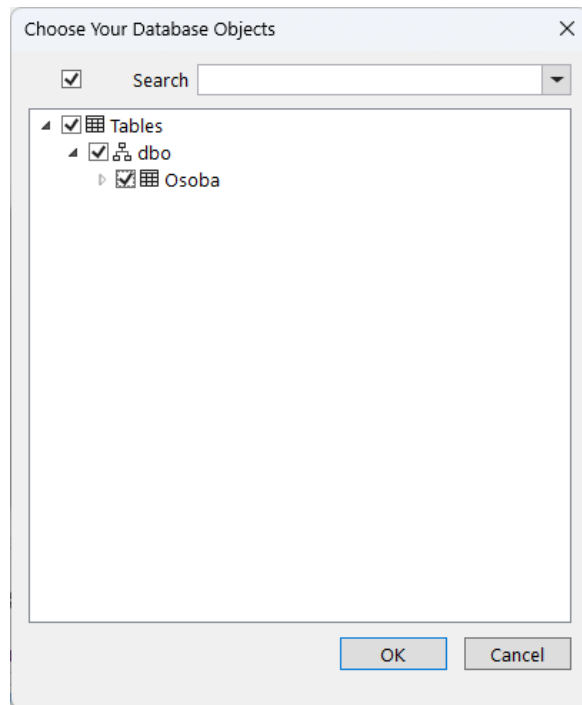


Tabela u bazi OsobaDb303ver2

```
CREATE TABLE Osoba(  
OsobaId INT IDENTITY(1,1) PRIMARY KEY NOT NULL,  
Ime NVARCHAR(30) NOT NULL,  
Prezime NVARCHAR(30) NOT NULL,  
Adresa NVARCHAR(60) NOT NULL  
)
```

EF Core Power Tools -1



EF Core Power Tools -2

Choose Your Settings for Project WebApp03v2

Context name:

Namespace:

EntityTypes path (f.ex. Models) - optional:

What to generate:

Naming

- Pluralize or singularize generated object names (English)
- Use table and column names directly from the database

Use DataAnnotation attributes to configure the model

Customize code using templates:

Include connection string in generated code

Install the EF Core provider package in the project

[Help](#) [★ Rate](#)

Generisana entitetska klasa

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

    [Required]
    [StringLength(50)]
    public string Ime { get; set; }

    [Required]
    [StringLength(30)]
    public string Prezime { get; set; }

    [Required]
    [StringLength(30)]
    public string Adresa { get; set; }
}
```


Generisana DbContext klasa

```
public partial class OsobaContext : DbContext
{
    public OsobaContext(DbContextOptions<OsobaContext> options)
        : base(options)
    {
    }

    public virtual DbSet<Osoba> Osobe { get; set; }
}
}
```

Modifikovani appsettings.json

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "DefaultConnection": "Data Source=Goran-HP;Initial Catalog=OsobaDb0903v2;Integrated Security=True;Encrypt=False"
  }
}
```

Metoda ConfigureServices() klase Startup

```
using Microsoft.EntityFrameworkCore;  
using WebApp03v2.Models;
```

```
// Add services to the container.  
var connectionString = builder.Configuration.GetConnectionString("DefaultConnection");  
builder.Services.AddControllersWithViews();  
builder.Services.AddDbContext<OsobaContext>(opcije =>  
    opcije.UseSqlServer(connectionString));
```

Kreiranje OsobaController klase

Add MVC Controller with views, using Entity Framework

Model class: Osoba (WebApp03v2.Models)

DbContext class: OsobaContext (WebApp03v2.Models) +

Database provider: SQL Server

Views

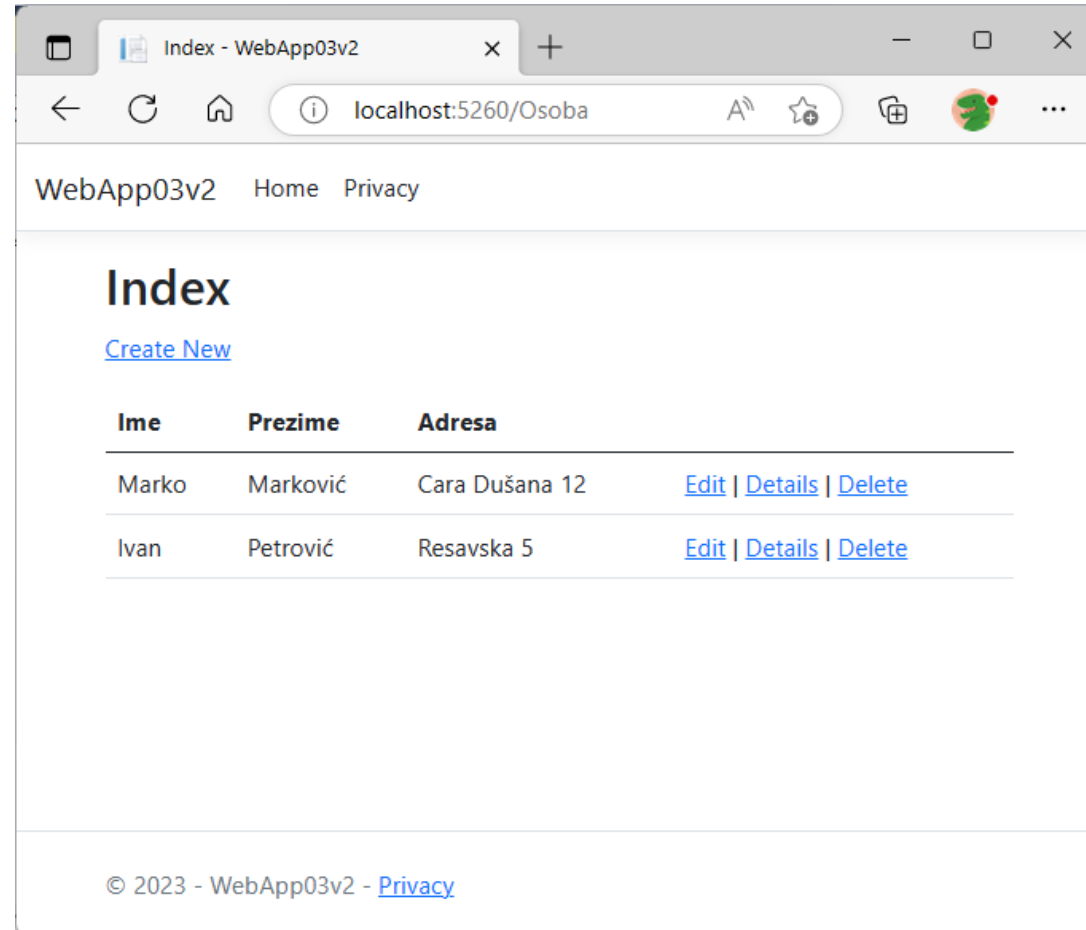
- Generate views
- Reference script libraries
- Use a layout page

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

Controller name: OsobaController

Add Cancel

Index pogled



WebApp03v2 Home Privacy

Index

[Create New](#)

Ime	Prezime	Adresa	
Marko	Marković	Cara Dušana 12	Edit Details Delete
Ivan	Petrović	Resavska 5	Edit Details Delete

© 2023 - WebApp03v2 - [Privacy](#)

Pitanje 1

Modeli podataka kod ASP.NET Core MVC web aplikacija definišu se unutar:

- a. foldera Models
- b. foldera wwwroot
- c. foldera Views

Odgovor: a

Pitanje 2

Unutar pogleda (cshtml) fajla podacima koje prosleđuje kontroler pristupa se:

- a. korišćenjem svojstva Model
- b. korišćenjem ključne reči @model
- c. korišćenjem klase ModelData

Odgovor: a