

View komponente

# Ekstenzija Open Command Line

Extensions -> Manage Extensions

The screenshot shows the 'Manage Extensions' window in Visual Studio. The window title is 'Manage Extensions'. On the left, there is a navigation pane with 'Online' expanded, showing 'Visual Studio Marketplace' and 'Search Results'. The main area displays a list of extensions sorted by 'Relevance'. The 'Open Command Line' extension is highlighted in blue. Below it are 'Smart Command Line Arguments', 'Debug Command Line', 'Command Task Runner', 'Set Project Command Line Arguments 2017', and 'Visual Studio Command Prompt (devCmd)'. On the right, the details for the 'Open Command Line' extension are shown, including 'Created By: Mads Kristensen', 'Version: 2.4.226', 'Downloads: 555225', 'Pricing Category: Free', and a 'Rating' of 5 stars (83 Votes). There are also links for 'More Information' and 'Report Extension to Microsoft'. At the bottom right, there is a 'Close' button.

Manage Extensions

Sort by: Relevance

Open Command Line

**Open Command Line**  
Opens a command line at the root of the project.  
Support for all consoles such as CMD, PowerShell,...

**Smart Command Line Arguments**  
A Visual Studio Extension which aims to provide a better UI to manage your command line arguments.

**Debug Command Line**  
Provides a toolbar dropdown to quickly switch between recently used debugger command lines.

**Command Task Runner**  
Adds support for command line batch files in Visual Studio 2015's Task Runner Explorer. Supports .cmd, .bat, .ps1 and .psm1 files.

**Set Project Command Line Arguments 2017**  
Provides a command that sets the startup project's command line arguments property from text in the text editor.

**Visual Studio Command Prompt (devCmd)**

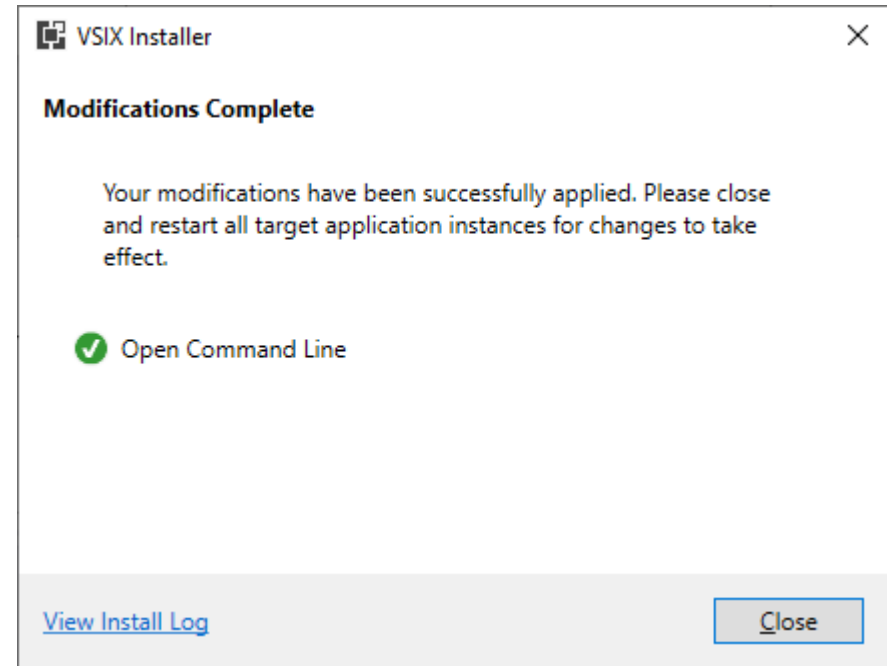
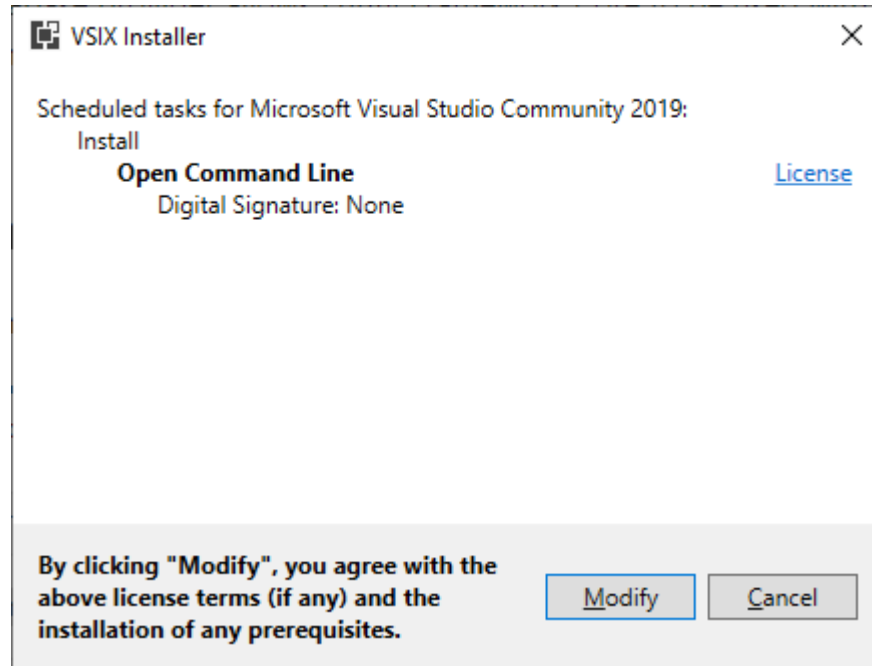
Created By: Mads Kristensen  
Version: 2.4.226  
Downloads: 555225  
Pricing Category: Free  
Rating: ★★★★★ (83 Votes)  
[More Information](#)  
[Report Extension to Microsoft](#)

Scheduled For Install: None  
Scheduled For Update: None  
Scheduled For Uninstall: None

Change your settings for Extensions

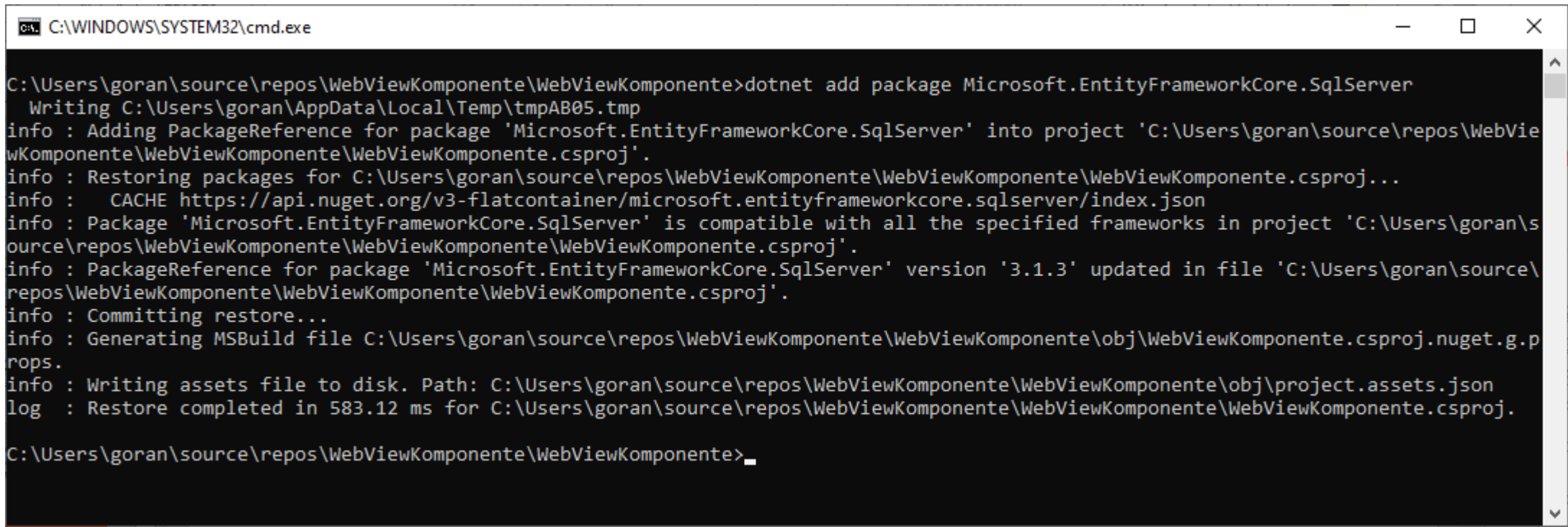
Close

# Ekstenzija Open Command Line



# .NET Core CLI

```
dotnet add package Microsoft.EntityFrameworkCore.SqlServer
```



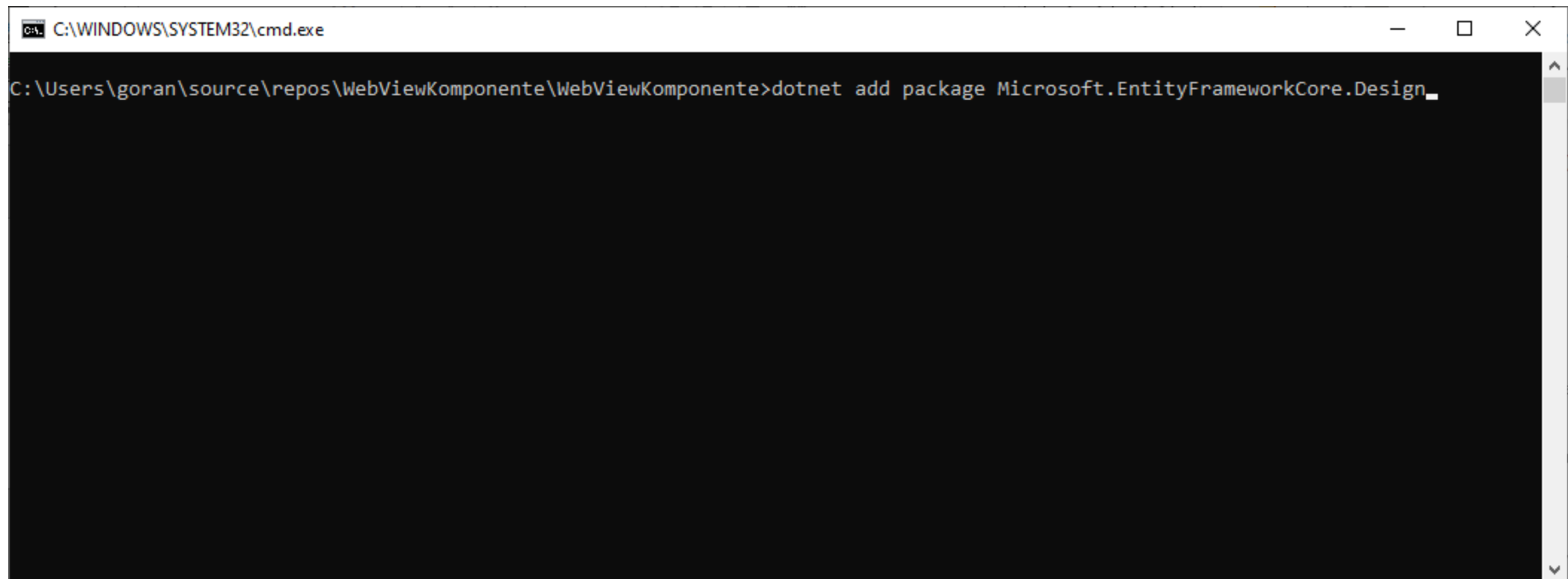
```
C:\WINDOWS\SYSTEM32\cmd.exe

C:\Users\goran\source\repos\WebViewKomponente\WebViewKomponente>dotnet add package Microsoft.EntityFrameworkCore.SqlServer
  Writing C:\Users\goran\AppData\Local\Temp\tmpAB05.tmp
info : Adding PackageReference for package 'Microsoft.EntityFrameworkCore.SqlServer' into project 'C:\Users\goran\source\repos\WebViewKomponente\WebViewKomponente\WebViewKomponente.csproj'.
info : Restoring packages for C:\Users\goran\source\repos\WebViewKomponente\WebViewKomponente\WebViewKomponente.csproj...
info :   CACHE https://api.nuget.org/v3-flatcontainer/microsoft.entityframeworkcore.sqlserver/index.json
info : Package 'Microsoft.EntityFrameworkCore.SqlServer' is compatible with all the specified frameworks in project 'C:\Users\goran\source\repos\WebViewKomponente\WebViewKomponente\WebViewKomponente.csproj'.
info : PackageReference for package 'Microsoft.EntityFrameworkCore.SqlServer' version '3.1.3' updated in file 'C:\Users\goran\source\repos\WebViewKomponente\WebViewKomponente\WebViewKomponente.csproj'.
info : Committing restore...
info : Generating MSBuild file C:\Users\goran\source\repos\WebViewKomponente\WebViewKomponente\obj\WebViewKomponente.csproj.nuget.g.props.
info : Writing assets file to disk. Path: C:\Users\goran\source\repos\WebViewKomponente\WebViewKomponente\obj\project.assets.json
log  : Restore completed in 583.12 ms for C:\Users\goran\source\repos\WebViewKomponente\WebViewKomponente\WebViewKomponente.csproj.

C:\Users\goran\source\repos\WebViewKomponente\WebViewKomponente>_
```

# Paket Microsoft.EntityFrameworkCore.Design

```
dotnet add package Microsoft.EntityFrameworkCore.Design
```



```
C:\WINDOWS\SYSTEM32\cmd.exe  
C:\Users\goran\source\repos\WebViewKomponente\WebViewKomponente>dotnet add package Microsoft.EntityFrameworkCore.Design_
```

# Fajl appsettings.json

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "DefaultConnection": "Data Source=(local)\\SqlExpress;Initial Catalog=OsobaDb3003;Integrated
Security=True"
  }
}
```

# Model klasa Osoba

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

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

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

    [Required()]
    [StringLength(20)]
    public string Telefon { get; set; }
}
```

# DbContext klasa

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

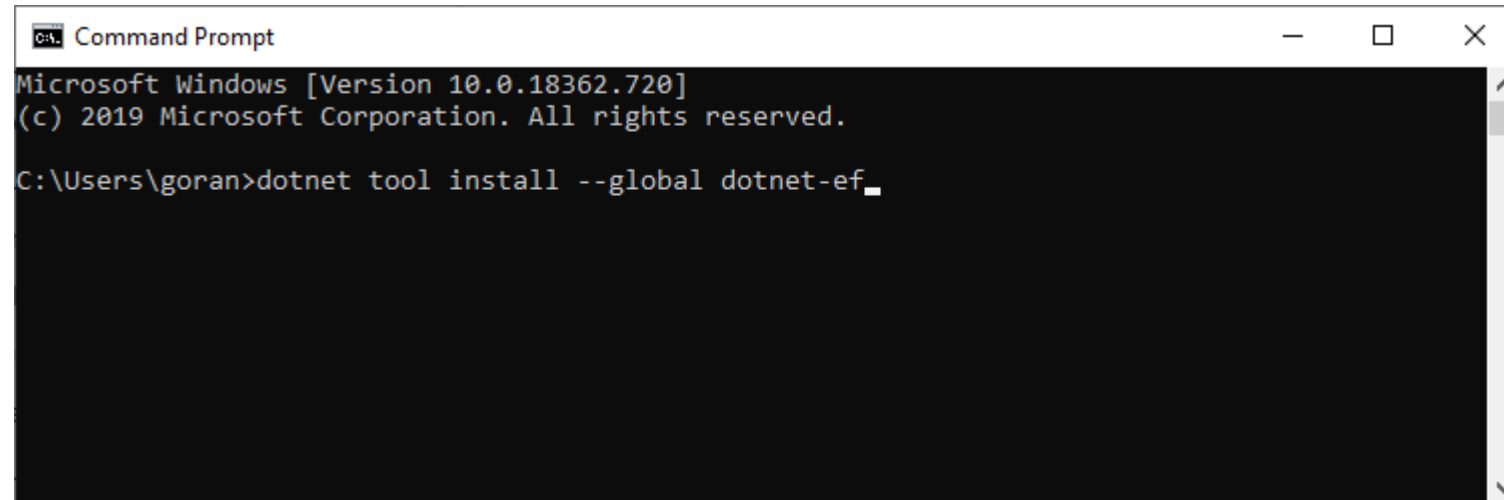


# Registrujemo DbContext kao servis

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

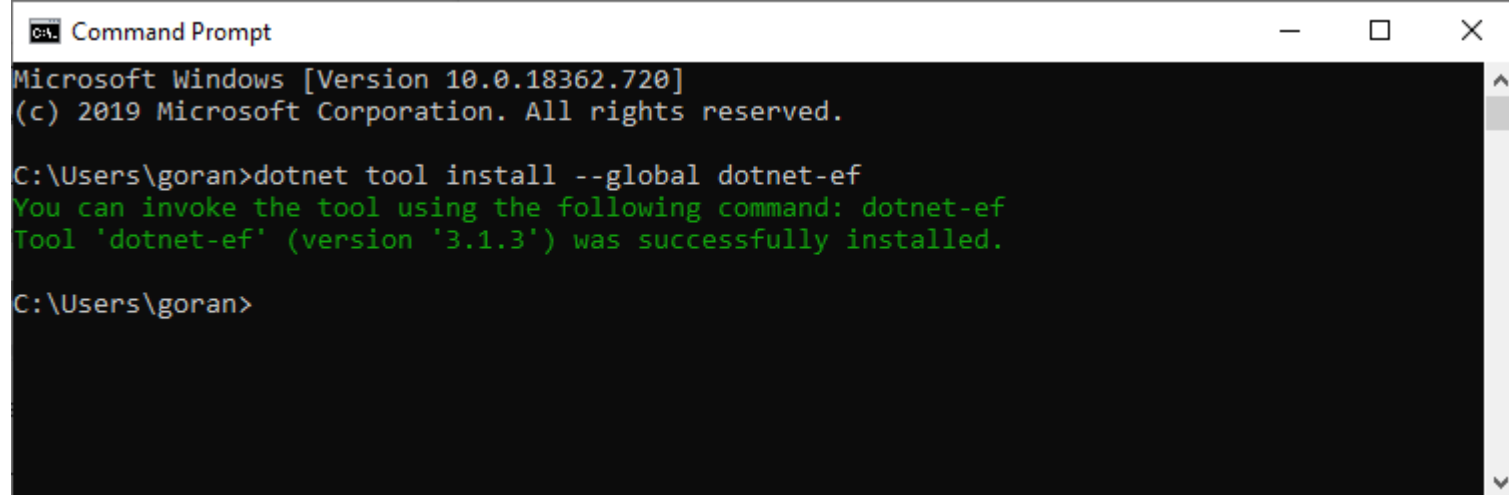
# dotnet ef alat

```
dotnet tool install --global dotnet-ef
```



```
Command Prompt
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\goran>dotnet tool install --global dotnet-ef_
```



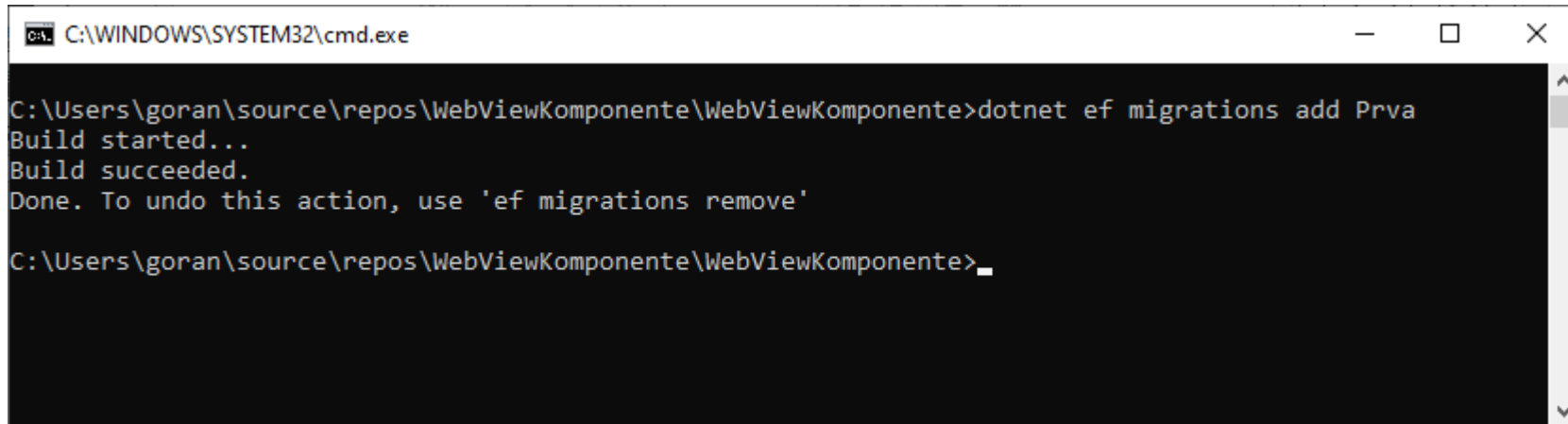
```
Command Prompt
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\goran>dotnet tool install --global dotnet-ef
You can invoke the tool using the following command: dotnet-ef
Tool 'dotnet-ef' (version '3.1.3') was successfully installed.

C:\Users\goran>
```

# Migracije

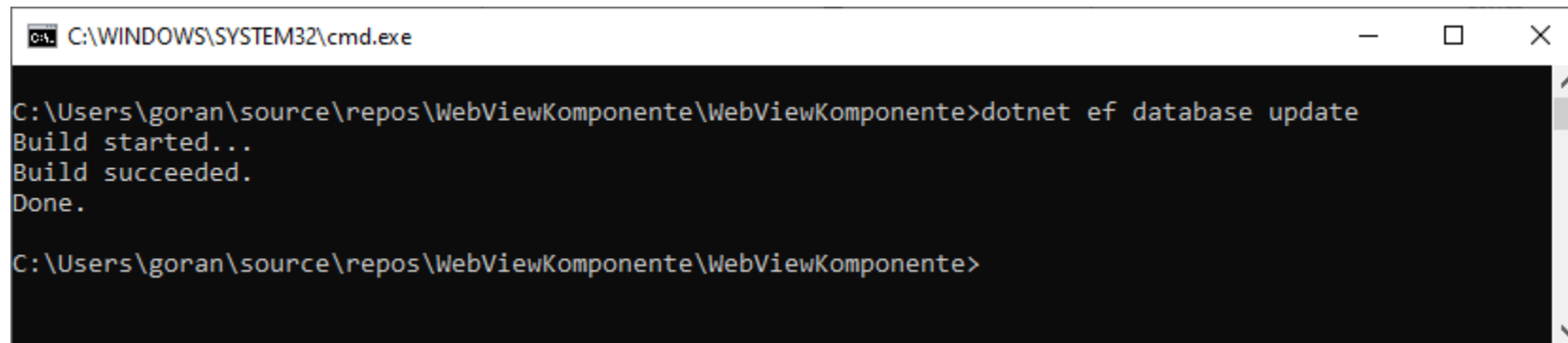
```
dotnet ef migrations add Prva
```



```
C:\WINDOWS\SYSTEM32\cmd.exe
C:\Users\goran\source\repos\WebViewKomponente\WebViewKomponente>dotnet ef migrations add Prva
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'
C:\Users\goran\source\repos\WebViewKomponente\WebViewKomponente>
```

# Primena migracije

dotnet ef database update



```
C:\WINDOWS\SYSTEM32\cmd.exe
C:\Users\goran\source\repos\WebViewKomponente\WebViewKomponente>dotnet ef database update
Build started...
Build succeeded.
Done.
C:\Users\goran\source\repos\WebViewKomponente\WebViewKomponente>
```

# View komponente

- View komponente ne koriste model binding kao parcijalni pogledi već zavise od podataka koji im se prosleđuju pri njihovom pozivu
- Često se poziva iz layout strane
- Klase komponente je obično izvedena iz klase `ViewComponent` i završava se sa **`ViewComponent`**
- Komponenta vraća pogled
- Logika View komponente se definiše u metodi **`Invoke`** ili **`InvokeAsync`** koja vraća **`ViewComponentResult`**
- Klasa može biti smeštena u bilo kom folderu aplikacije
- Obično kreiramo folder **`Components`** gde smeštamo klase komponenti

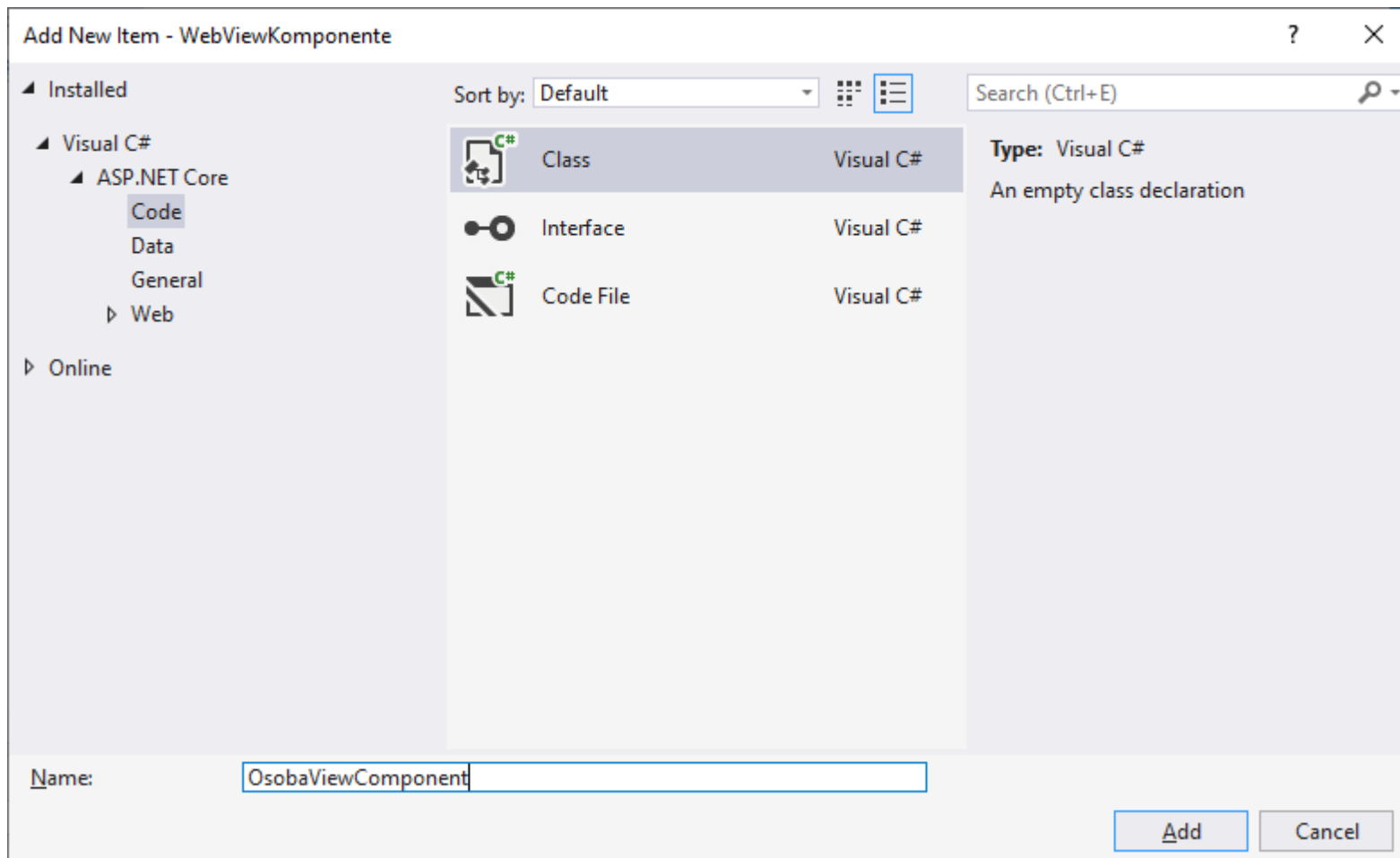
# Poziv View komponente

- Obično se pogled koji odgovara View komponenti smešta unutar **Views/Shared/Components/Ime\_Komponente** foldera
- Podrazumevano ime pogleda za komponentu je Default.cshtml
- View komponenta se može pozivati direktno iz kontrolera ali je uobičajeno pozivamo iz pogleda

```
public IActionResult Index()  
{  
    return ViewComponent("Osoba");  
}
```

```
@await Component.InvokeAsync("Ime_View_Komponente", {parametar1="vrednost1",  
parametar2="vrednost"})
```

# Klasa OsobaViewComponent



# Klasa OsobaViewComponent

```
using Microsoft.AspNetCore.Mvc;
```

```
public class OsobaViewComponent : ViewComponent
{
    private readonly OsobaContext db;

    public OsobaViewComponent(OsobaContext _db)
    {
        db = _db;
    }

    public IActionResult Invoke()
    {
        return View(db.Osobe.ToList());
    }

    //public async Task<IViewComponentResult> InvokeAsync()
    //{
    //    return View(await db.Osobe.ToListAsync());
    //}
}
```



# Kreiranje pogleda za komponentu

Add MVC View

View name:

Template:

Model class:

Data context class:

Options:

Create as a partial view

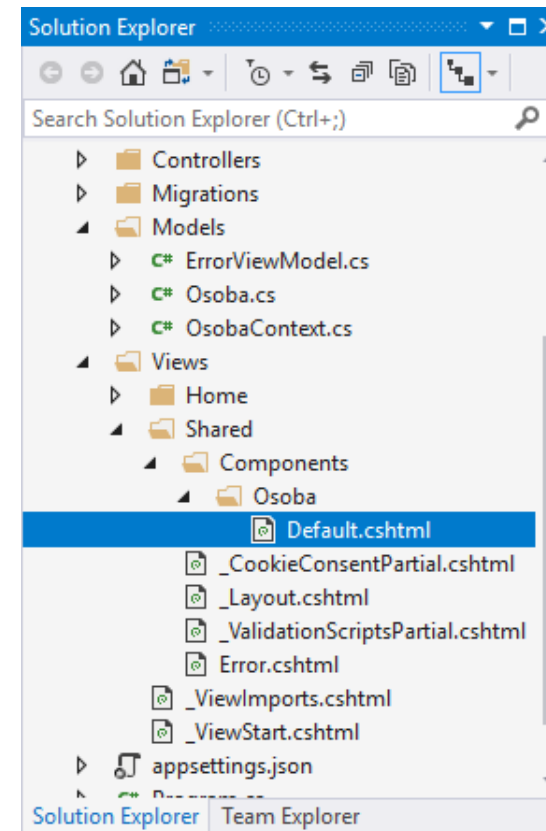
Reference script libraries

Use a layout page:

...

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

Add Cancel



# Default.cshtml

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

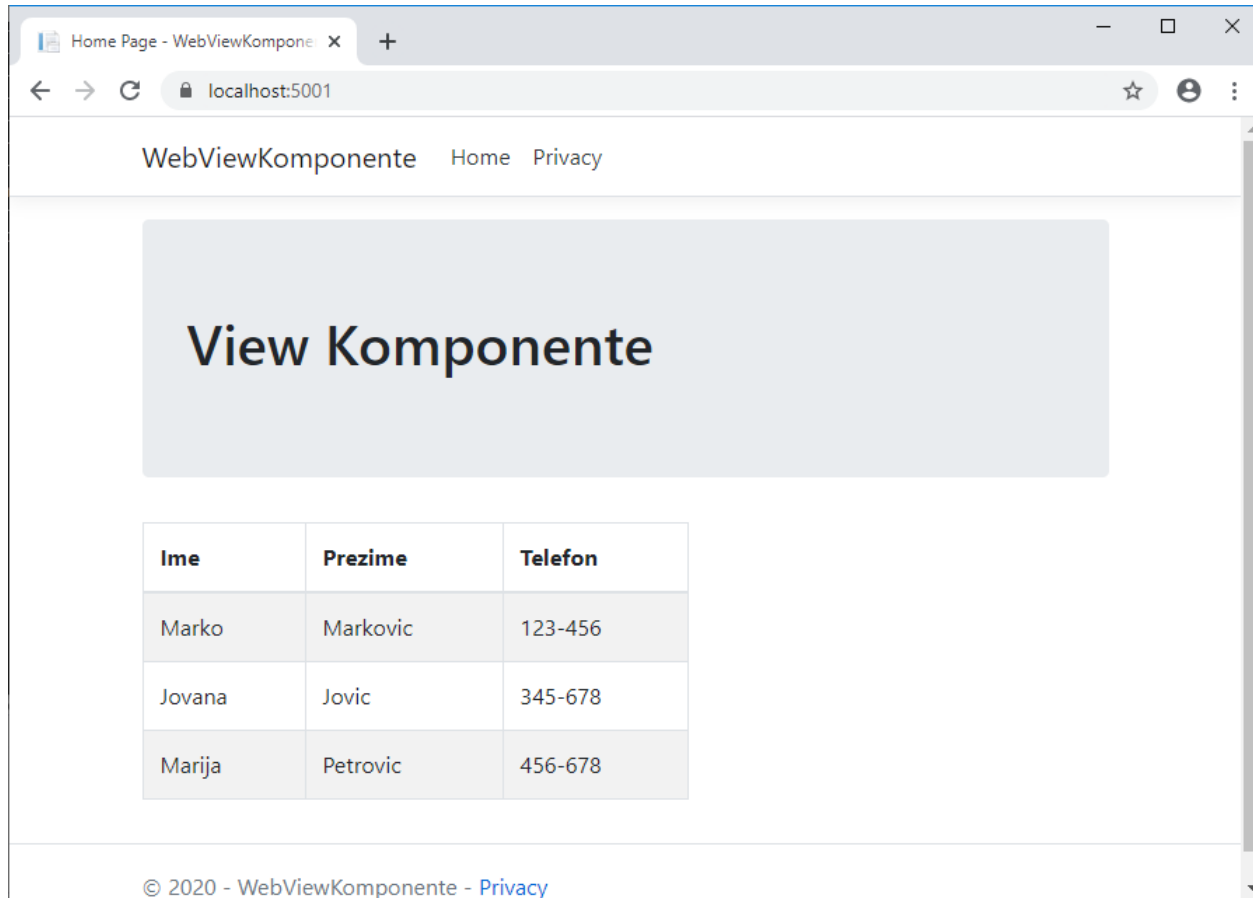
<table class="table table-bordered table-striped">
  <thead>
    <tr>
      <th>
        Ime
      </th>
      <th>
        Prezime
      </th>
      <th>
        Telefon
      </th>
    </tr>
  </thead>
  <tbody>
    @foreach (Osoba os in Model)
    {
      <tr>
        <td>
          @os.Ime
        </td>
        <td>
          @os.Prezime
        </td>
        <td>
          @os.Telefon
        </td>
      </tr>
    }
  </tbody>
</table>
```

# Index pogled klasa HomeController

```
<div class="jumbotron">
  <h1>View Komponente</h1>
</div>

<div class="row">
  <div class="col-md-7">
    @await Component.InvokeAsync("Osoba")
  </div>
</div>
```

# Index strana



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

- Browser tab: Home Page - WebViewKomponente
- Address bar: localhost:5001
- Page title: WebViewKomponente
- Page navigation: Home Privacy
- Main content area:
  - Header: View Komponente
  - Table with contact information:
- Footer: © 2020 - WebViewKomponente - [Privacy](#)

Ime	Prezime	Telefon
Marko	Markovic	123-456
Jovana	Jovic	345-678
Marija	Petrovic	456-678

# \_ViewImports.cshtml

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

Registracija View komponente kao tag helpera

# Poziv komponente

```
<div class="jumbotron">
  <h1>View Komponente</h1>
</div>

<div class="row">
  <div class="col-md-7">
    <vc:osoba />
  </div>
</div>
```

# Komponenta sa parametrom

```
public IActionResult Invoke(int id=0)
{
    IEnumerable<Osoba> listaOsoba = db.Osobe;
    if (id != 0)
    {
        listaOsoba = listaOsoba.Where(os => os.OsobaId > id);
    }
    return View(listaOsoba);
}
```

# Poziv komponente sa parametrima

```
<div class="row">
  <div class="col-md-7">
    @await Component.InvokeAsync("Osoba", new {id=1 })
  </div>
</div>
<div class="row">
  <div class="col-md-7">
    <vc:osoba id="1" />
  </div>
</div>
```



# Pitanje 1

Poziv komponente koja je predstavljena klasom `OsobaViewComponent` unutar nekog `cshtml` fajla vrši se korišćenjem koda:

- a. `@await Component.InvokeAsync("Osoba")`
- b. `@await Component.Invoke("OsobaViewComponent ")`
- c. `@await Component("Osoba")`

Odgovor: a

# Pitanje 2

Logika View komponente se definiše unutar sledeće metode klase View komponente

- a. InvokeComponent
- b. View
- c. InvokeAsync

Odgovor: c