Entity Framework Lab

Adapted from *Pro Entity Framework Core 2 for ASP.NET Core MVC*

Abstract

This lab is adapted from *Pro Entity Framework Core 2 for ASP.NET Core MVC* by Adam Freeman, chapter 2, ISBN-13 (pbk): 978-1-4842-3434-1. Please ensure that you purchase a copy of this book before using this lab.

1 Beginning the lab

1. First, check your version of .NET Core. Run `dotnet -version` in your Power Shell prompt. See Figure 1. If you do not have a current version of `dotnet`, please install a current version of the .NET Core SDK.

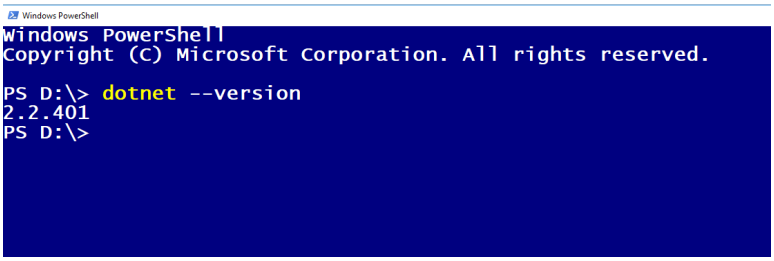

Figure 1: Checkihg `dotnet` version

2. Start Visual Studio. Click the `Create a new project` button. See Figure 2.

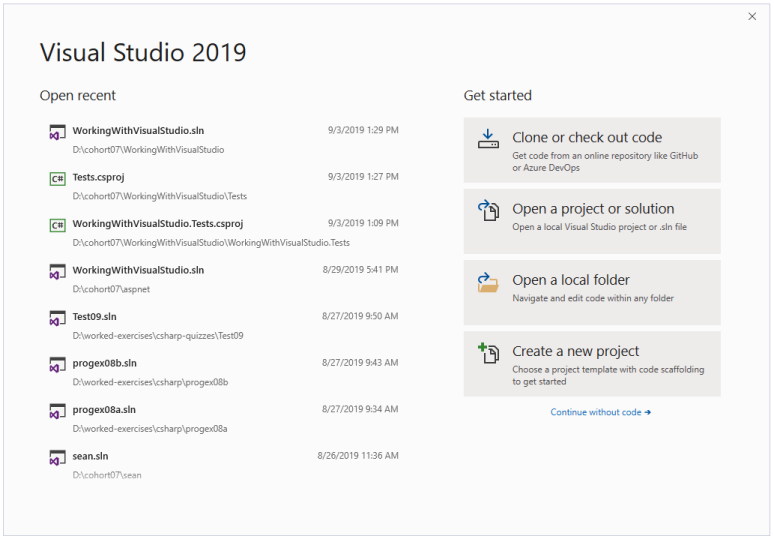

Figure 2: Start Visual Studio

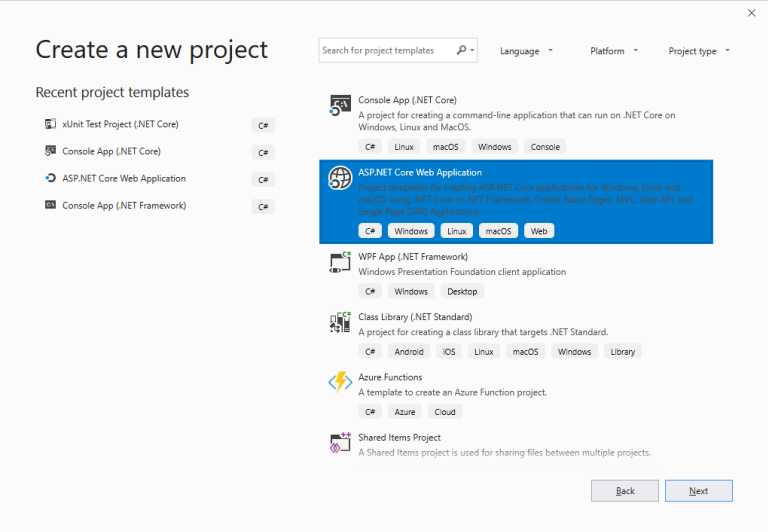

Figure 3: ASP.NET Core Web Application

3. Select the `ASP.NET Core Web Application` and click `Next`. See figure 3.
4. Name your project `EFPartyInvites`, save it in your appropriate lab folder, and click `Create`. See Figure 4.

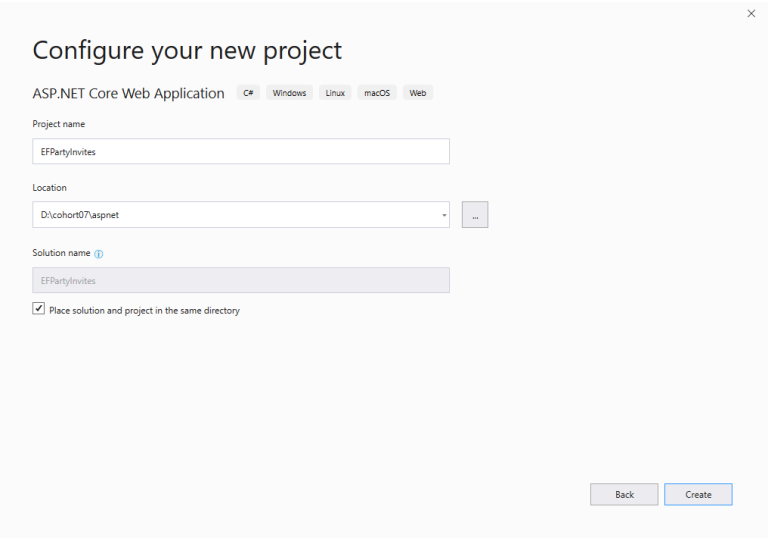

Figure 4: Naming and saving your project

5. Select the `Empty` template, ensure that `.NET Core` and `ASP.NET Core 2.2` are selected, that Authentication is set to `No Authentication`, and that Docker support is not enabled. See Figure 5. Click `Create`.
6. To install LibMan, run `dotnet tool install -global Microsoft.Web.LibraryManager.Cli` in your Power Shell prompt. See figure 6.
7. To install Bootstrap, navigate to the project folder (which contains the Startup.cs file) and run the following command in your Power Shell prompt: `libman install twitter-bootstrap@4.0.0 -destination wwwroot/lib/bootstrap/dist -provider cdnjs`. See Figure 7.
8. Bootstrap installs. See Figure 8.

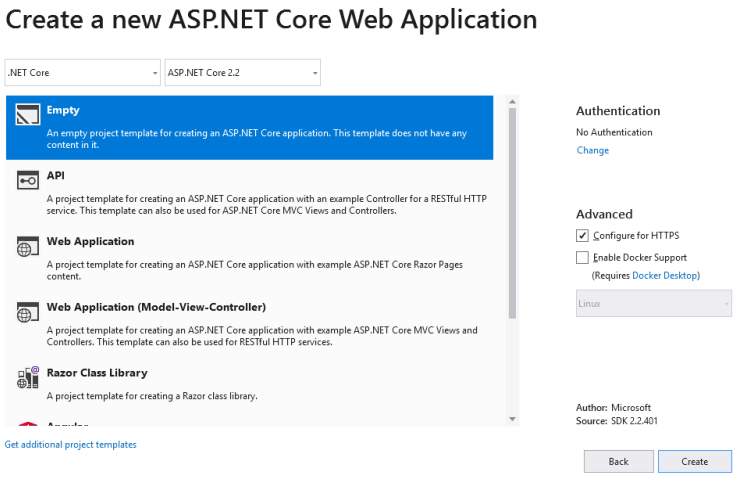

Figure 5: Select the `Empty` template

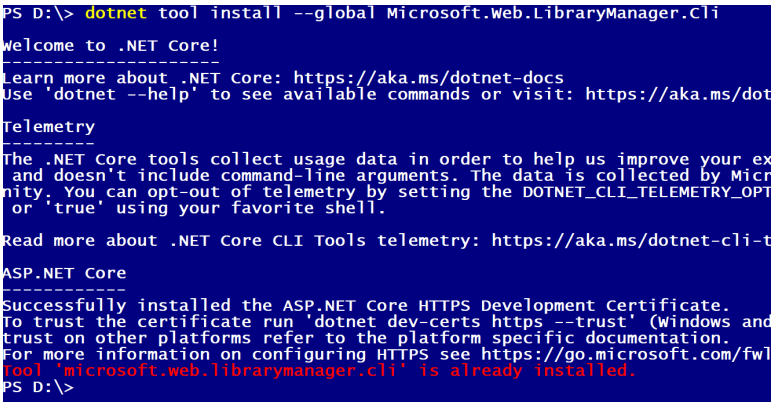

Figure 6: Install LibMan

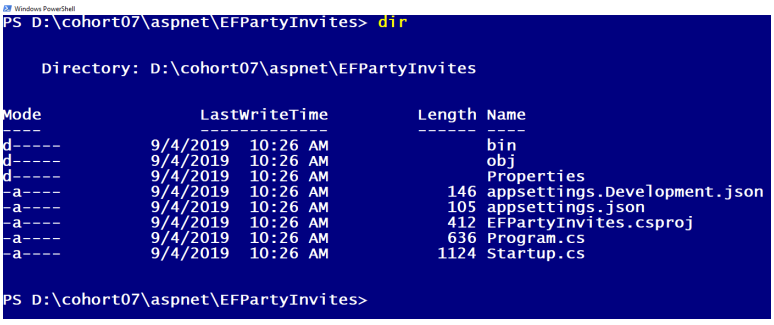

Figure 7: Installing Bootstrap

2 Creating the Model and Context

9. Create a `Models` folder by right clicking on the `EFPartyInvites` project and selecting `Add New Folder`. Name the folder `Models`. ▶

```
PS D:\cohort07\aspnet\EFPartyInvites> libman install twitter-bootstrap@4.0.0 --destination wwwroot/lib/bo
otstrap/dist --provider cdnjs
wwwroot/lib/bootstrap/dist/css/bootstrap-grid.css written to disk
wwwroot/lib/bootstrap/dist/css/bootstrap-grid.css.map written to disk
wwwroot/lib/bootstrap/dist/css/bootstrap-grid.min.css written to disk
wwwroot/lib/bootstrap/dist/css/bootstrap-grid.min.css.map written to disk
wwwroot/lib/bootstrap/dist/css/bootstrap-reboot.css written to disk
wwwroot/lib/bootstrap/dist/css/bootstrap-reboot.css.map written to disk
wwwroot/lib/bootstrap/dist/css/bootstrap-reboot.min.css written to disk
wwwroot/lib/bootstrap/dist/css/bootstrap-reboot.min.css.map written to disk
wwwroot/lib/bootstrap/dist/css/bootstrap.css written to disk
wwwroot/lib/bootstrap/dist/css/bootstrap.css.map written to disk
wwwroot/lib/bootstrap/dist/css/bootstrap.min.css written to disk
wwwroot/lib/bootstrap/dist/css/bootstrap.min.css.map written to disk
wwwroot/lib/bootstrap/dist/js/bootstrap.bundle.js written to disk
wwwroot/lib/bootstrap/dist/js/bootstrap.bundle.js.map written to disk
wwwroot/lib/bootstrap/dist/js/bootstrap.bundle.min.js written to disk
wwwroot/lib/bootstrap/dist/js/bootstrap.bundle.min.js.map written to disk
wwwroot/lib/bootstrap/dist/js/bootstrap.js written to disk
wwwroot/lib/bootstrap/dist/js/bootstrap.js.map written to disk
wwwroot/lib/bootstrap/dist/js/bootstrap.min.js written to disk
wwwroot/lib/bootstrap/dist/js/bootstrap.min.js.map written to disk
Installed library "twitter-bootstrap@4.0.0" to "wwwroot/lib/bootstrap/dist"
PS D:\cohort07\aspnet\EFPartyInvites>
```

Figure 8: Bootstrap installs

10. Create a `GuestResponse` class by right clicking on the `Models` folder and selecting `Add` ▶ `Class`. Name the class `GuestResponse`. See Figure 9. Click `Add`. Edit `GuestResponse` as shown in Listing 1.

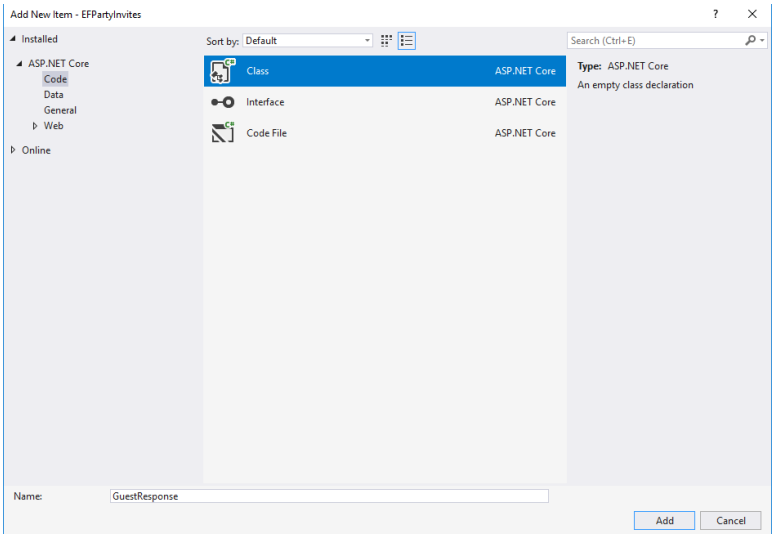

Figure 9: Add the `GuestResponse` class

Listing 1: `GuestRespose.cs`

```
namespace EFPartyInvites.Models
{
    public class GuestResponse
    {
        public long Id { get; set; }
        public string Name { get; set; }
        public string Email { get; set; }
        public string Phone { get; set; }
        public bool? WillAttend { get; set; }
    }
}
```

11. Create a `DataContext` class by right clicking on the `Models` folder and selecting `Add` ▶ `Class`. Name the class `DataContext`. Click `Add`. Edit `DataContext` as shown in Listing 2.

Listing 2: `DataContext.cs`

```
using Microsoft.EntityFrameworkCore;

namespace EFPartyInvites.Models
{
    public class DataContext : DbContext
    {
        public DataContext(DbContextOptions<DataContext> options) : base(options)
        { }

        public DbSet<GuestResponse> Responses { get; set; }
    }
}
```

12. Create a `Controllers` folder by right clicking on the `EFPartyInvites` project and selecting `Add` ► `New Folder`. Name the folder `Controllers`.
13. Create a `HomeController` class by right clicking on the `Controllers` folder and selecting `Add` ► `Class`. Name the class `HomeController`. Click `Add`. Edit `HoeController` as shown in Listing 3.

Listing 3: `HomeController.cs`

```
using Microsoft.AspNetCore.Mvc;
using EFPartyInvites.Models;
using System.Linq;

namespace EFPartyInvites.Controllers
{
    public class HomeController : Controller
    {
        private DataContext context;
        public HomeController(DataContext ctx) => context = ctx;
        public IActionResult Index() => View();
        public IActionResult Respond() => View();

        [HttpPost]
        public IActionResult Respond(GuestResponse response)
        {
            context.Responses.Add(response);
            context.SaveChanges();
            return RedirectToAction(nameof(Thanks),
            new { Name = response.Name, WillAttend = response.WillAttend });
        }

        public IActionResult Thanks(GuestResponse response)
        {
            return View(response);
        }

        public IActionResult ListResponses()
        {
            return View(context.Responses.OrderByDescending(r => r.WillAttend));
        }
    }
}
```

14. Create a `Views` folder by right clicking on the `EFPartyInvites` project and selecting `Add` ► `New Folder`. Name the folder `Views`.

15. Create a `Views/Home` folder by right clicking on the `Views` folder and selecting `Add` ▶ `New Folder`. Name the folder `Home`.
16. Create a `_Layout.cshtml` Razor Layout Page in `Views/Home` by right clicking the `Home` folder and selecting `Add` ▶ `New Item` ▶ `Web` ▶ `Razor Layout`. Name the page `_Layout.cshtml` and edit thee page as shown in Listing 4.

Listing 4: `_Layout.cshtml`

```
<!DOCTYPE html>

<html>
    <head>
        <meta name="viewport" content="width=device-width" />
        <title>Party Invites</title>
        <link rel="stylesheet" href="/lib/bootstrap/dist/css/bootstrap.css" />
    </head>
    <body>
        @RenderBody()
    </body>
</html>
```

17. Create a `_ViewStart.cshtml` page in `Views` by right clicking the `Views` folder and selecting `Add` ▶ `New Item` ▶ `Web` ▶ `Razor View Start`. See Figure 10. Name the page `_ViewStart.cshtml` and edit thee page as shown in Listing 5.

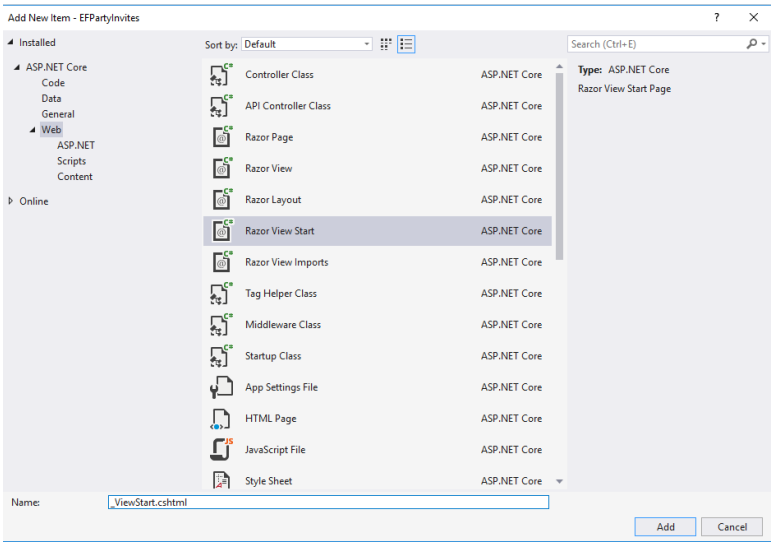

Figure 10: `_ViewStart.cshtml`

Listing 5: `_ViewStart.cshtml`

```
@{
    Layout = "_Layout";
}
```

18. Create a `Index.cshtml` Razor View in `Views/Home` by right clicking the `Home` folder and selecting `Add` ▶ `New Item` ▶ `Web` ▶ `Razor View`. See Figure 11. Name the page `Index.cshtml` and edit the page as shown in Listing 6.

Listing 6: `Index.cshtml`

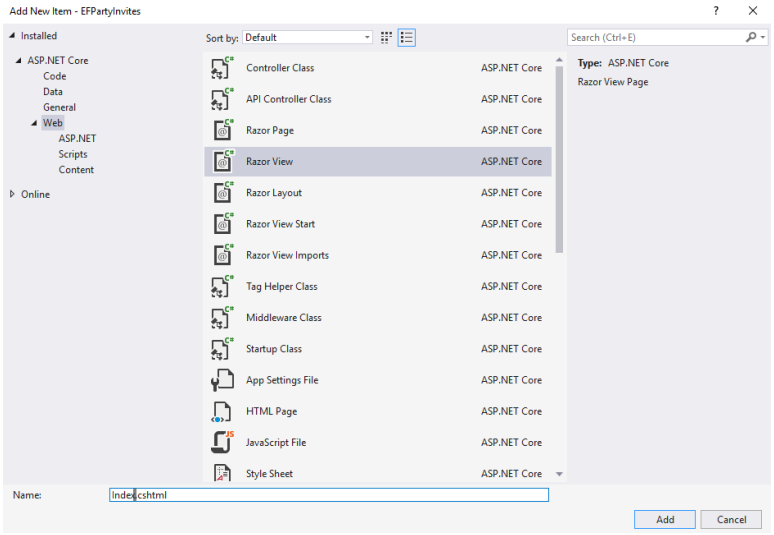

Figure 11: `Index.cshtml`

```
<div class="text-center m-4">
    <h3>We're going to have an exciting party!</h3>
    <h4>And you are invited</h4>
    <a class="btn btn-primary" asp-action="Respond">RSVP Now</a>
</div>
```

19. Create a `Respond.cshtml` Razor View in `Views/Home` by right clicking the `Home` folder and selecting `Add` ▶ `New Item` ▶ `Web` ▶ `Razor View`. Name the page `Respond.cshtml` and edit the page as shown in Listing 7.

Listing 7: `Respond.cshtml`

```
@model GuestResponse

<div class="bg-primary p-2 text-white text-center">
    <h2>RSVP</h2>
</div>
<form asp-action="Respond" method="post" class="m-4">
    <div class="form-group">
        <label>Your Name</label>
        <input asp-for="Name" class="form-control" />
    </div>
    <div class="form-group">
        <label>Your Email</label>
        <input asp-for="Email" class="form-control" />
    </div>
    <div class="form-group">
        <label>Your Phone Number</label>
        <input asp-for="Phone" class="form-control" />
    </div>
    <div class="form-group">
        <label>Will You Attend?</label>
        <select asp-for="WillAttend" class="form-control">
            <option value="">Choose an option</option>
            <option value="true">Yes, I'll be there</option>
            <option value="false">No, I can't come</option>
```

```
        </select>
    </div>
    <div class="text-center">
        <button type="submit" class="btn␣btn-primary">Submit RSVP</button>
    </div>
</form>
```

20. Create a `Thanks.cshtml` Razor View in `Views/Home` by right clicking the `Home` folder and selecting `Add` ▶ `New Item` ▶ `Web` ▶ `Razor View`. Name the page `Thanks.cshtml` and edit the page as shown in Listing 8.

Listing 8: `Thanks.cshtml`

```
@model GuestResponse

<div class="text-center␣mt-3">
    <h1>Thank you, @Model.Name!</h1>
    @if (Model.WillAttend == true)
    {
        <div>
            It's great that you're coming. The drinks are already in the fridge!
        </div>
    }
    else
    {
        <div>
            Sorry to hear that you can't make it, but thanks for letting us know.
        </div>
    }
    Click <a asp-action="ListResponses">here</a> to see who is coming.
</div>
```

21. Create a `ListResponses.cshtml` Razor View in `Views/Home` by right clicking the `Home` folder and selecting `Add` ▶ `New Item` ▶ `Web` ▶ `Razor View`. Name the page `ListResponses.cshtml` and edit the page as shown in Listing 9.

Listing 9: `ListResponses.cshtml`

```
@model IEnumerable<GuestResponse>

<h3 class="bg-primary␣p-2␣text-white␣text-center">
    Here is the list of people who have
    responded
</h3>
<div class="container-fluid">
    <div class="row␣p-1">
        <div class="col␣font-weight-bold">Name</div>
        <div class="col␣font-weight-bold">Email</div>
        <div class="col␣font-weight-bold">Phone</div>
        <div class="col␣font-weight-bold">Attending</div>
    </div>
    @foreach (GuestResponse r in Model)
    {
        <div class="row␣p-1">
            <div class="col">@r.Name</div>
            <div class="col">@r.Email</div>
            <div class="col">@r.Phone</div>
            <div class="col">@(r.WillAttend == true ? "Yes" : "No")</div>
```

```
        </div>
    }
</div>
```

22. Create a `_ViewImmports.cshtml` page in `Views` by right clicking the `Views` folder and selecting `Add` ▶ `New Item` ▶ `Web` ▶ `Razor View Imports`. See Figure 12. Name the page `_ViewIports.cshtml` and edit thee page as shown in Listing 10.

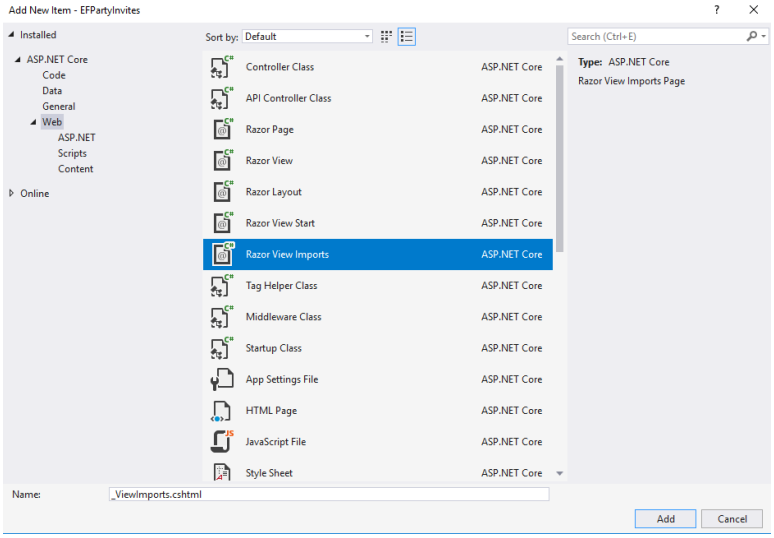

Figure 12: `_ViewIports.cshtml`

Listing 10: `_ViewIports.cshtml`

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

,

3 Configuring Entity Framework Core

23. Edit the project `csproj` file as shown in Listing 11. To edit the file, right click on the project and select `Edit EFPartyInvites.csproj`.

Listing 11: `csproj` file

```
<Project Sdk="Microsoft.NET.Sdk.Web">

  <PropertyGroup>
    <TargetFramework>netcoreapp2.2</TargetFramework>
    <AspNetCoreHostingModel>InProcess</AspNetCoreHostingModel>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="Microsoft.AspNetCore.App" />
    <PackageReference Include="Microsoft.AspNetCore.Razor.Design" Version="2.2.0"
        PrivateAssets="All" />
    <DotNetCliToolReference Include="Microsoft.EntityFrameworkCore.Tools.DotNet"
                            Version="2.0.1" />
  </ItemGroup>
```

```
</Project>
```

24. Next, configure `appsettings.json` by editing the file by adding the connection string dictionary as shown in Listing 12. If necessary, you can add this file by right clicking on the project and selecting `Add` ▶ `New Item` ▶ `Web` ▶ `ASP.NET` ▶ `App Settings File` and name the file `appsettings.json`. See Figure 13. Do not forget that the elements in JSON are comma separated, so don't forget the prededing comma.

Listing 12: `appsettings.json`

```
"ConnectionStrings": {
    "DefaultConnection":"Server=(localdb)\MSSQLLocalDB;Database=PartyInvites"
}
```

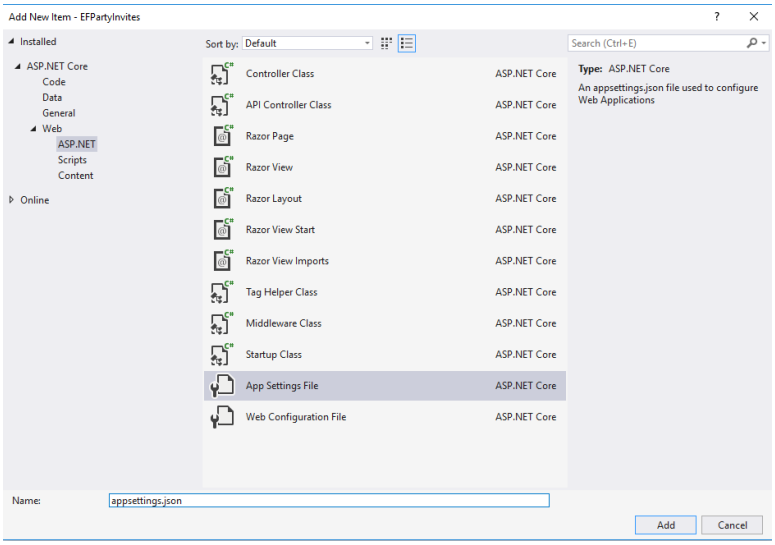

Figure 13: Addition to `appsettings.json`

25. Edit the `Startup.cs` file as shown in Listing 13.

Listing 13: Edits to `Startup.cs`

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Http;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Configuration;
using EFPartyInvites.Models;

namespace EFPartyInvites
{
    public class Startup
    {
        public Startup(IConfiguration config) => Configuration = config;
```

```
        public IConfiguration Configuration  get;
        public void ConfigureServices(IServiceCollection services)
        {
            services.AddMvc();
            string conString =
                Configuration["ConnectionStrings:DefaultConnection"];
            services.AddDbContext<DataContext>(options =>
                options.UseSqlServer(conString));
        }
        public void Configure(IApplicationBuilder app, IHostingEnvironment env)
        {
            app.UseDeveloperExceptionPage();
            app.UseStatusCodePages();
            app.UseStaticFiles();
            app.UseMvcWithDefaultRoute();
        }
    }
}
```

26. In your Power Shell prompt, run the following two commands. Ensure that you are in the application home directory, the same one that contains `Startup.cs` and `Program.cs`. See Figures 14 and 15.

```
dotnet ef migrations add Initial
dotnet ef database update
```

```
PS D:\cohort07\aspnet\EFPartyInvites> dotnet --diagnostics ef migrations add Initial
Telemetry is: Enabled
Running C:\Program Files\dotnet\dotnet.exe "C:\Program Files\dotnet\sdk\2.2.401\DotnetTools\dotnet-ef\2.2
.6\tools\netcoreapp2.2\any\dotnet-ef.dll" migrations add Initial
Process ID: 30232
Telemetry is: Enabled
Telemetry is: Enabled
info: Microsoft.EntityFrameworkCore.Infrastructure[10403]
      Entity Framework Core 2.2.6-servicing-10079 initialized 'DataContext' using provider 'Microsoft.Ent
ityFrameworkCore.SqlServer' with options: None
Done. To undo this action, use 'ef migrations remove'
PS D:\cohort07\aspnet\EFPartyInvites>
```

Figure 14: Running initial migration

```
PS D:\cohort07\aspnet\EFPartyInvites> dotnet ef database update
info: Microsoft.EntityFrameworkCore.Infrastructure[10403]
      Entity Framework Core 2.2.6-servicing-10079 initialized 'DataContext' using provider 'Microsoft.Ent
ityFrameworkCore.SqlServer' with options: None
```

Figure 15: Updating database

Here is a copy of the verbose output.

```
PS D:\cohort07\aspnet\EFPartyInvites> dotnet ef database update
info: Microsoft.EntityFrameworkCore.Infrastructure[10403]
      Entity Framework Core 2.2.6-servicing-10079 initialized 'DataContext' using provider
      'Microsoft.EntityFrameworkCore.SqlServer' with options: None
info: Microsoft.EntityFrameworkCore.Database.Command[20101]
      Executed DbCommand (861ms) [Parameters=[], CommandType='Text', CommandTimeout='60']
      CREATE DATABASE [EFPartyInvites];
info: Microsoft.EntityFrameworkCore.Database.Command[20101]
      Executed DbCommand (95ms) [Parameters=[], CommandType='Text', CommandTimeout='60']
      IF SERVERPROPERTY('EngineEdition') <> 5
```

```
      BEGIN
          ALTER DATABASE [EFPartyInvites] SET READ_COMMITTED_SNAPSHOT ON;
      END;
info: Microsoft.EntityFrameworkCore.Database.Command[20101]
      Executed DbCommand (8ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
      CREATE TABLE [__EFMigrationsHistory] (
          [MigrationId] nvarchar(150) NOT NULL,
          [ProductVersion] nvarchar(32) NOT NULL,
          CONSTRAINT [PK___EFMigrationsHistory] PRIMARY KEY ([MigrationId])
      );
info: Microsoft.EntityFrameworkCore.Database.Command[20101]
      Executed DbCommand (4ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
      SELECT OBJECT_ID(N'[__EFMigrationsHistory]');
info: Microsoft.EntityFrameworkCore.Database.Command[20101]
      Executed DbCommand (2ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
      SELECT [MigrationId], [ProductVersion]
      FROM [__EFMigrationsHistory]
      ORDER BY [MigrationId];
info: Microsoft.EntityFrameworkCore.Migrations[20402]
      Applying migration '20190904214730_Initial'.
Applying migration '20190904214730_Initial'.
info: Microsoft.EntityFrameworkCore.Database.Command[20101]
      Executed DbCommand (2ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
      CREATE TABLE [Responses] (
          [Id] bigint NOT NULL IDENTITY,
          [Name] nvarchar(max) NULL,
          [Email] nvarchar(max) NULL,
          [Phone] nvarchar(max) NULL,
          [WillAttend] bit NULL,
          CONSTRAINT [PK_Responses] PRIMARY KEY ([Id])
      );
info: Microsoft.EntityFrameworkCore.Database.Command[20101]
      Executed DbCommand (3ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
      INSERT INTO [__EFMigrationsHistory] ([MigrationId], [ProductVersion])
      VALUES (N'20190904214730_Initial', N'2.2.6-servicing-10079');
Done.
PS D:\cohort07\aspnet\EFPartyInvites>
```

27. Start the application and run it. Correct any errors. RSVP a guest, close the application, reopen it, and REVP anotheer guest. Everyone should be there.