# EFCoreExistingDatabase ASPDotNETCore

**Jacob Duenke**
*May 9, 2019*

Tutorial from: Getting Started with EF Core on ASP.NET Core with an Existing Database ## Prerequisites

Install the following: - Visual Studio 2017 version 15.7 or later with these workloads: - ASP.NET and web development (under Web & Cloud) - .NET Core cross-platform development (under Other Toolsets) - .NET Core 2.1 SDK or higher - Have completed the previous exercise, ***EFCoreNewDatabase***

## Create a new project

- Open Visual Studio
- Select ***Create a new project***
  - Select ***ASP.NET Core Web Application***
    - Enter ***FriendsApp2*** in the *Project name* field
- In the Create a new ASP.NET Core Web Application screen:
  - Select ***.NET Core*** and ***ASP.NET Core 2.0*** or higher
  - Select the ***Empty*** project template
  - Make sure that Authentication is set to ***No Authentication***

## Install Entity Framework Core

- For this tutorial, you don't have to install a provider package because the tutorial uses SQL Server.
  - The SQL Server provider package is included in the Microsoft.AspnetCore.App metapackage.

## Reverse engineer your model

- Select ***Tools > NuGet Package Manager > Package Manager Console***
  - Run the following command: `Create a model from an existing database Scaffold-DbContext "Server=(localdb)\mssqllocaldb;Database=Friendship;Trus Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models`

## Register the context with dependency injection

- In the ***Startup.cs*** file, add the following: ***Startup.cs > using statements***

```
using FriendsApp2.Models;
using Microsoft.EntityFrameworkCore;
```

***Startup.cs > ConfigureServices()***

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddMvc();

    string connection = @"Server=(localdb)\\mssqllocaldb;Database=Friendship;
                                Trusted_Connection=True;ConnectRetryCount=0";
        services.AddDbContext<FriendshipContext>
            (options => options.UseSqlServer(connection));
}
```

***Startup.cs > Configure***

```
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    app.UseStaticFiles();
    app.UseMvc(routes =>
    {
        routes.MapRoute(
            name: "Default",
            template: "{controller}/{action}/{id?}",
            defaults: new { controller = "People", action = "Index" }
        );
    });
}
```

## Create a controller and views

- Right-click on the ***FriendsApp2*** Project Name (again, not the Solution Name above) and select ***Add > New Folder***
  - Enter ***Controllers*** as the folder name
- Right-click on the ***Controllers*** folder and select ***Add > Controller***
  - Select ***MVC Controller with views, using Entity Framework***
    - Set Model class to ***People*** and Data context class to ***FriendshipContext***
    - Make sure that ***Use a Layout Page*** is ***not*** selected.

## Import ASP.NET Core Tag Helpers

- Right-click on the ***Views*** folder and select ***Add > New Item***
  - On the left, navigate to ***ASP.NET Core > Web > ASP.NET***
    - Select ***Razor View Imports*** and leave the default ***_ViewImports.cshtml*** file name
- Replace the contents of the file with the following code: `ViewImports.cshtml code @addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers`

## Run the application

- On your keyboard, press CTRL + F5 to start without debugging
  - Displayed in the browser is your previous list of Persons, or Friends
- Click Create New to add your first Friend
  - Explore the rest of the options available

## (Optional) Add style to your application

- Select ***Tools > NuGet Package Manager > Package Manager Console***
- Run the following command as one single line: `Bootstrap installation commands libman install twitter-bootstrap@4.0.0-alpha.6 --destination FriendsApp2/wwwroot/lib/bootstrap/dist --provider cdnjs` ***install location may be at the .sln level... move into the FriendsApp project folder***
- Navigate to each individual view .cshtml file
  - In the < head > section, add ***< link href="~/lib/bootstrap/dist/css/bootstrap.css" rel="stylesheet" />***
  - In the < body > tag, add ***style="margin: 20px; padding: 20px; box-shadow: 0 1px 5px rgba(104, 104, 104, 0.8);"***
  - In every < a asp-action > tag, add ***class="btn btn-info"***
  - In every < table > tag, add ***class="table table-striped"***