**安装环境**

**Visual Studio**

Visual Studio 2017 15.7.3 版或更高版本

**.NET Core**

安装 .NET Core Runtime（运行 .net core 程序）和 .NET Core SDK （开发 .net core 程序）

.net core下载地址：https://dotnet.microsoft.com/download

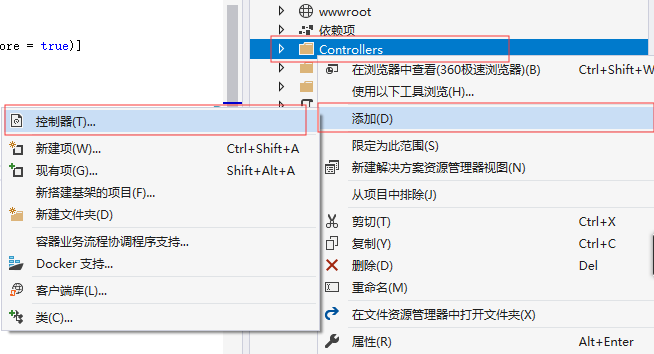
注：注意版本

**创建 Web 应用**

在Vs创建Asp.net core Web应用程序，选择创建MVC应用

**控制器**

**添加控制器**



这里新建了一个My控制器



**添加控制器方法**

在My控制器中添加方法，这里添加了一个MyAction方法

public class MyController : Controller

{

    public IActionResult Index () {

        return View ();

    }

    public IActionResult MyAction () {

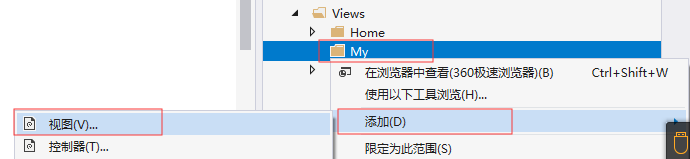
        return View ();

    }

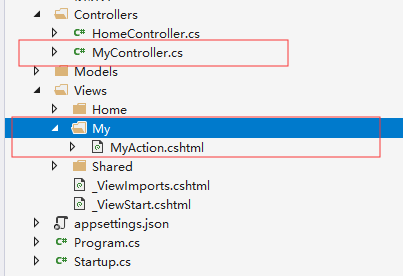
}

**添加视图**

在View文件夹下新建与控制器同名的文件夹，在该文件件在新建与方法同名的视图



最终文件目录如下



至此，访问浏览器地址“站点地址+/My/MyAction”就可以访问我们的视图了

**应用EntityFramewordCore**

EFCore包：Microsoft.EntityFrameworkCore.SqlServer，但我们不需要安装，原因如下：

由于我们生成的MVC项目依赖与包Microsoft.AspNetCore.App，而包Microsoft.AspNetCore.App又依赖于Microsoft.EntityFrameworkCore.SqlServer，所以我们不需要安装包Microsoft.EntityFrameworkCore.SqlServer

**新建数据库**

这里新建MyDB数据库

**新建一个实体**

这新建一个MyEntity实体

public class MyEntity

{

    [Key]

    public int Id { get; set; }

    public string Name { get; set; }

}

**新建数据库上下文DBContext**

这里新建MyDBContext

public class MyDbContext : DbContext

{

    public MyDbContext (DbContextOptions<MyDbContext> options) : base (options) {

}

    public DbSet<MyEntity> MyEntities { get; set; }

}

**在StartUp中添加上下文（向依赖容器注入上下文）**

public void ConfigureServices (IServiceCollection services)

{

    // 注入上下文

    // 如果对依赖注入有了解可以看如下解释

    // 这里实际上向依赖注入容器注入了 MyDbContext 和 DbContextOptions<MyDbContext>

    // MyDbContext 为临时注入（Transient）

    // DbContextOptions<MyDbContext> 为单例注入（Singleton）

    services.AddDbContext<MyDbContext> ((DbContextOptionsBuilder builder) => {

        builder.UseSqlServer (@"数据库连接字符串");

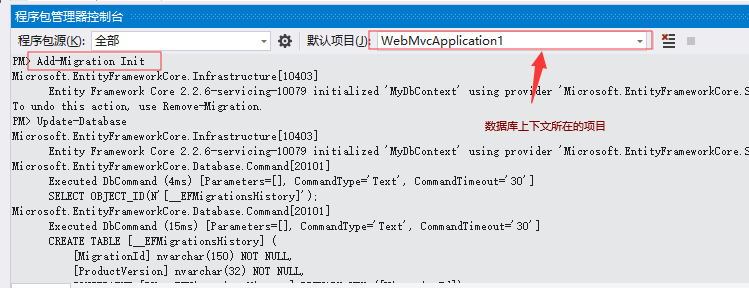
    });

    ...

}

**利用包控制台初始化数据数据库**

输入Add-Migration Init生成迁移文件



输入Update-Database将迁移文件更新到数据库



**修改我们的MyController文件**

public class MyController : Controller {

    private MyDbContext \_myDbContext { get; set; }

    // 通过依赖容器注入数据库上下文

    public MyController (MyDbContext myDbContext) {

        \_myDbContext = myDbContext;

}

    public IActionResult Index () {

        return View ();

}

    // 访问数据库上下文

    public IActionResult MyAction () {

        if (!\_myDbContext.MyEntities.Any ()) {

            \_myDbContext.MyEntities.Add (new MyEntity () { Name = "MyName" });

            \_myDbContext.SaveChanges ();

        }

        ViewData["Name"] = \_myDbContext.MyEntities.FirstOrDefault ().Name;

        return View ();

    }

}