# 添加controller与Views

1. 建立的一个空core web项目可以看出来，里面只包含了最核心的内容，运行后，在页面输出了一个hello word！！

接下来我们在项目新增一个Controllers和Views文件夹，您肯定已经知道我接下来要做什么操作了。

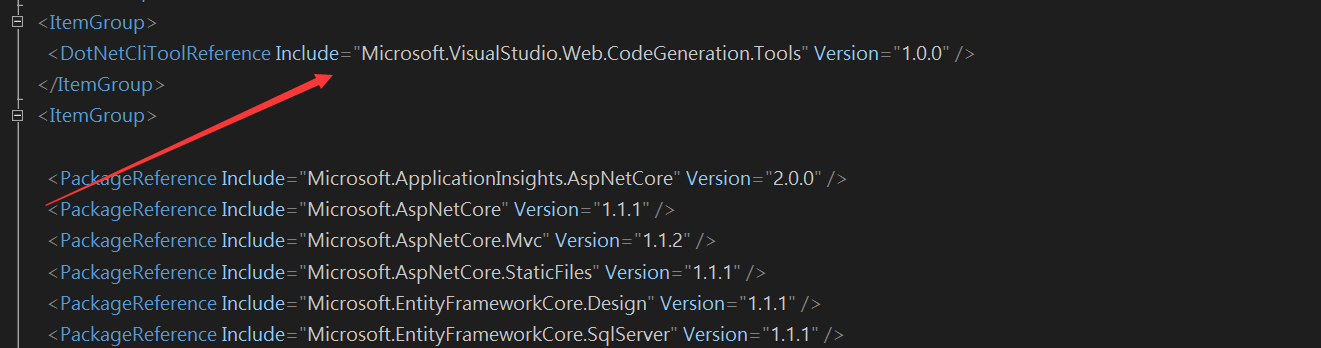
在Controllers添加一个控制器是会失败的,但vs会生成一个ScaffoldingReadMe.txt文档告诉你还要进行什么操作.

**1. Add Scaffolding CLI tool to the project:**

<ItemGroup>

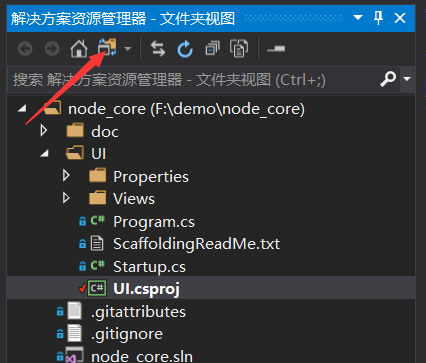
<DotNetCliToolReference Include="Microsoft.VisualStudio.Web.CodeGeneration.Tools" Version="1.0.0" />

</ItemGroup>



如图所示,在在项目的csproj文件添加添加此行配置.保存,重新生成解决方案.

这里有个vs2017的新功能,切换至文件夹模式.



双击,csproj即可进行相关的操作.修改完成后,切换回去解决方案模式,重新编译项目即可.

1. **Suggested changes to Startup class:**

**2.1 Add a constructor:**

public IConfigurationRoot Configuration { get; }

public Startup(IHostingEnvironment env)

{

var builder = new ConfigurationBuilder()

.SetBasePath(env.ContentRootPath)

.AddJsonFile("appsettings.json", optional: true, reloadOnChange: true)

.AddJsonFile($"appsettings.{env.EnvironmentName}.json", optional: true)

.AddEnvironmentVariables();

Configuration = builder.Build();

}

**2.2 Add MVC services:**

public void ConfigureServices(IServiceCollection services)

{

// Add framework services.

services.AddMvc();//添加MVC 服务

}

**2.3 Configure web app to use use Configuration and use MVC routing:**

public void Configure(IApplicationBuilder app, IHostingEnvironment env, ILoggerFactory loggerFactory)

{

loggerFactory.AddConsole(Configuration.GetSection("Logging"));

loggerFactory.AddDebug();

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

else

{

app.UseExceptionHandler("/Home/Error");

}

app.UseStaticFiles();

app.UseMvc(routes =>

{

routes.MapRoute(

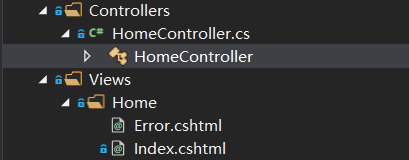
name: "default",

template: "{controller=Home}/{action=Index}/{id?}");

});

}

跟着上面的步骤逐一完成后,就可以添加控制器.



接下来就跟普通的MVC差不多了.