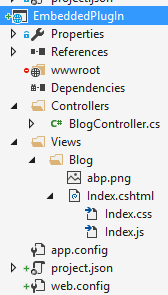
**创建嵌入文件**

首先,我们应当创建一个文件并把它标志为嵌入式资源,任何程序集都可以包含嵌入式资源文件.步骤依项目类型而变.

**xproj/project.json 格式**

假设我们有个名为EmbeddedPlugIn的项目:

****

把Views文件夹的所有文件标志为嵌入式资源,我们可以添加一个如下配置到project.json:

"buildOptions": {

"embed": {

"include": [

"Views/\*\*/\*.\*"

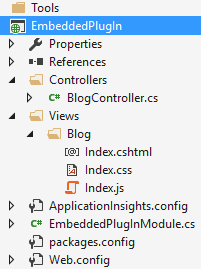
]

}

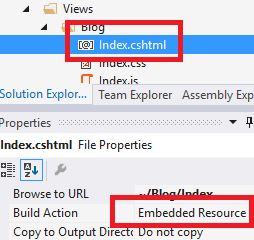
}

**csproj 格式**

假设我们有个名为EmbeddedPlugIn的项目:



我选中Index.cshtml文件,转到属性窗口(快捷键为F4)并修改它的"创建操作"为"嵌入式资源":



你应当在你的一个web应用里,为所有你想做为嵌入式资源的文件的创建操作.

**添加到嵌入式资源管理器**

我们一旦把文件嵌入到程序集里,我们可以使用[启动配置](http://www.cnblogs.com/kid1412/p/5975791.html" \t "http://www.cnblogs.com/kid1412/p/_blank)把它们添加到嵌入式资源管理器,添加如下一行代码到你[模块](http://www.cnblogs.com/kid1412/p/5973274.html" \t "http://www.cnblogs.com/kid1412/p/_blank)的PreInitialize方法里:

Configuration.EmbeddedResources.Sources.Add(

new EmbeddedResourceSet(

**"/Views/",**

Assembly.GetExecutingAssembly(),

**"EmbeddedPlugIn.Views"**

)

);

让我们解释一下参数:

* 第一个定义了文件的根目录
* 第二个包含文件的程序集
* 最后一个定义了这些文件在程序集里的根命名空间,这是默认的命名空间(通常是程序集名称)+程序集里文件的位置,用一个.号连接.

**使用嵌入式视图**

对于.cshtml文件,直接从一个控制器的Action里返回.EmbeddedPlugIn程序集里的BlogController如下所示:

using Abp.AspNetCore.Mvc.Controllers;using Microsoft.AspNetCore.Mvc;

namespace EmbeddedPlugIn.Controllers

{

public class BlogController : AbpController

{

public ActionResult Index()

{

**return View();**

}

}

}

**使用嵌入式资源**

为使用嵌入式资源(js,css,img...),我们只要在我们的视图中如常地使用它们:

@section Styles {

<link href="~/Views/Blog/Index.css" rel="stylesheet" />

}

@section Scripts

{

<script src="~/Views/Blog/Index.js"></script>

}

<h2 id="BlogTitle">Blog plugin!</h2>

**忽略文件**

出于安全或其它意图,你可以忽略一些文件扩展名,.cshtml和.config默认情况下被忽略(防止从客户端直接访问).你可以在你模块的PreInitialize里添加更多扩展名:

Configuration.Modules.AbpWebCommon().EmbeddedResources.**IgnoredFileExtensions**.Add("exe");