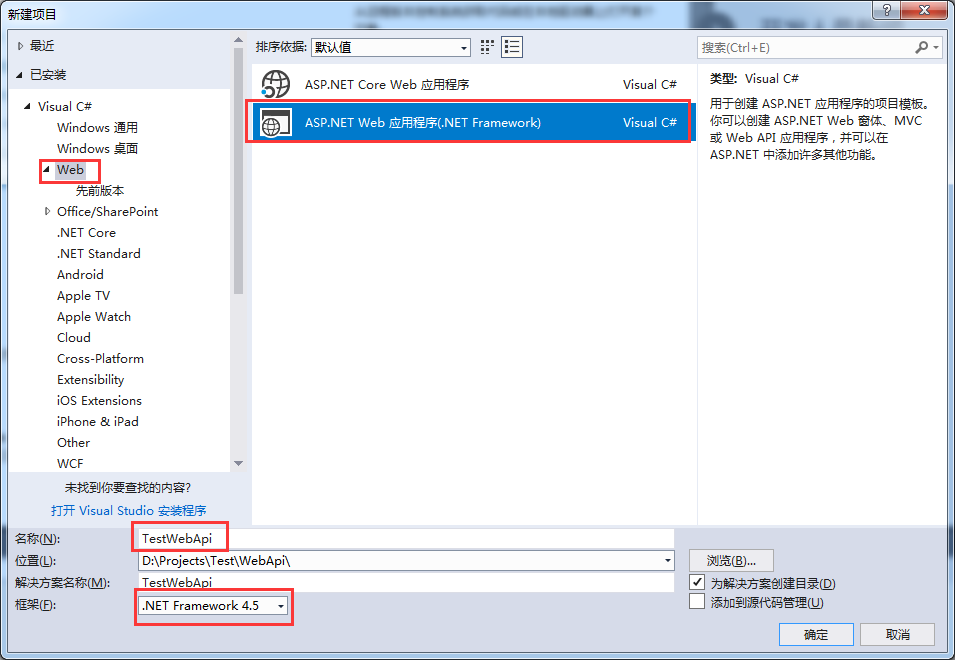
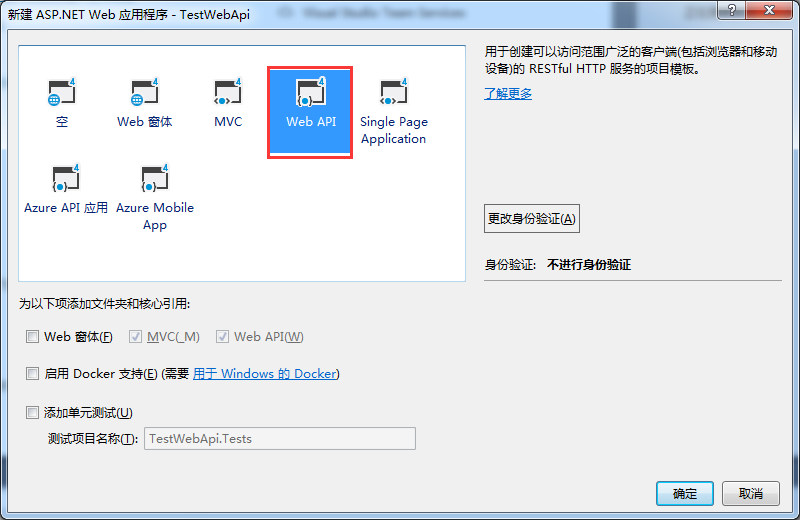
### Web API入门

#### 创建 Web API 项目

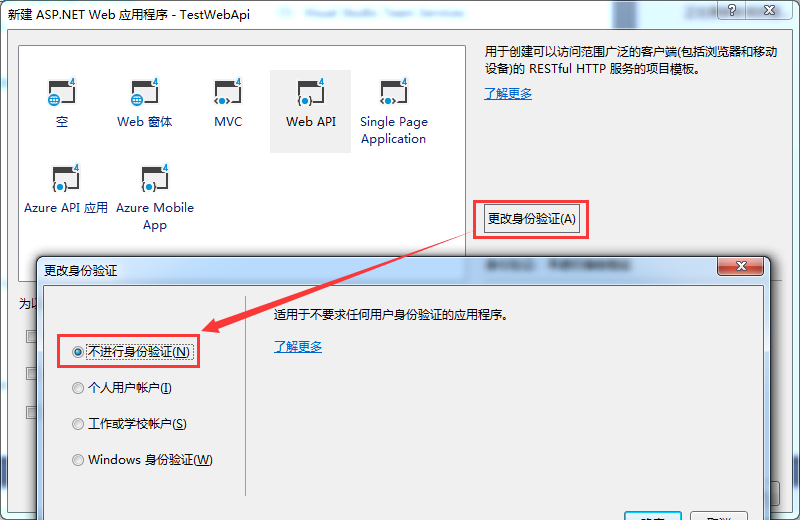
在这里我们使用VS2017, .NET 4.5创建一个Web API的项目



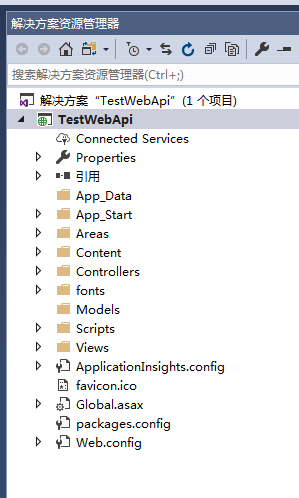
选择项目WEB API模板, 在最下方的MVC主要是默认会自带微软的API Helper, 使用MVC发布



身份认证暂时没有研究，有兴趣的可以研究一下，暂时选不进行验证

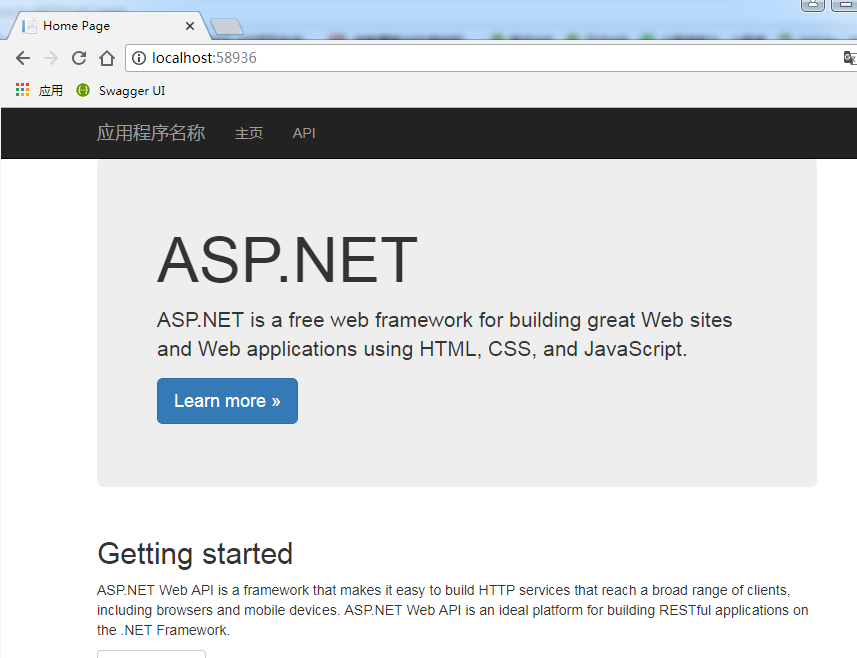


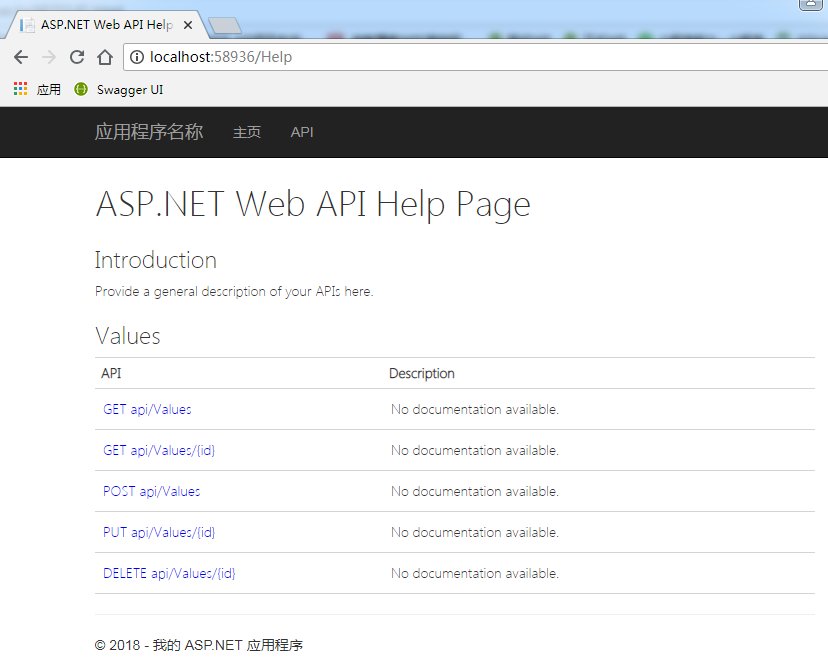
这样我们一个项目就初始化好了



F５启动项目之后

, 点击界面上API按钮可以看到默认创建的API

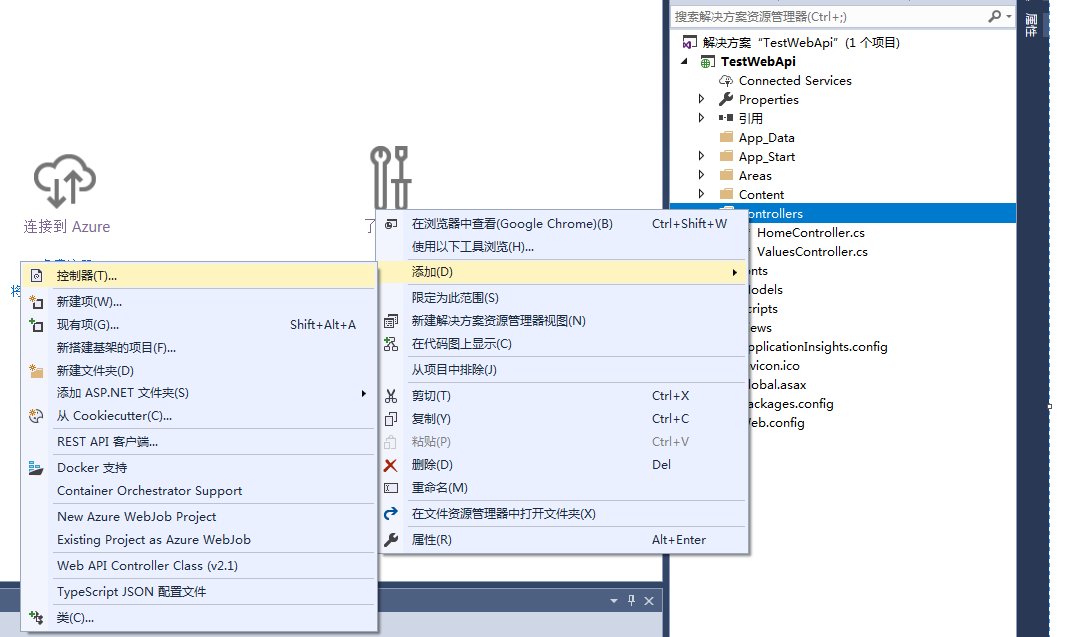


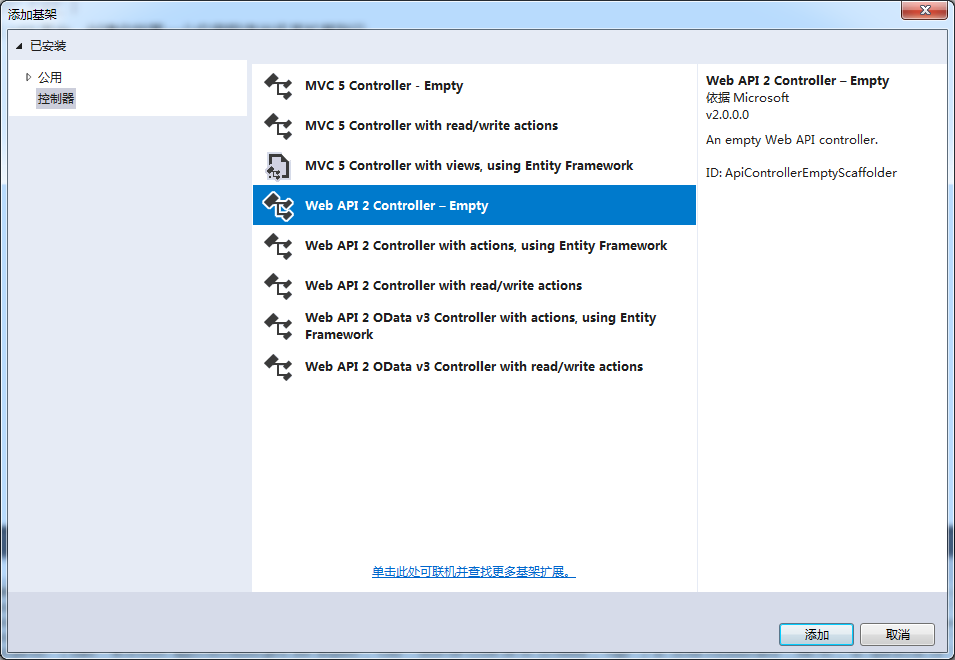


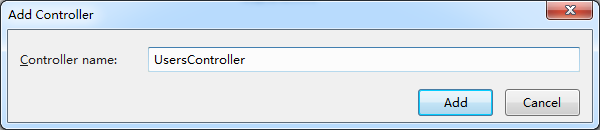
这时就可以用上面的地址测试一下。

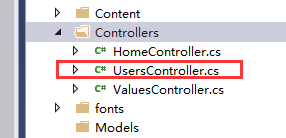
#### 创建我们自己的API-Users API

##### 添加UsersController









##### 如何制定URL格式

我们的 Users API简单的设计为下面格式:

登录API: api/users/user/login

在这里我们在刚才新添加的UsersController里中实现上面的API

/// <summary>

/// 用户操作信息

/// </summary>

[RoutePrefix("api/users")]

public class UsersController : ApiController

{

/// <summary>

/// 用户登录

/// </summary>

/// <param name="userCode">用户编码</param>

/// <param name="pwd">密码</param>

/// <returns>成功返回True，失败返回False</returns>

[HttpGet, Route("user/login")]

public bool Login(string userCode,string pwd)

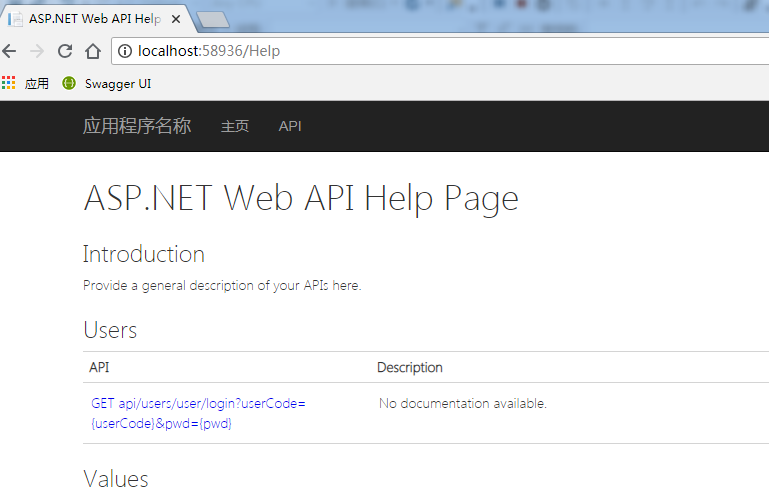
{

return (userCode == pwd);

}

}

添加之后启动程序，看到的结果如下



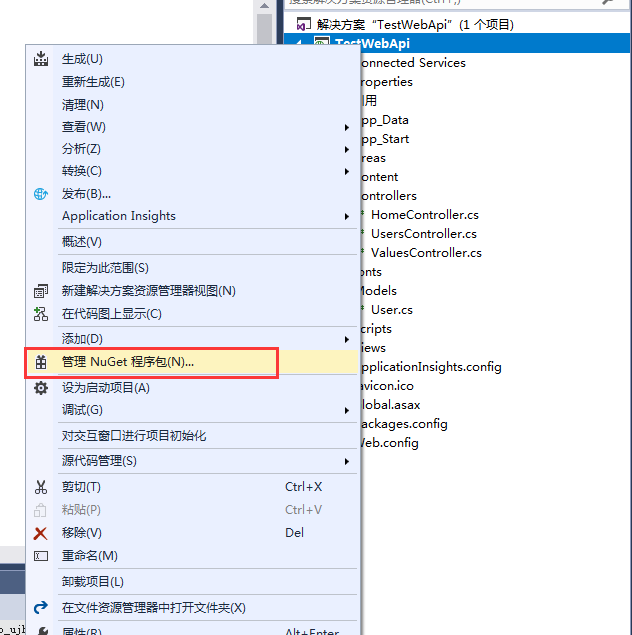
OK, 到这里大家加上自己的后端业务逻辑，完成业务层面的操作就可以发布使用了。

### [接口文档与在线测试](https://www.cnblogs.com/feinian/p/4870373.html)

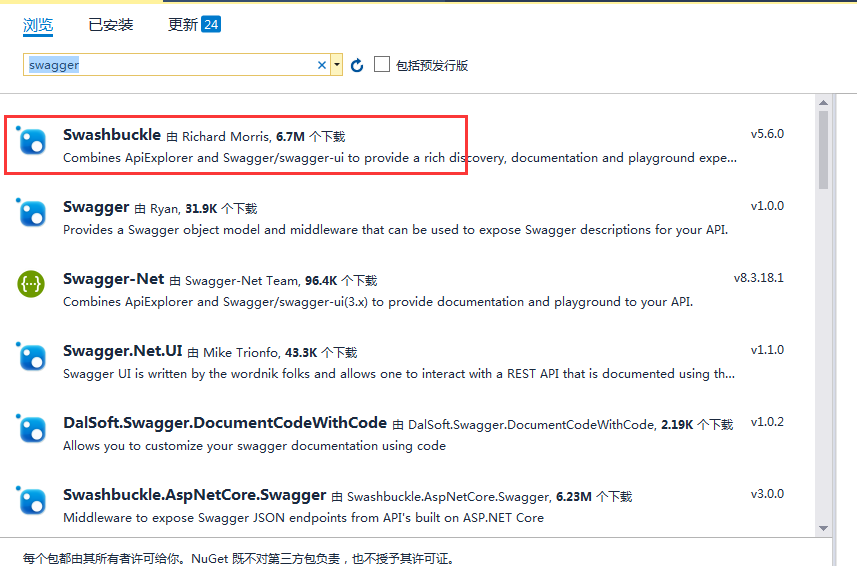
本节介绍如何做API帮助文档，给API的调用人员介绍各个 API的功能, 输入参数，输出参数, 以及在线测试 API功能(这个也是方便我们自己开发调试)

#### Swagger 引入

##### ：右键工程选择“管理NuGet程序包”

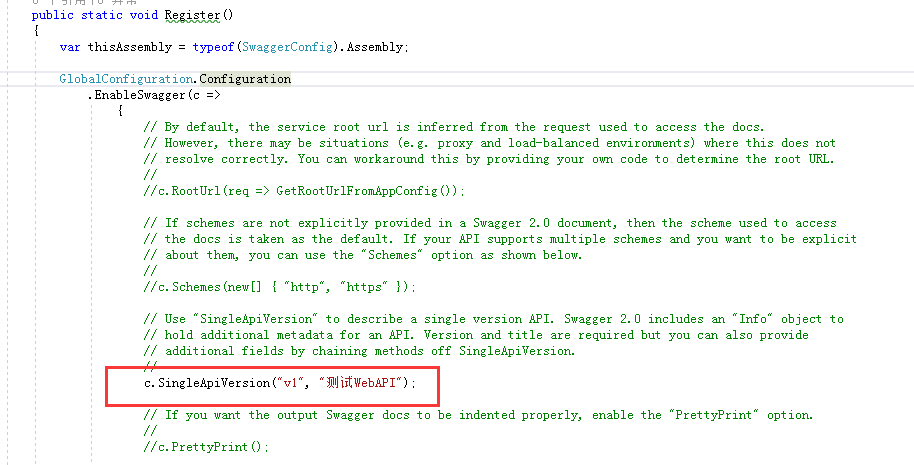


##### ：搜索swagger，选择Swashbuckle点击下载安装



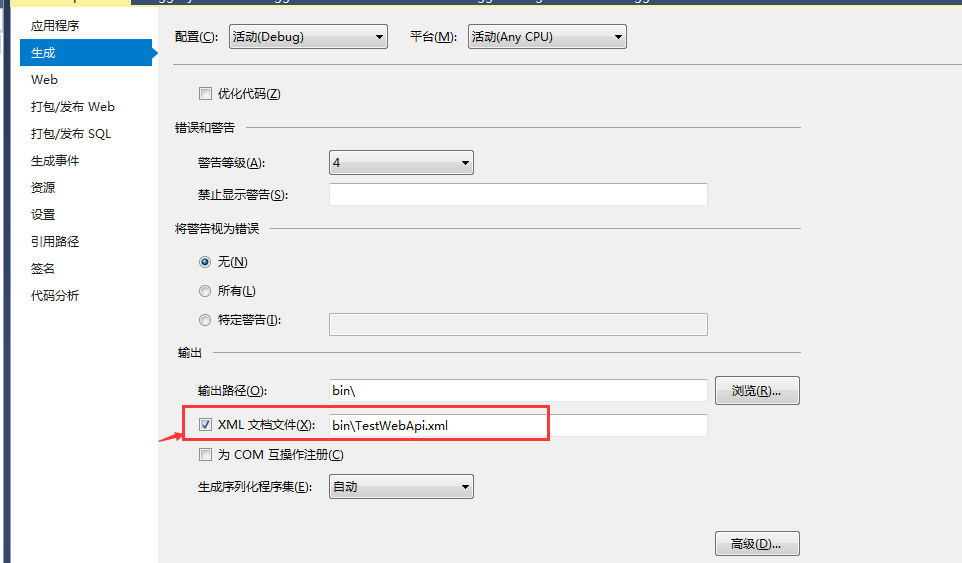
##### ：修改SwaggerConfig.cs

 如 api 版本号，title



##### ：创建项目XML注释文档

右键项目→属性→生成→选中下方的 "XML文档文件" 然后保存



配置启用：

c.IncludeXmlComments(GetXmlCommentsPath());

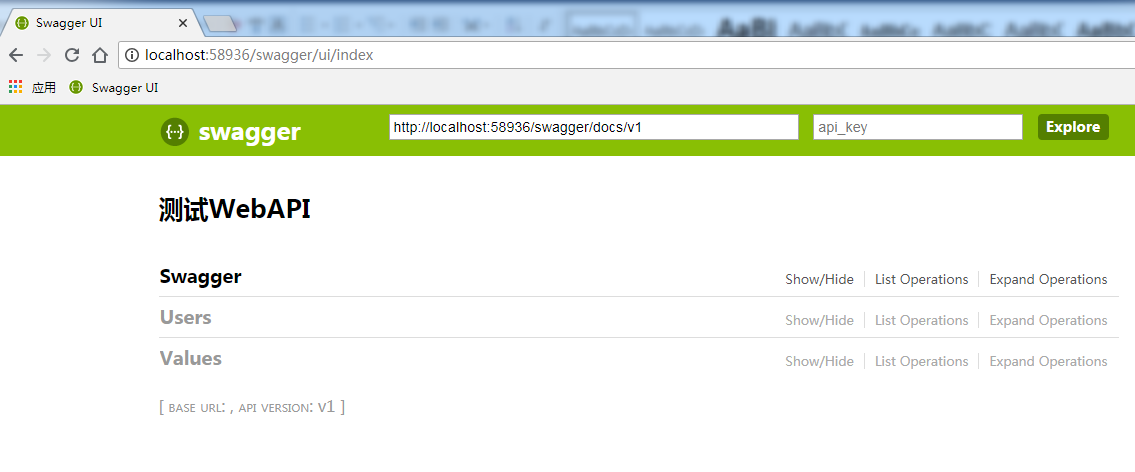
public static string GetXmlCommentsPath()

{

return string.Format("{0}/bin/TestWebApi.xml", System.AppDomain.CurrentDomain.BaseDirectory);

}

##### ：启动项目

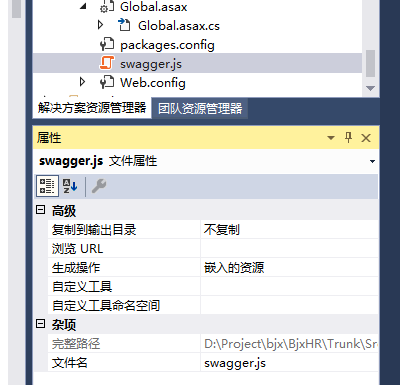
地址：http://localhost:XXXX/swagger 

#### 汉化及controller说明

##### ：定义一个provider实现ISwaggerProvider接口 包含了缓存 名：SwaggerCacheProvider

代码：

##### ：定义一个JS文件,做成嵌入资源，这个js文件的功能主要有两个，一个是汉化，另一个就是在界面上显示控制器的描述文字



代码：

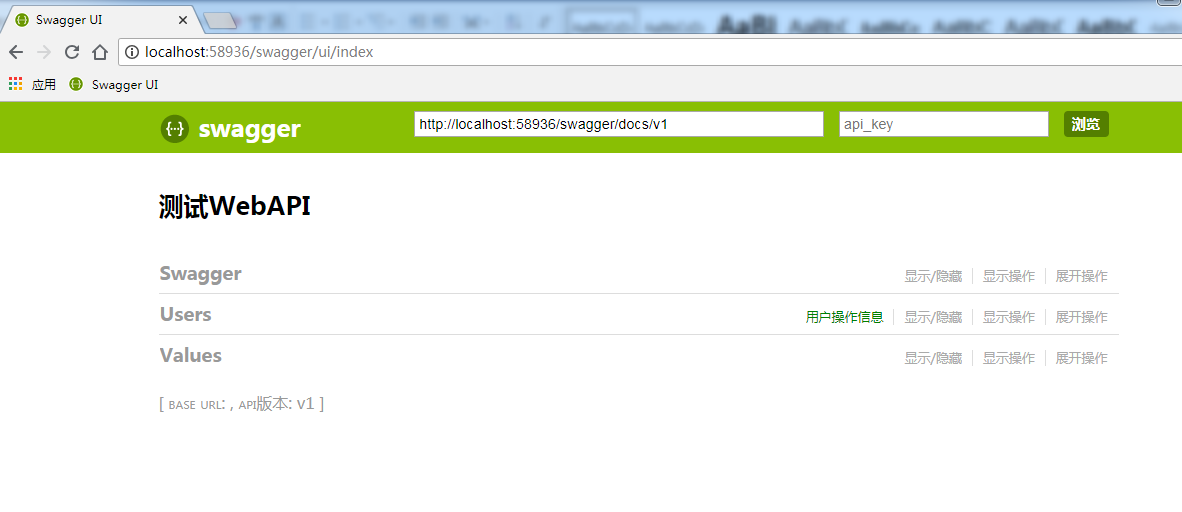
##### ：修改App\_Start中的SwaggerConfig.cs文件，主要代码有两行

c.CustomProvider((defaultProvider) => new SwaggerCacheProvider(defaultProvider, GetXmlCommentsPath()));

c.InjectJavaScript(thisAssembly, "TestWebApi.Scripts.swagger.js");

JS资源文件命名空间是：**文件所在项目的命名空间.文件径路.文件名**

 执行：



##### ：统一返回HttpResponseMessage 返回类型 指定

 很多时候我们会使用**HttpResponseMessage**作为返回对象 很方便，但是Swagger 不知道我们具体返回啥，它不看我们的业务代码！！

使用SwaggerResponse 指定返回类型

转json代码：

public static HttpResponseMessage ToJson(Object obj)

{

String str;

if (obj is String || obj is Char)

{

str = obj.ToString();

}

else

{

JavaScriptSerializer serializer = new JavaScriptSerializer();

str = serializer.Serialize(obj);

}

HttpResponseMessage result = new HttpResponseMessage { Content = new StringContent(str, Encoding.GetEncoding("UTF-8"), "application/json") };

return result;

}



效果展示



### 问题

#### 设置API版本号不能设置为小数

#### 发布时配置文件不会发布

### 总结

规范化api的编写和注释，以及标准化文档，对于团队的开发效率有很大的提升，也有利于项目的维护。使用在线接口文档后，方便前后端工程师沟通，测试人员测试只需要在页面输入参数，点击调用就可以看到调用结果。