**[C#进阶系列——WebApi 接口测试工具：WebApiTestClient](http://www.cnblogs.com/landeanfen/p/5210356.html)**

**阅读目录**

* [一、WebApiTestClient介绍](http://www.cnblogs.com/landeanfen/p/5210356.html#_label0)
* [二、WebApiTestClient展示](http://www.cnblogs.com/landeanfen/p/5210356.html#_label1)
* [三、WebApiTestClient使用](http://www.cnblogs.com/landeanfen/p/5210356.html#_label2)
  + [1、如何引入组件](http://www.cnblogs.com/landeanfen/p/5210356.html#_label2_0)
  + [2、如何使用组件](http://www.cnblogs.com/landeanfen/p/5210356.html#_label2_1)
* [四、总结](http://www.cnblogs.com/landeanfen/p/5210356.html#_label3)

**正文**

前言：这两天在整WebApi的服务，由于调用方是Android客户端，Android开发人员也不懂C#语法，API里面的接口也不能直接给他们看，没办法，只有整个详细一点的文档呗。由于接口个数有点多，每个接口都要详细说明接口作用、参数类型、返回值类型等等，写着写着把博主惹毛了，难道这种文档非要自己写不成？难道网上没有这种文档的展示工具吗？带着这两个问题，在网络世界里寻找，网络世界很奇妙，只要你用心，总能找到或多或少的帮助！这不就被博主找到了这个好用的组件：WebApiTestClient。它对于接口的作用、参数作用以及类型、返回值说明等等都能很好展现，并且提供测试调用的功能，输入测试参数发送请求，就能得到相应的返回结果。哇塞，这一系列可视化的效果不正是博主需要的吗，还写什么文档，赶紧用起来！

WebApi系列文章

* [C#进阶系列——WebApi接口测试工具：WebApiTestClient](http://www.cnblogs.com/landeanfen/p/5210356.html)
* [C#进阶系列——WebApi 跨域问题解决方案：CORS](http://www.cnblogs.com/landeanfen/p/5177176.html)
* [C#进阶系列——WebApi身份认证解决方案：Basic基础认证](http://www.cnblogs.com/landeanfen/p/5287064.html)
* [C#进阶系列——WebApi接口传参不再困惑：传参详解](http://www.cnblogs.com/landeanfen/p/5337072.html)
* [C#进阶系列——WebApi接口返回值不困惑：返回值类型详解](http://www.cnblogs.com/landeanfen/p/5501487.html)
* C#进阶系列——WebApi异常处理解决方案
* C#进阶系列——WebApi区域Area使用小结

[回到顶部](http://www.cnblogs.com/landeanfen/p/5210356.html#_labelTop)

# 一、WebApiTestClient介绍

1、WebApiTestClient组件作用主要有以下几个：

（1）、将WebApi的接口放到了浏览器里面，以可视化的方式展现出来，比如我们通过http://localhost:8080/Help这个地址就能在浏览器里面看到这个服务里面所有的API接口以及接口的详细说明，省去了我们手写接口文档的麻烦。

（2）、能够详细查看API的类说明、方法说明、参数说明、返回值说明。只需要我们在定义方法时候加上 /// <summary> 这种详细注释即可，组件自动读取注释里面的内容展现在界面上面。

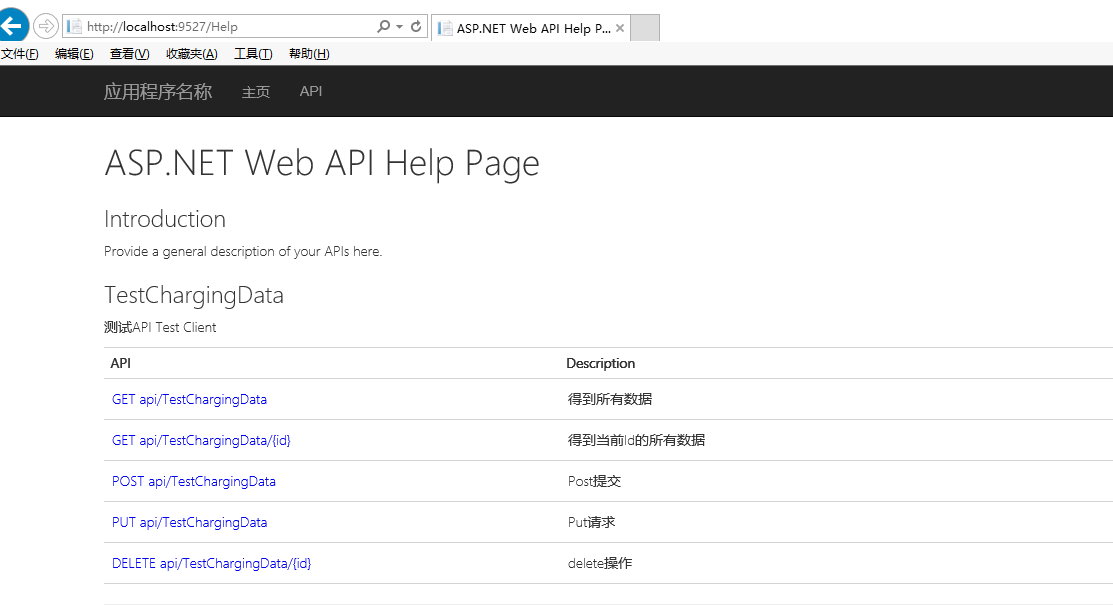
（3）、可以修改http请求头文件Head和请求体Body里面的参数，指定发送http请求的特性，比如指定我们最常见的contentType指示参数的类型。

（4）、组件拥有测试接口的功能，用过Soup UI的朋友应该知道，通过Soup UI能够方便测试WebService参数以及返回值。我们的WebApiTestClient也可以实现类似的功能，直接通过页面上的测试按钮，就能测试接口。

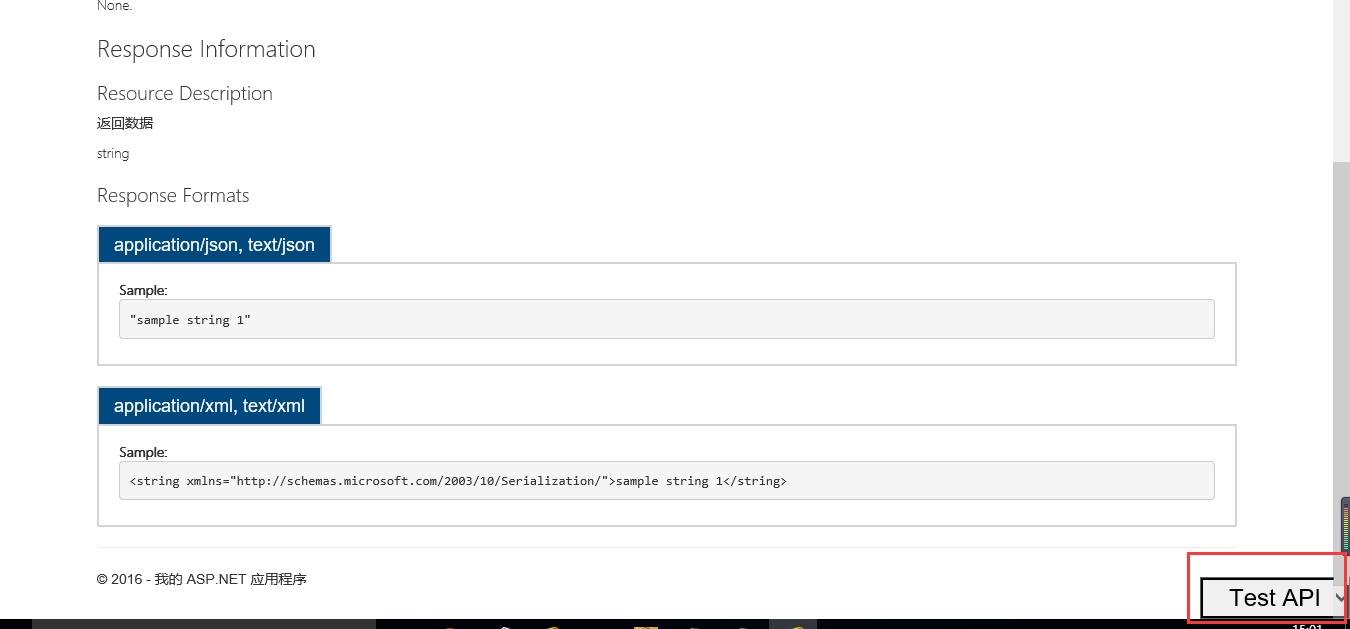
2、WebApiTestClient是一个开源组件。开源地址：[https://github.com/yaohuang/WebApiTestClient](https://github.com/yaohuang/WebApiTestClient" \t "_blank)

[回到顶部](http://www.cnblogs.com/landeanfen/p/5210356.html#_labelTop)

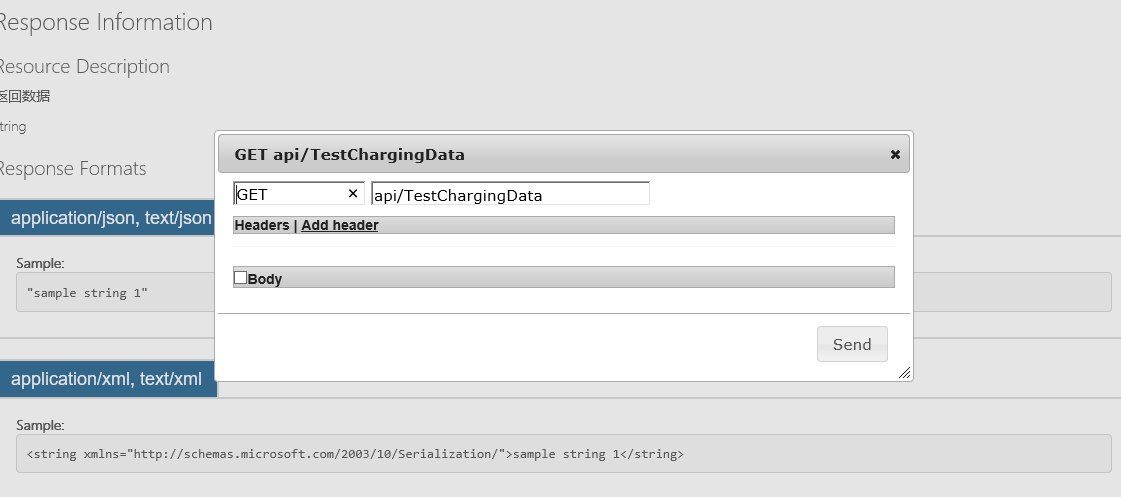
# 二、WebApiTestClient展示

第一印象：接口列表。 

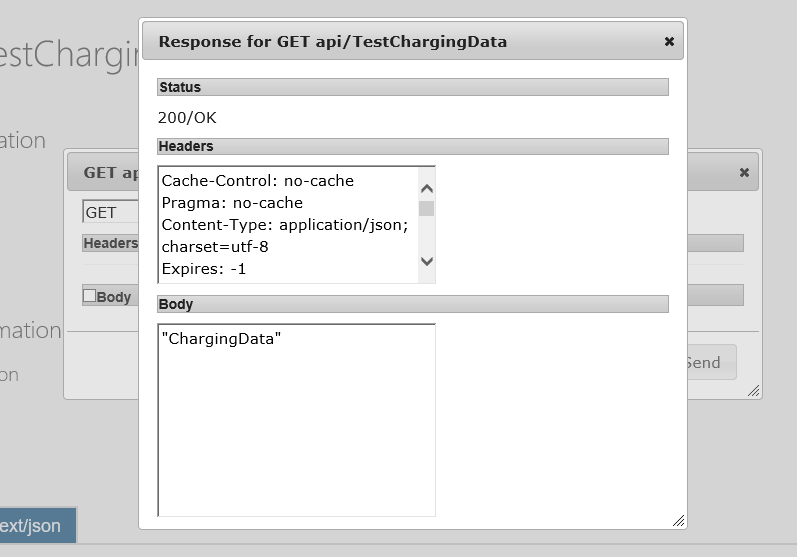
点击某一个接口查看接口详细。例如本文查看Get请求的无参方法，右下角有按钮可以测试接口。



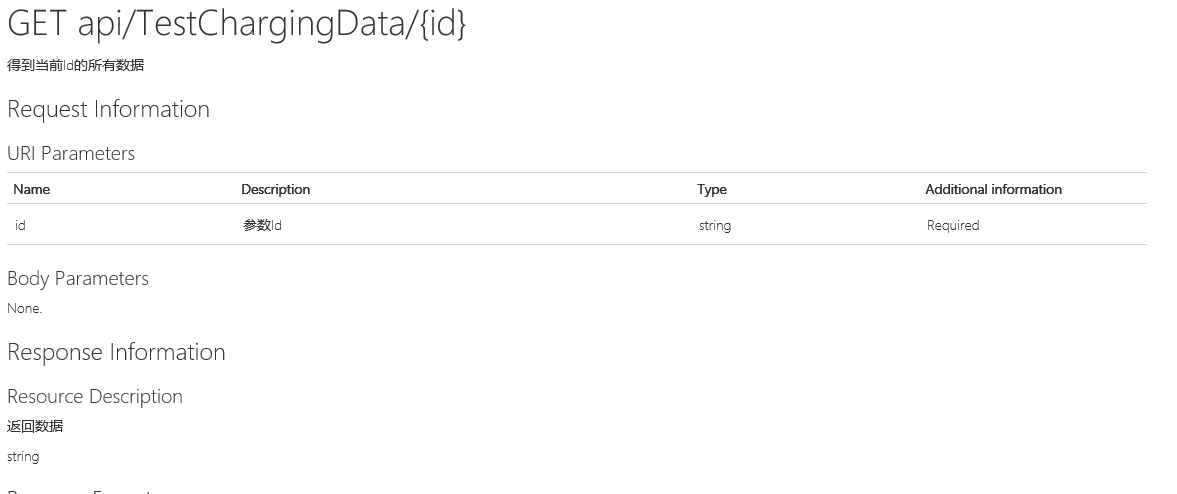
点击“Test API”按钮



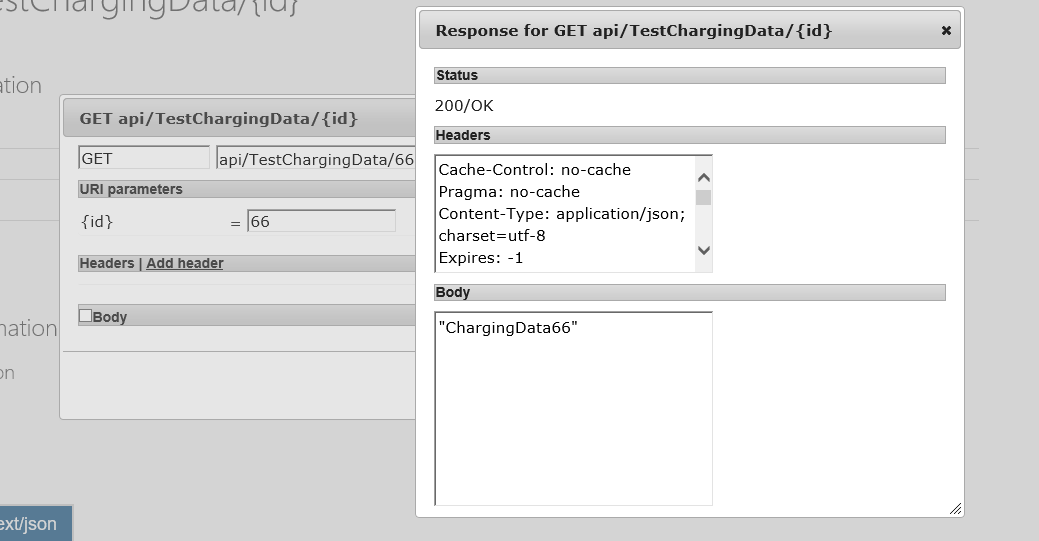
点击Send发送请求



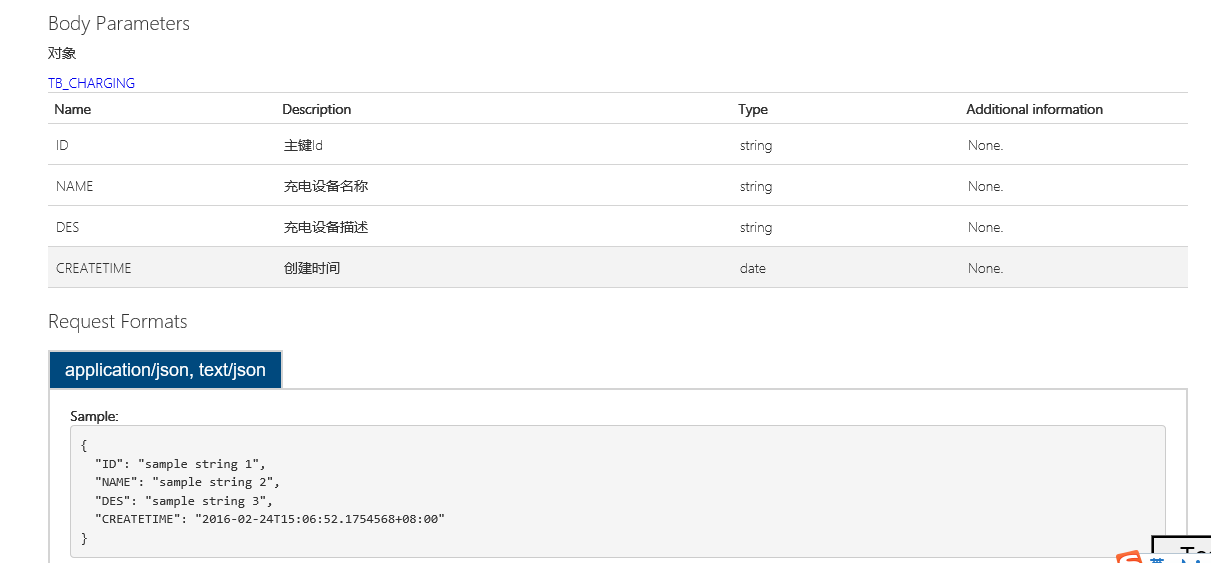
第二个有参数的接口



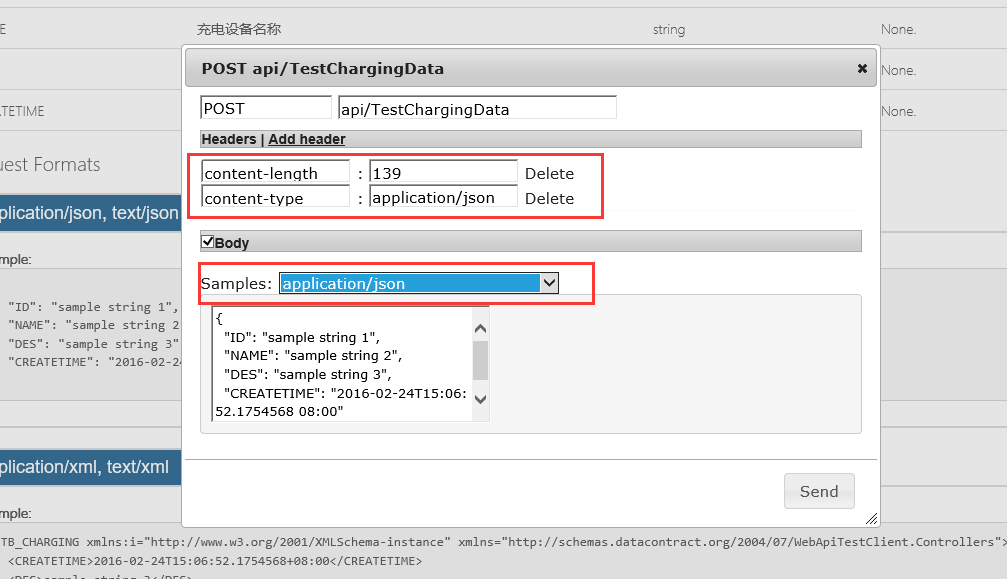
手动输入参数，得到返回结果

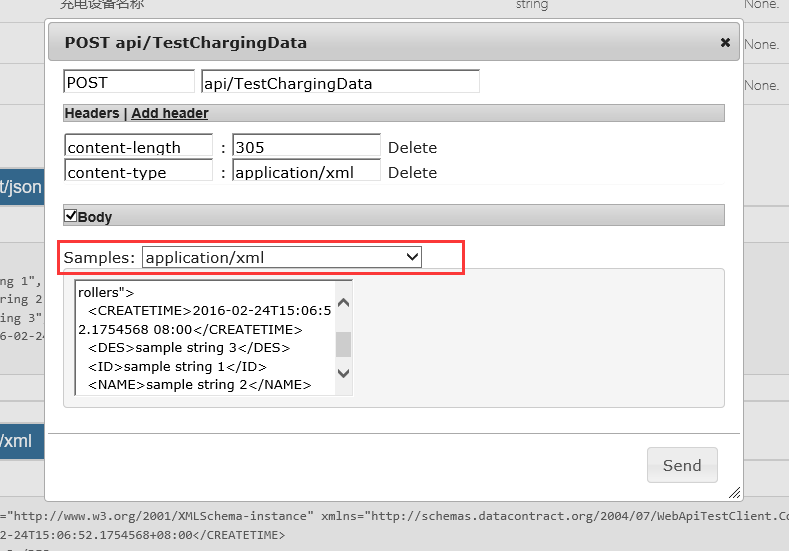


如果参数的类型是对象，可以直接解析class定义上面的 /// <summary> 标注，显示如下

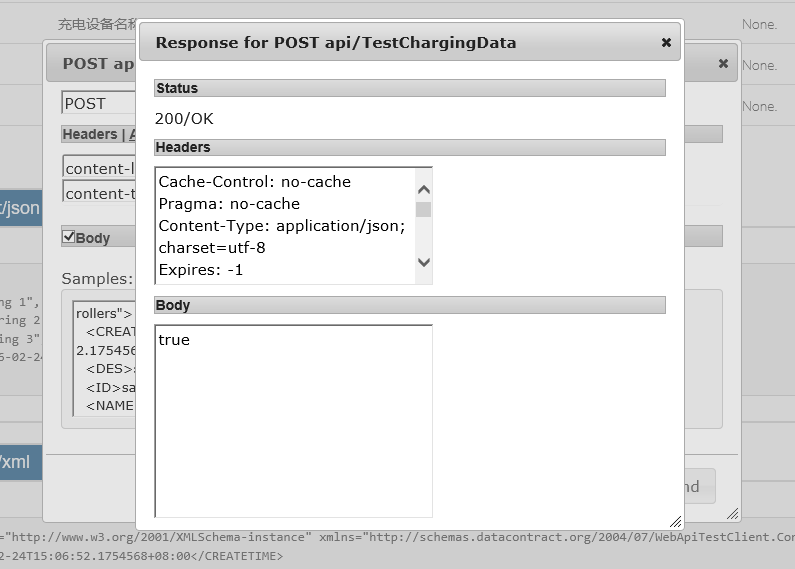


由于是post请求，如果需要执行参数长度和类型，可以通过content-length和content-type来指定。并且具体的参数可以指定不同格式显示，比如下图的application/json和application/xml





得到返回值



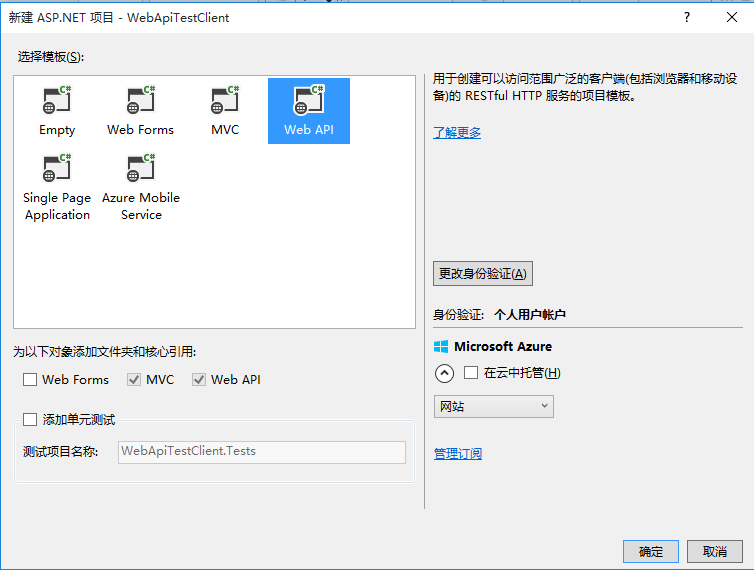
[回到顶部](http://www.cnblogs.com/landeanfen/p/5210356.html#_labelTop)

# 三、WebApiTestClient使用

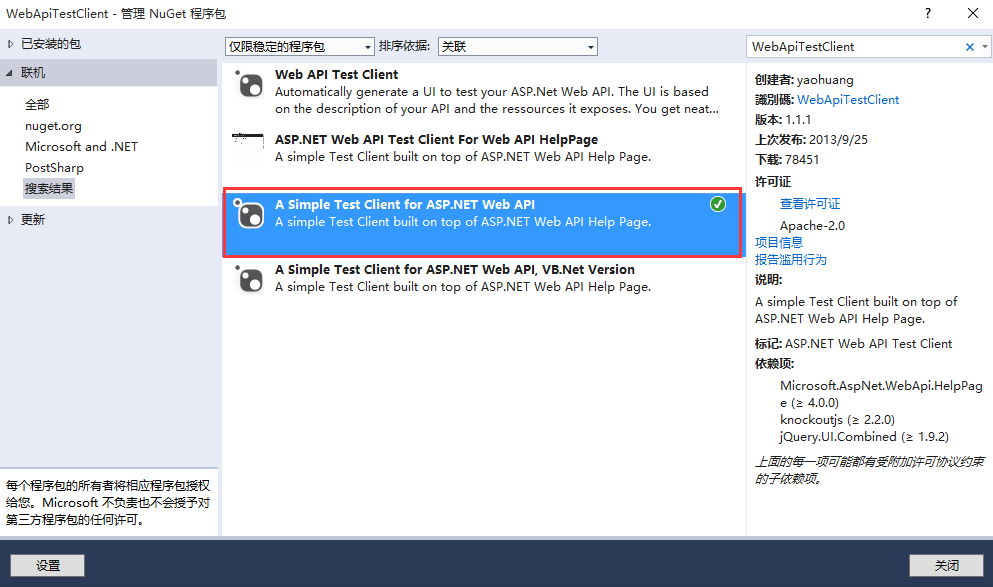
[回到顶部](http://www.cnblogs.com/landeanfen/p/5210356.html#_labelTop)

## 1、如何引入组件

首先，我们需要定义一个API项目



然后通过Nuget引入组件，如下图。记住选下图中的第三个。



引入成功后，将向项目里面添加一些主要文件：

* Scripts\WebApiTestClient.js
* Areas\HelpPage\TestClient.css
* Areas\HelpPage\Views\Help\DisplayTemplates\TestClientDialogs.cshtml
* Areas\HelpPage\Views\Help\DisplayTemplates\TestClientReferences.cshtml

[回到顶部](http://www.cnblogs.com/landeanfen/p/5210356.html#_labelTop)

## 2、如何使用组件

### 1、修改Api.cshtml文件

通过上述步骤，就能将组件WebAPITestClient引入进来。下面我们只需要做一件事：打开文件 (根据 Areas\HelpPage\Views\Help) Api.cshtml 并添加以下内容:

* @Html.DisplayForModel("TestClientDialogs")
* @Html.DisplayForModel("TestClientReferences")

添加后Api.cshtml文件的代码如下

[复制代码](javascript:void(0);)

@using System.Web.Http

@using WebApiTestClient.Areas.HelpPage.Models

@model HelpPageApiModel

@{

var description = Model.ApiDescription;

ViewBag.Title = description.HttpMethod.Method + " " + description.RelativePath;

}

<link type="text/css" href="~/Areas/HelpPage/HelpPage.css" rel="stylesheet" />

<div id="body" class="help-page">

<section class="featured">

<div class="content-wrapper">

<p>

@Html.ActionLink("Help Page Home", "Index")

</p>

</div>

</section>

<section class="content-wrapper main-content clear-fix">

@Html.DisplayForModel()

</section>

</div>

@Html.DisplayForModel("TestClientDialogs")

@section Scripts{

<link href="~/Areas/HelpPage/HelpPage.css" rel="stylesheet" />

@Html.DisplayForModel("TestClientReferences")

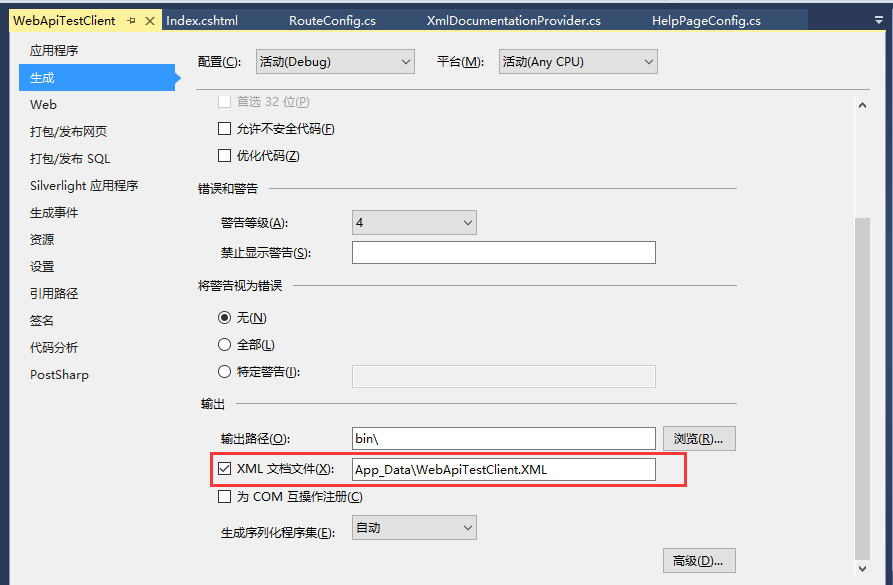
}

[复制代码](javascript:void(0);)

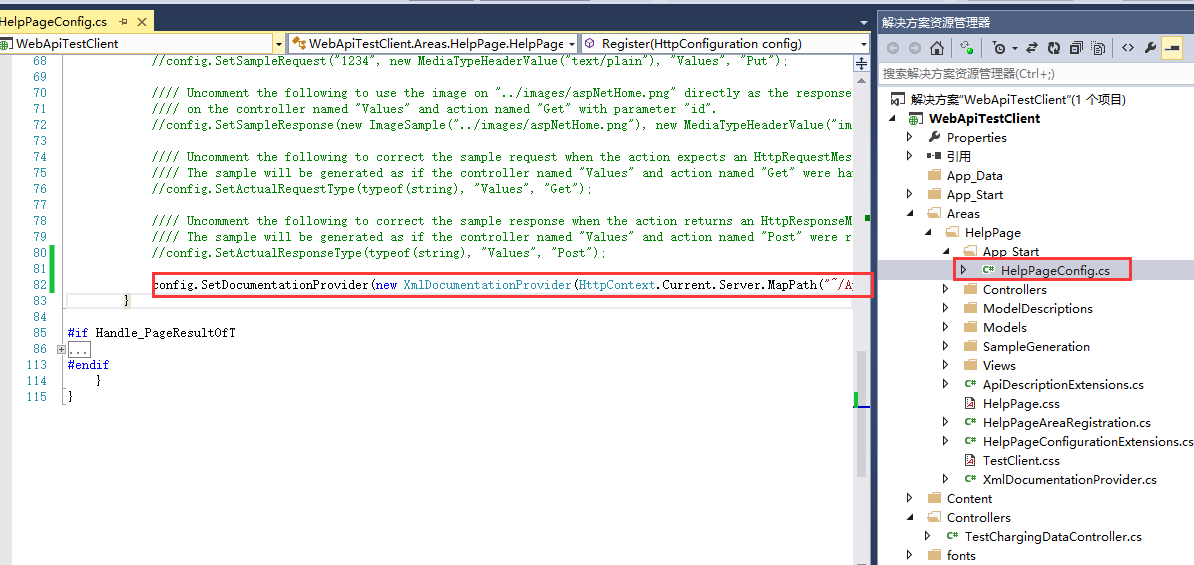
### 2、配置读取注释的xml路径

其实，通过上面的步骤，我们的项目已经可以跑起来了，也可以调用接口测试。但是，还不能读取 /// <summary> 注释里面的东西。需要做如下配置才行。

#### （1）配置生成xml的路径。我们在项目上面点右键→属性→生成标签页配置xml的路径



#### （2）在xml的读取路径：在下图的HelpPageConfig.cs里面配置一句话，指定xml的读取路径。



这句代码如下：

config.SetDocumentationProvider(new XmlDocumentationProvider(HttpContext.Current.Server.MapPath("~/App\_Data/WebApiTestClient.XML")));

### 3、测试接口

[复制代码](javascript:void(0);)

/// <summary>

/// 测试API Test Client

/// </summary>

public class TestChargingDataController : ApiController

{

/// <summary>

/// 得到所有数据

/// </summary>

/// <returns>返回数据</returns>

[HttpGet]

public string GetAllChargingData()

{

return "ChargingData";

}

/// <summary>

/// 得到当前Id的所有数据

/// </summary>

/// <param name="id">参数Id</param>

/// <returns>返回数据</returns>

[HttpGet]

public string GetAllChargingData(string id)

{

return "ChargingData" + id ;

}

/// <summary>

/// Post提交

/// </summary>

/// <param name="oData">对象</param>

/// <returns>提交是否成功</returns>

[HttpPost]

public bool Post(TB\_CHARGING oData)

{

return true;

}

/// <summary>

/// Put请求

/// </summary>

/// <param name="oData">对象</param>

/// <returns>提交是否成功</returns>

[HttpPut]

public bool Put(TB\_CHARGING oData)

{

return true;

}

/// <summary>

/// delete操作

/// </summary>

/// <param name="id">对象id</param>

/// <returns>操作是否成功</returns>

[HttpDelete]

public bool Delete(string id)

{

return true;

}

}

/// <summary>

/// 充电对象实体

/// </summary>

public class TB\_CHARGING

{

/// <summary>

/// 主键Id

/// </summary>

public string ID { get; set; }

/// <summary>

/// 充电设备名称

/// </summary>

public string NAME { get; set; }

/// <summary>

/// 充电设备描述

/// </summary>

public string DES { get; set; }

/// <summary>

/// 创建时间

/// </summary>

public DateTime CREATETIME { get; set; }

}

[复制代码](javascript:void(0);)

至此，组件就搭完了，剩下的就是运行了。我们在url里面敲地址http://localhost:8080/Help/Index或者http://localhost:8080/Help就能得到上述效果。还不赶紧试试~~

[回到顶部](http://www.cnblogs.com/landeanfen/p/5210356.html#_labelTop)

# 四、总结

上面简单总结了下WebApiTestClient的使用，如果你的WebApi也是一个单独的服务，可以整起来当帮助文档用，省得你自己写。哈哈，不要谢我，请叫我红领巾。欢迎推荐。[Demo下载](http://pan.baidu.com/s/1bnRl2XL)