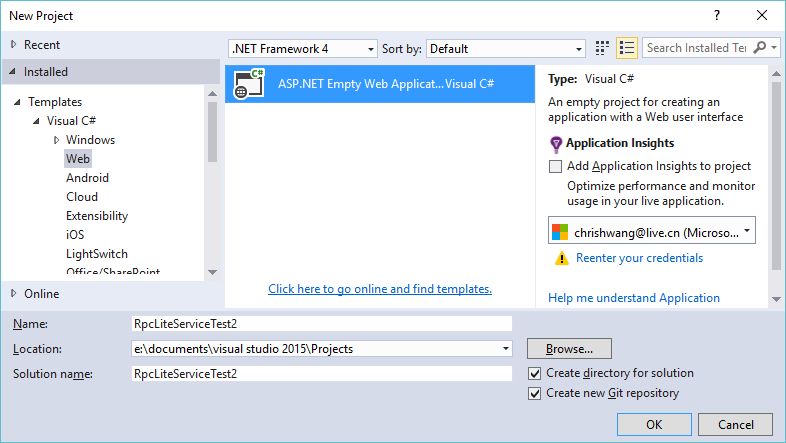
**使用RpcLite构建SOA/Web服务**

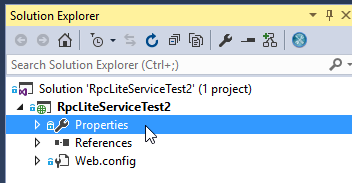
提到Web服务最先想到的就是WebService此外常用的还有WCF、ServiceStack、WebApi等。

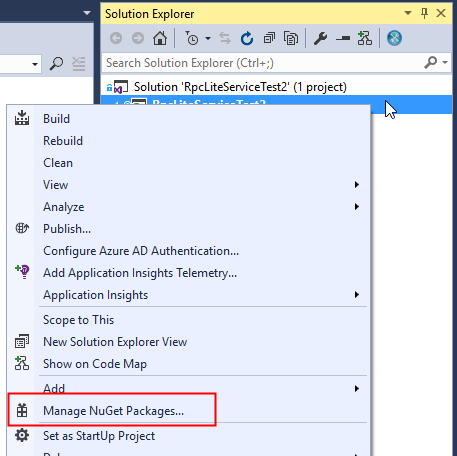
RpcLite是一个开源的轻量级SOA服务框架，除了基本的提供Web服务还包括SOA治理系统（本文暂不展开在后继的文章会详细说明）。RpcLite与WCF比较像，下面介绍使用方法。

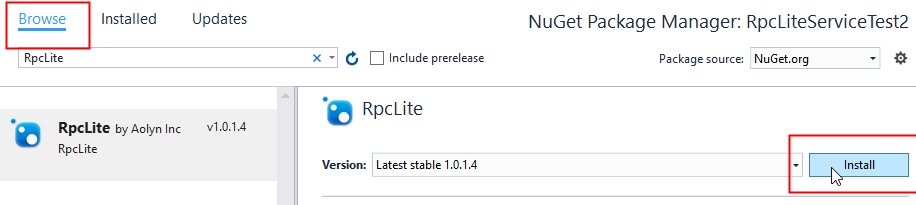
## **创建一个Web项目，Framework选择>=4.0**



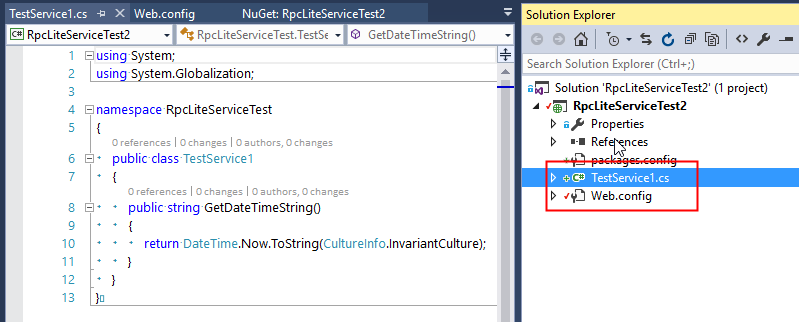
## **从NuGet中添加RpcLite**



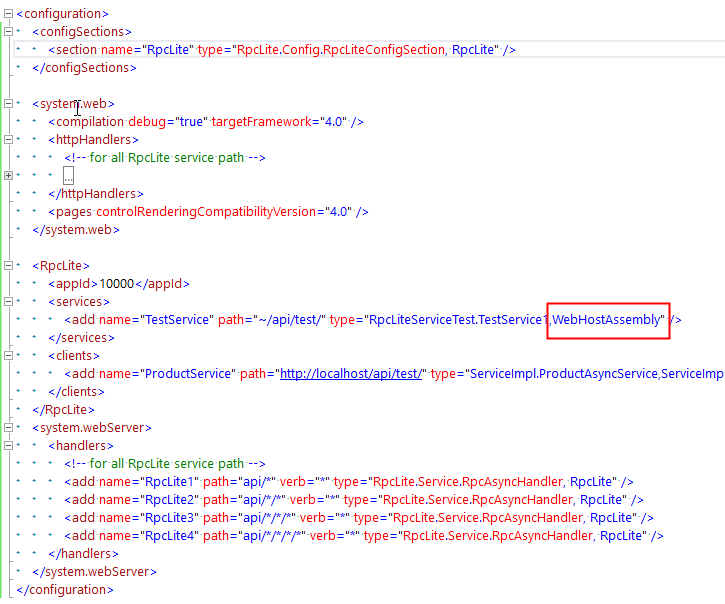




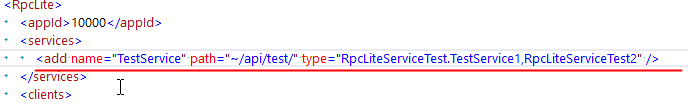
安装后项目中会添加一个TestService1.cs文件，此文件是一个最简单的服务提供一个接口GetDateTimeString返回当前时间的字符串



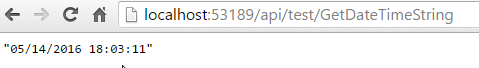
Web.config也添加了相关的配置



## **修改配置**



到此为止，最简单有服务就已经完成了。F5运行在浏览器中把地址改为<http://localhost:53189/api/test/GetDateTimeString>可看到接口返回的结果，用JQuery, Ajax都可以访问了。

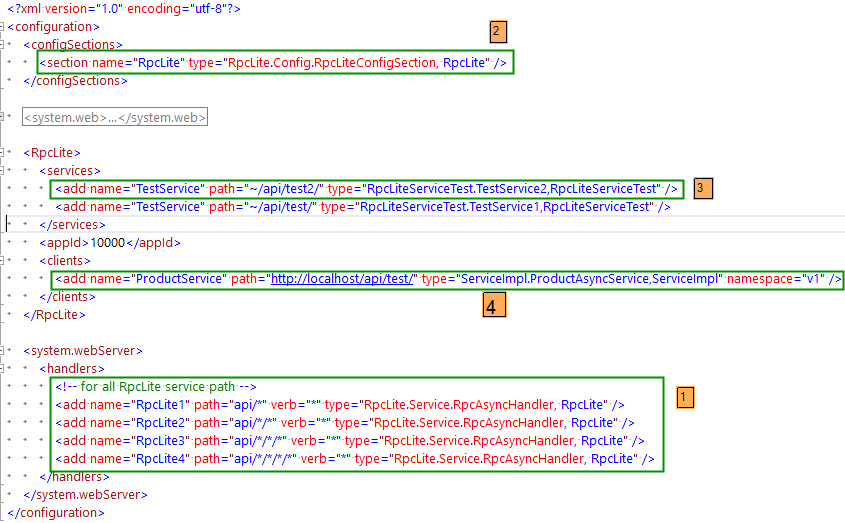


## **基本原理**

架构图

RpcLite的入口是一个异步HttpHandler，在Handler中会把RequestUrl、ContentType、RequestStream、ResponseStream、HttpContext等放到一个新的ServiceContext中交给RpcService处理，RpcService会根据RequestUrl找到对应的Service及Action根据ContentType选择序列化方式（默认支持json、xml，可以添加自定义序列化方式），然后反序列化得到参数，再使用参数调用具体的服务方法，得到结果后返回给客户端。

## **配置文件说明**



再来看Web.config，

标记1. 添加HttpHandler添加后对应地址的请求会被RpcAsyncHandler处理。

标记2. 添加RpcLite配置节点处理器，添加后RpcLite就能读取RpcLite节点中的数据。

标记3. 此处添加了一个服务，name只起识别作用最好不要重复，path确定此服务会处理来自哪些url地址的请求需要使用虚拟目录相对地址即以~/开头，type设置实现此服务的类格式是<类全名>, 程序集名。

标记4. 这里是添加一个客户端配置，如果创建客户端实例时未指定服务url则会使用这里设置的path。如ClientFactory.GetInstance<ITestService>()此处未指定服务url就会从配置中查找。

其它. appId、environment、namespace与治理系统相关此处不展开明说。

代码从下面给出的git中获取。

## **相关链接**

**Demo git 地址：<https://github.com/chrishaly/RpcLiteDemo>**

**RpcLite git 地址：**[**https://github.com/aolyn/RpcLite**](https://github.com/aolyn/RpcLite)