**[WCF4.0 -- RESTful WCF Services(创建基于SSL的服务)](http://blog.csdn.net/fangxing80/article/details/6299168)**

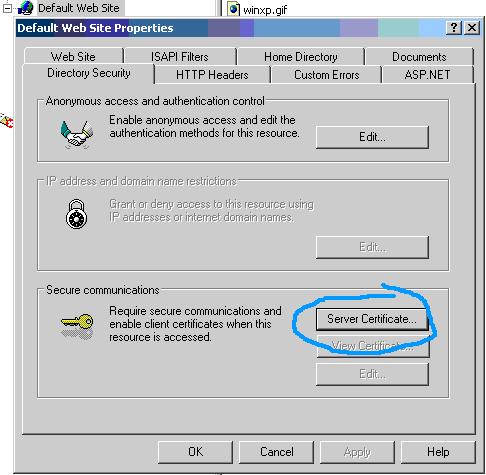
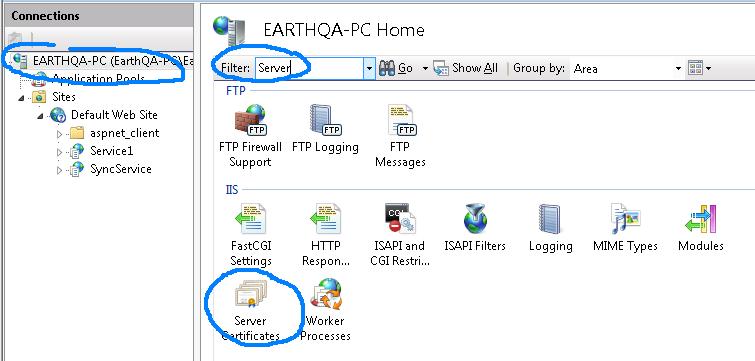
分类： [[05] WCF](http://blog.csdn.net/fangxinggood/article/category/767778) 2011-04-02 19:57 4555人阅读 [评论](http://blog.csdn.net/fangxing80/article/details/6299168#comments)(12) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/fangxing80/article/details/6299168#report)

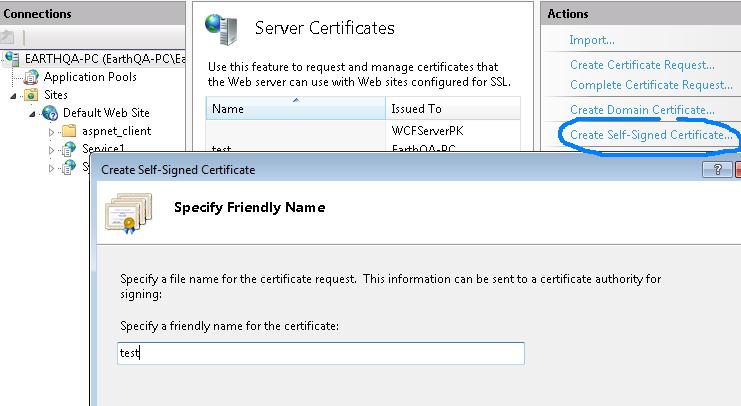
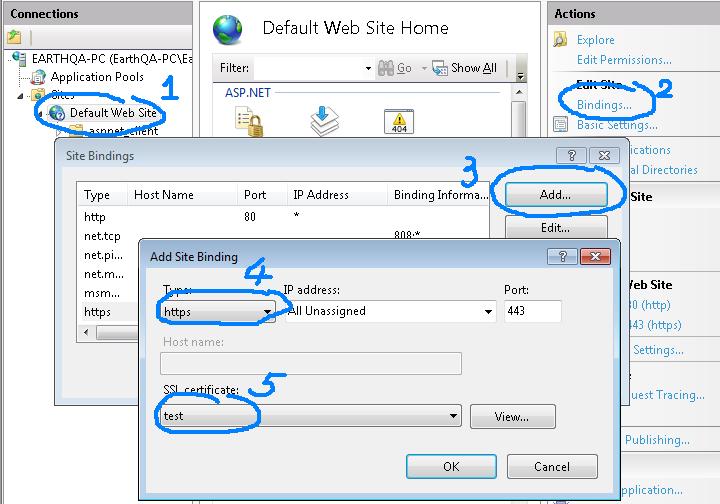
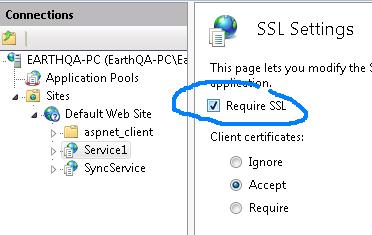
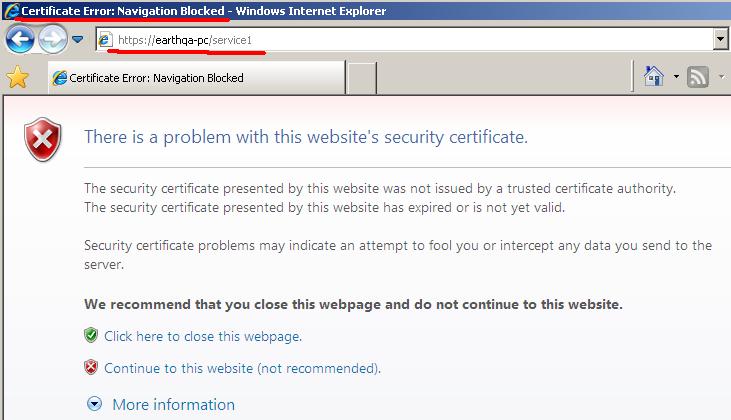
[wcf](http://www.csdn.net/tag/wcf)[ssl](http://www.csdn.net/tag/ssl)[application](http://www.csdn.net/tag/application)[iis](http://www.csdn.net/tag/iis)[rest](http://www.csdn.net/tag/rest)[string](http://www.csdn.net/tag/string)

其实这个标题和WCF本身关系不大，因为REST WCF服务一般寄宿于IIS，而SSL是传输层的事儿，说白了就是IIS的事。  
首先创建个简单的REST WCF服务：(使用 WCF REST Service Application 模板)

**[c-sharp]** [view plaincopy](http://blog.csdn.net/fangxing80/article/details/6299168)

1. [ServiceContract]
2. [AspNetCompatibilityRequirements(RequirementsMode = AspNetCompatibilityRequirementsMode.Allowed)]
3. [ServiceBehavior(InstanceContextMode = InstanceContextMode.PerCall)]
4. public class Service1
5. {
6. [WebGet(UriTemplate="GetHello")]
7. public string GetHello()
8. {
9. return "Hello Client by Get";
10. }
12. [WebInvoke(UriTemplate = "PostHello", Method="POST")]
13. public string PostHello()
14. {
15. return "Hello Client by Post";
16. }
17. }

服务很简单，就是为了让客户端通过HttpRequst在SSL下能访问到就行。  
  
**IIS6的配置: 在网站根目录下选"Directory Security">"Server Certificate...">Next>Assign an existing certificate  
然后选一个自己已经准备好的证书就可以了。如何生成证书就不说了，网上自己搜搜吧。  
  
IIS7的配置:   
1.在IIS7里可以直接创建新证书：  
  
  
为了测试，生成自签名的证书。**

  
  
**接下来需要为IIS设置Https Binding和证书  
  
  
设置完毕后，在Default Web Site下的所有Application都可以通过两种方式来访问：Http或Https  
如果需要设置只能通过Https访问，还需要到指定的Application上设置下SSL Settings，让该Application要求SSL:  
  
现在直接通过IE来访问这个WCF服务，可以看到如下的效果：  
  
因为生成的服务端证书是自签名的(没有第3方认证)所以被浏览器拦截了。问你是否要继续访问，点继续访问就可以看到结果了。  
  
  
OK，再来看看客户端如何调用：**

**[c-sharp]** [**view plaincopy**](http://blog.csdn.net/fangxing80/article/details/6299168)

1. **static void Main(string[] args)**
2. **{**
3. **var url = "https://earthqa-pc/Service1/gethello";**
4. **try**
5. **{**
6. **ServicePointManager.ServerCertificateValidationCallback +=**
7. **new RemoteCertificateValidationCallback(allowCert);**
8. **WebClient wc = new WebClient();**
9. **var str = wc.DownloadString(url);**
10. **Console.WriteLine(str);**
11. **}**
12. **catch (Exception ex)**
13. **{**
14. **Console.WriteLine("Error: {0}", ex.Message);**
15. **}**
16. **Console.ReadLine();**
17. **}**
19. **private static bool allowCert(object sender, X509Certificate cert, X509Chain chain, SslPolicyErrors error)**
20. **{**
21. **var request = sender as HttpWebRequest;**
22. **Console.WriteLine(request.Address.Host);**
23. **//Console.WriteLine(cert.Subject);**
24. **return true;**
25. **}**

**因为没有客户端证书，所以客户端只需要承认下服务端证书即可：在 ServicePointManager.ServerCertificateValidationCallback 事件中，返回true即可。**