

しovの沉没

博客园 首页 社区 新随笔 联系 管理 订阅 XML

随笔- 4 文章- 1 评论- 30

Web Service 身份验证

解决方案一：通过通过SOAP Header身份验证。

1.我们实现一个用于身份验证的类，文件名MySoapHeader.cs

MySoapHeader类继承自System.Web.Services.Protocols.SoapHeader。且定义了两个成员变量，UserName和PassWord，还定义了一个用户认证的函数ValideUser。它提供了对UserName和PassWord检查的功能



```
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
```

昵称：しovの沉没

园龄：6年7个月

粉丝：7

关注：0

+加关注

<	2009年8月						>
日	一	二	三	四	五	六	
26	27	28	29	30	31	1	
2	3	4	5	6	7	8	
9	10	11	12	13	14	15	
16	17	18	19	20	21	22	
23	24	25	26	27	28	29	
30	31	1	2	3	4	5	

搜索

<input type="text"/>	找找看
<input type="text"/>	谷歌搜索

常用链接

```
using System.Web.Services;
using System.Web.Services.Protocols;
/// <summary>
///MySoapHeader 的摘要说明
/// </summary>
public class MySoapHeader:SoapHeader
{
    public MySoapHeader()
    {
        //
        //TODO: 在此处添加构造函数逻辑
        //
    }
    public string UserName;
    public string PassWord;
    public bool ValideUser(string in_UserName, string in_PassWord)
    {
        if ((in_UserName == "zxq") && (in_PassWord == "123456"))
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}
```



2.下面我们创建WebService.asmx WebService.cs代码如下:

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)
[更多链接](#)

我的标签

[.net\(1\)](#)
[Web Service\(1\)](#)
[Windows服务\(1\)](#)
[安装程序\(1\)](#)
[身份验证\(1\)](#)

随笔分类

[.NET\(2\)](#)
[JS](#)
[SQL\(1\)](#)
[随笔](#)
[网络安全\(1\)](#)

随笔档案

[2010年1月 \(1\)](#)
[2009年8月 \(1\)](#)
[2009年2月 \(1\)](#)
[2009年1月 \(1\)](#)

文章分类

[生活\(1\)](#)
[所想](#)

阅读排行榜



```
using System;
using System.Collections;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;

/// <summary>
///WebService 的摘要说明
/// </summary>
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class WebService : System.Web.Services.WebService
{

    public WebService()
    {

        //如果使用设计的组件，请取消注释以下行
        //InitializeComponent();

    }

    public MySoapHeader header; ///定义用户身份验证类变量header
    [WebMethod(Description = "用户验证测试")]
    [System.Web.Services.Protocols.SoapHeader("header")]//用户身份验证的soap头
    public string HelloWorld(string contents)
    {

        //验证是否有权访问
        if (header.ValidateUser(header.UserName, header.Password))
        {

            return contents + "执行了";

        }
        else
        {

            return "您没有权限访问";

        }

    }

}
```

1. .net Windows服务程序和安装程序制作图解(22279)
2. Web Service 身份验证(3760)
3. SQL2005数据库转换成2000的数据库(856)
4. 网站安全之常见入侵总结(336)

评论排行榜

1. .net Windows服务程序和安装程序制作图解(23)
2. Web Service 身份验证(7)

推荐排行榜

1. .net Windows服务程序和安装程序制作图解(13)
2. Web Service 身份验证(3)



3.客户端 创建个Default.aspx

Default.aspx .cs代码



```
using System;
using System.Configuration;
using System.Data;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        com.cnlyw.WebService test = new com.cnlyw.WebService();//web引用(改成您自己的)
        com.cnlyw.MySoapHeader Header = new com.cnlyw.MySoapHeader();//web引用创建soap头对象(改成您自己的)
        //设置soap头变量
        Header.UserName = "zxq";
        Header.Password = "123456";
        test.MySoapHeaderValue = Header;
        //调用web 方法
        Response.Write(test.HelloWorld("我是强"));
    }
}
```

```
}
```



解决方案二：通过集成windows身份验证。

1. 将web服务程序设为集成windows身份验证



2. 客户端web引用代码

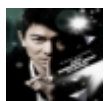
```
Test.WebReference.Service1 wr = new Test.WebReference.Service1(); //生成web service实例  
wr.Credentials = new NetworkCredential("guest","123"); //guest是用户名，该用户需要有一定的权限  
lblTest.Text = wr.Add(2,2).ToString(); //调用web service方法
```

该方案的优点是比较安全，性能较好，缺点是不便于移植，部署工作量大。

分类: [.NET](#)

标签: [Web Service](#), [身份验证](#)

绿色通道: [好文要顶](#) [关注我](#) [收藏该文](#) [与我联系](#) 



[Lovの沉没](#)

[关注 - 0](#)

[粉丝 - 7](#)

[+加关注](#)

3

0

(请您对文章做出评价)

« 上一篇: [网站安全之常见入侵总结](#)

» 下一篇: [.net Windows服务程序和安装程序制作图解](#)

posted @ 2009-08-04 18:09 [Lovの沉没](#) 阅读(3760) 评论(7) [编辑](#) [收藏](#)

发表评论

#1楼 2009-08-04 18:25 | 黎波

[回复](#) [引用](#)

第一种方式是明文的，考虑用WSE吧

[支持\(0\)](#) [反对\(0\)](#)

#2楼 2009-08-04 21:04 | Frank Xu Lei

[回复](#) [引用](#)

不错，支持一下，学习态度可嘉~

当初我为了学习WCF，自己也整理过一个Web Service的文章，当时也是实现了SoapHeader携带用户名和密码，实现身份验证。

WSE3.0安全机制更加完善，我也整理的系列文章，你有兴趣的话可以看看：[WCF分布式开发必备知识\(4\):Web Service](#)

[WSE3.0构建Web服务安全\(4\)](#)

我最近在这个理WCF安全开发系列文章，你要是对WCF开发有兴趣，我们可以一起交流~

支持(0) 反对(0)

#3楼 2009-08-04 22:05 | 北漂的大尾巴老猫

回复 引用

请教一个问题，现在有WCF了，还需要WSE3.0吗？WSE3.0和VS2008不兼容。

支持(0) 反对(0)

#4楼 2009-08-04 23:04 | Kinglee

回复 引用

挺实用的。

支持(0) 反对(0)

#5楼 2009-08-05 09:04 | wuxiaoqqqq

回复 引用

现在WSE的解决方案好像越来越成熟了，虽然比这两种更难部署

支持(0) 反对(0)

#6楼 2009-08-05 09:26 | Aydn1979[未注册用户]

回复 引用

第一种怎么成明文的了？

#7楼 2009-08-05 14:07 | Vincent.Feng

回复 引用


这个文章不是原创吧.我记得以前我看过.

支持(0) 反对(0)


发表评论


昵称：


评论内容：




B









提交评论

注销

订阅评论

[Ctrl+Enter快捷键提交]

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【推荐】融云即时通讯云 - 专注为 App 开发者提供IM云服务



最新IT新闻:

- 可能是最漂亮的：锤子便签2.5下载
 - 899的情怀：锤子坚果首发评测！
 - 坚果手机极速拆解：神奇的耳机孔 服了！
 - 一图看懂坚果U1 其实很“俗”
 - 全网通的乐视手机来啦
- » 更多新闻...

【连载】0基础7天入门Python(第1天)

Google、Instagram、豆瓣、知乎，崇尚优雅的互联网公司都在用Python！



最新知识库文章:

- 关于软件开发，你老板不知道的7件事
 - 关于烂代码的那些事（中）
 - 关于烂代码的那些事（上）
 - 作为码农，我们为什么要写作
 - 今天你写了自动化测试吗
- » 更多知识库文章...

Copyright ©2015 Lovの沉没