您的潜力，我们的动力

Microsoft
微软（中国）有限公司

ASP.NET 2.0 角色控制与管理

讲师： 苏鹏 MSDN特约讲师

msdn

MSDN Webcasts

内容介绍

- 认证与授权机制
- 使用用户管理控件
- 成员资格与角色管理

ASP.NET 2.0 安全性必要性

对于网站而言，用户身份认证与权限管理是非常重要的部分。

通过用户名和密码，对用户进行身份验证，并指派他可访问的资源，这部分工作一直都是网站开发的重要内容。

ASP.NET 2.0 安全性必要性

在另外一些情况下，需要根据用户的身份进行权限识别，不同用户访问相同页面，也需要显示不同内容。

这涉及到用户权限管理部分，也是网站开发的核心内容。

ASP.NET 2.0 角色控制概述

- ASP.NET 2.0的membership和role manager能够非常好的解决这个问题，不但可以对用户的登陆信息进行统一管理，还可以就用户的权限进行分类管理，让开发者方便的就网站权限与安全性进行设定。
- ASP.NET 2.0的Login控件更提供了一种非常方便的建造登陆与用户管理信息的方法。

认证与授权

- ASP.NET通过与IIS协同工作来进行授权管理。共两种身份认证方式
  - 通过查询acls列表或者许可证来判定该访问是否拥有浏览的权利。
  - 通过URL认证

认证方式

当用户以访问某网站的时候。两种授权方式分别会进行不同的动作

第一种认证方式会根据用户的登陆信息来判定asp.net针对该用户所指定的系统帐号，然后再判断该系统帐号是否对被请求的本地资源有访问权限。

第二种身份认证方式通过检查asp.net配置文件来进行授权认证。

认证方式

- Asp.net的页面认证方式中，可以使用以下三种方式进行身份认证。
- 通过修改config文件中的authentication属性，可以配置不同的认证方式

认证方式

取值	描述
None	不进行授权与身份验证
Windows	基于windows身份验证，首先判断windows用户的身份和组
Form	基于cookie的身份认证机制
Passport	使用PassPort SDK进行二次开发

认证方式

```
<configuration>
    <system.web>
        <authentication mode="Forms"/>
    </system.web>
</configuration>
```

Windows认证方式

- Window认证方式通过使用windowsprincipal类对用户的windows身份进行判定，然后根据用户所属的windows身份组来进行认证。

DEMO WINDOWS AUTHENTICATION

- 需要在wen.config里加上
- <authentication mode="Windows"/>

Form认证方式

- From认证方式是在窗体内提供用户输入ID和密码的地方，并根据用户输入的ID和密码进行身份认证。
- Form认证方式同时还使用cookie记录用户的信息，当用户访问其他页面的时候，程序通过访问cookie来获得用户的身份信息。

Form认证方式配置文件


```
<configuration>
<system.web>
 <authentication mode="Forms"/> <authorization>
<forms name=".ASPXCOOKIEDEMO"
    loginUrl="login.aspx" protection="All" timeout="30"
    path="/">
<!-- protection="[All|None|Encryption|Validation]" -->
    </forms>
 <deny users="?" />
 </authorization>
 </system.web>
</configuration>
```

From认证配置文件讲解

- 配置文件中的属性意义如下表所示

属性	描述
loginUrl	指定一个用于登陆的页面
name	Cookie的名字，注意，如果一个服务器有很多应用的话，要给cookie其不同的名字
TimeOut	Cookie的存活时间默认值是30分钟
Protection	Cookie被保存的方式
Path	Cookie的保存时间

Protection属性

- Protection是用来描述cookie的保存方式的，有下列四个可选择项目

属性	描述
None	不使用任何方法保护cookie
Encryption	使用des或者三层des对cookie进行加密，但是并不对cookie传输中是否被监听或篡改进行监视
Validation	监视cookie，保证传输过程中不会被监听或者篡改。但是并不对cookie进行加密。
All	同时使用Encryption和Validation

使用文件记录用户的帐户和密码

- 用户还可以通过指定可访问的用户名和密码来指定访问用户。
- <authentication>
- <credentials passwordFormat="SHA1" >
- <user name="Mary" password="94F85995C7492EEC546C321821AA4BECA9A3E2B1"/>
- <user name="John" password="5753A498F025464D72E088A9D5D6E872592D5F91"/>
- </credentials>
- </authentication>

使用文件记录帐户信息

- 在指定密码的保存方式时，可以指定密码的存放方式，有3种方式。如下表所示

Hash 类型	描述
Clear	不加密进行存储
SHA1	使用SHA1进行加密
MD5	使用MD5进行加密

授权用户与角色

- 用户访问还可以通过定制访问规则来实现对用户的角色分配。
- <authorization>
- <allow users="someone@www.frontfree.com" />
- <allow roles="Admins" />
- <deny users="*" />
- </authorization>
- 以上代码指定只有someone@www.frontfree.com的用户可以访问该站点，并且该用户具有的权限是管理员

授权用户信息

- 在web.config里，同样可以通过配置all和deny属性来对访问用户的Id，访问方法进行设定
- <allow VERB="POST" users="John,Mary" />
- <deny VERB="POST" users="*" />
- <allow VERB="GET" users="*" />

User属性的描述

- User属性有两种配置方法
- 如下表所示

UserName	描述
*	所有用户
?	匿名用户

DEMO

- Form认证演示

使用用户管理控件

- Login,Loginstatus,CreateUserWizard控件示例
- LoginView示例
- ChangePassword示例

DEMO

成员管理

- 成员管理特性基于membership、menbershipuser两个类。可以使用memship类为asp.net创建用户
- membership类还可以完成以下工作
  - 建立一个新的membershipuser
  - 可以对用户身份进行验证
  - 找回一个membershipuser实例
  - 更新一个membershipuser实例
  - 通过不同条件寻找一个用户
  - 获得当前在线用户数量
  - 删除一个已经不再需要的帐户

成员管理

- 对于membership类可以完成以下工作
  - 访问一个membership示例的属性
  - 找回一个用户的密码
  - 修改一个用户的密码
  - 修改一个用户的密码问题以及密码问题的答案
  - 为一个已经因为多次尝试密码失败而锁定的用户解除锁定。

角色管理

- 角色管理基于role类实现。
- 通过角色管理类，可以实现以下工作
- 新建一种角色
- 删除一种角色
- 给一个用户分配角色
- 去除一个用户的角色
- 判断用户是否被授权给一个特殊的角色
- 在一种角色中寻找一个用户
- 从一个用户信息中获得他所具有的角色信息

创建用户

- 通过调用Membership的createuser方法，可以创建用户。需要注意的是membership的密码要求长于7位，并且需要至少包括一个特殊字符。

DEMO

- 创建用户

用户登陆以及访问用户属性

- 下面使用Membershiip中的validateuser方法来确认用户是否合法。
- Membership还提供了getuser方法返回一个membershipuser类，用以对获得用户的属性信息。

DEMO

- 用户登陆
- 用户帐户信息显示

更新用户属性

- 通过使用dataview和一个Membership相结合，可以方便的修改用户的注册信息。

DEMO

- 更改用户信息演示

解除锁定

- 当用户尝试密码次数过多时，系统会把用户的帐户锁定，可以通过Membershipuser的islockout属性来判断用户是否被锁定，如果用户被锁定了，可以使用unlockuser方法来解除锁定。

DEMO

- 解除用户锁定

删除用户

- 可以通过调用Membership的deleteuser方法来删除一个用户。并通过返回值来判定删除是否成功。

DEMO

- 删除用户

角色管理

- 角色管理是基于认证用户身份与权限的一种管理方式。下面的几个例子使用角色管理
- 添加和删除角色

添加和删除角色

- 通过使用roles的createrole和deleterole方法我们可以添加和删除角色，同时还可以调用

DEMO

- 添加删除role

为用户配置角色属性

- 可以通过roles类的addusertorole和removeuserfromrole方法来为用户指定一个角色或者移除一个角色

DEMO

- 为用户配置角色

使用role manage对页面进行授权

- 还可以通过使用role manage对指定角色的用户进行页面授权。
- 通过在web.config里进行配置

DEMO

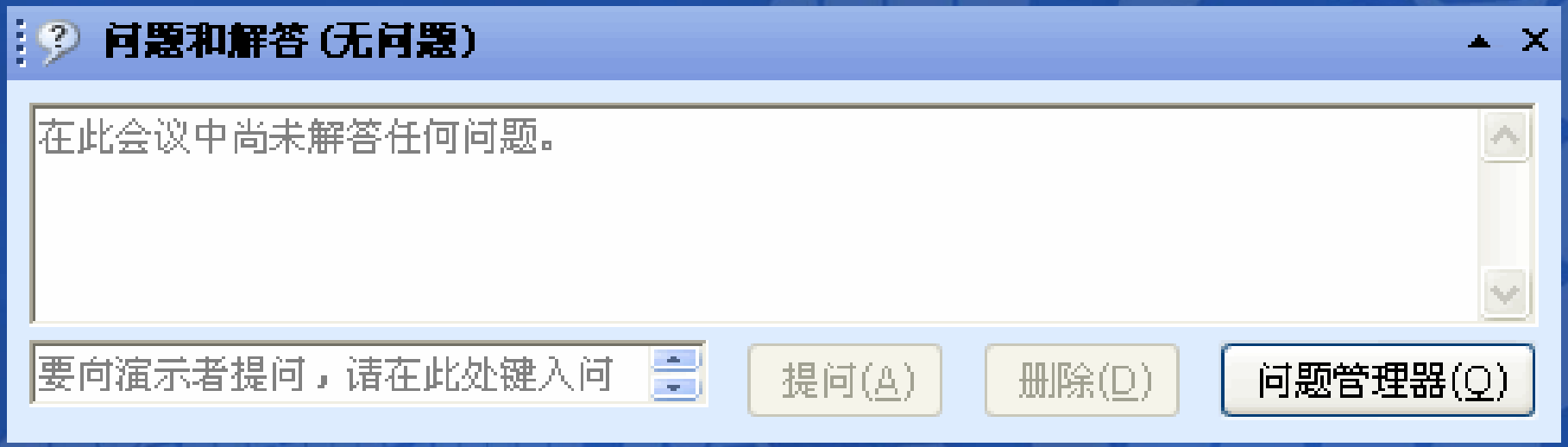
- 用户授权页测试

总结

- 认证与授权机制
- 使用用户管理控件
- 成员资格与角色管理

Question & Answer

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。




您的潜力，我们的动力
Microsoft
微软（中国）有限公司
Microsoft
msdn
MSDN Webcasts