## [ASP.NET身份验证机制membership入门——配置篇(2)](http://www.cnblogs.com/xlb2000/archive/2010/05/12/1733514.html)

　　　　上篇讲到如何配置membership以及ASP.NET Forms身份验证，本篇将对权限控制部分，做一个详细的解释。

　　　　在所有的基本配置都完毕后，我们还需要配置哪些目录允许被匿名访问，哪些是需要用户登录后允许访问的页面。

　　　　首先：我们在项目中建立一个admin文件夹，在admin文件夹中添加一个web.config文件，然后在其中的<system.web>节点下面添加如下代码：

<authorization>  
    <allow users="admin"/>  
    <deny users="\*"/>  
</authorization>

　　　　然后我们在admin目录下再添加一个页面，然后访问该页面，看一下效果。如果您按照我们上篇所说的内容全部正确配置了，那么你会发现，页面并没有显示出来，而是跳转到了我们之前在<authentication>下<forms>节点的LoginUrl属性所指向的页面，要求登录！这不就是我们需要的效果吗？

　　　　现在来解释一下上面配置的含义：

　　　　<allow>节点：顾名思义就是允许访问的意思，<allow users="admin"/>就是允许用户名为"admin"的用户访问。

           <deny>节点：就是禁止访问。这里用到了一个通配符“\*”，通配符有两个：\*代表所有用户，还有一个“?”代表所有匿名用户。所以<deny users="\*"/>就是不允许所有用户访问的意思啦，当然如果是：<deny users="?"/>那意思就是不允许所有匿名用户访问。

           需要说明的是：所有的配置都是按照从上到下的顺序来匹配的，一但匹配成功，就不再向下匹配。举个例子：

<authorization>  
    <allow users="admin"/>  
    <allow users="zhangsan"/>  
    <deny users="\*"/>  
</authorization>

           ASP.NET首先检测当前登录的用户名=="admin"？如果等于则不继续判断，直接允许该用户访问。如果不等于则继续判断当前登录的用户名=="zhangsan"等于，则允许访问，不等于则接着向下，读取到了<deny users="\*"/>这个配置，拒绝所有用户访问，跳转到LoginUrl指定的页面要求重新登录。

　　　　可是如果这样，只能实现目录一级的权限控制，如果要控制某个文件的访问权限，又该如何做呢？难道非得把这个文件放到一个文件夹中，然后再添加web.config进行控制么？答案是否定的，对于单个文件的访问控制，ASP.NET也有相应的配置：

[IMG_256](http://www.cnblogs.com/xlb2000/archive/2010/05/12/javascript:void(0);)

<configuration>  
    <location path="a.aspx">  
      <system.web>  
        <authorization>  
          <deny users="?"/>  
        </authorization>  
      </system.web>  
    </location>  
  
    <system.web>  
      <authorization>  
        <allow users="admin"/>  
        <allow users="zhangsan"/>  
        <deny users="\*"/>  
      </authorization>  
    </system.web>  
</configuration>

[IMG_257](http://www.cnblogs.com/xlb2000/archive/2010/05/12/javascript:void(0);)

　　　　看上面这个配置，我们可以在<system.web>节点之上（必须是上面）再增加一个<location>节点，通过path属性指明location内的配置是单独针对哪个文件即可。具体内容我就不多说了，想必大家也都看得懂。不过需要注意的一点就是：location节点可以有多个。这意味着同一个目录下的不同文件可以有不同的访问权限。

　　　　好了，配置到这里算是完成一多半了。还有一个问题就是：如果我们的用户比较多，那么需要在配置文件中把这些用户全部罗列其中。并且以后添加了新用户，还得继续修改配置，太麻烦了！该怎么办呢？通用的做法就是引入角色的概念。给所有的用户分配一个角色，比如：users，admin等。然后我们只需要控制这些角色的访问权限即可。以后添加了新用户只需给这个新用户分配角色，而不用去修改配置，实在是方便。其实，membership也提供了角色的概念，只需要简单的配置一下就可以实现了。

　　　　要实现角色功能，非常简单，我们还是去machine.config的<system.web>节点下面找到<roleManager>节点，然后整个节点复制过来到web.config中去，一般会有两个<add>子结点,我们删除一个，留下一个就可以了。全部内容如下：

[IMG_258](http://www.cnblogs.com/xlb2000/archive/2010/05/12/javascript:void(0);)

<roleManager>  
  <providers>  
    <add name="AspNetSqlRoleProvider"  
         connectionStringName="LocalSqlServer"  
         applicationName="/"   
         type="System.Web.Security.SqlRoleProvider, System.Web, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />  
  </providers>  
</roleManager>

[IMG_259](http://www.cnblogs.com/xlb2000/archive/2010/05/12/javascript:void(0);)

　　　　这个配置比较简单，各项配置和前面配置都一样，不再多说。稍微修改一下：

[IMG_260](http://www.cnblogs.com/xlb2000/archive/2010/05/12/javascript:void(0);)

<roleManager enabled="true" defaultProvider="myAspNetSqlRoleProvider">  
  <providers>  
    <add name="myAspNetSqlRoleProvider"  
         connectionStringName="ConnectionString"  
         applicationName="TestMembership"   
         type="System.Web.Security.SqlRoleProvider, System.Web, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />  
  </providers>  
</roleManager>

[IMG_261](http://www.cnblogs.com/xlb2000/archive/2010/05/12/javascript:void(0);)

　　　　注意：这个roleManager多了个属性enable="true"，这是因为角色管理默认情况下是关闭的，所以我们必须得设置为开启才行。

　　　　好了，所有的配置到这里算是全部结束了。欲知后事如何且听下回分解：《ASP.NET身份验证机制membership入门——编码篇》。

ss