How do I best handle role based permissions using Forms Authentication on my ASP.NET web application?

Asked 16 years, 3 months ago Modified 12 years, 8 months ago

Viewed 3k times

3

I'm using the ASP.NET Login Controls and Forms Authentication for membership/credentials for an ASP.NET web application.

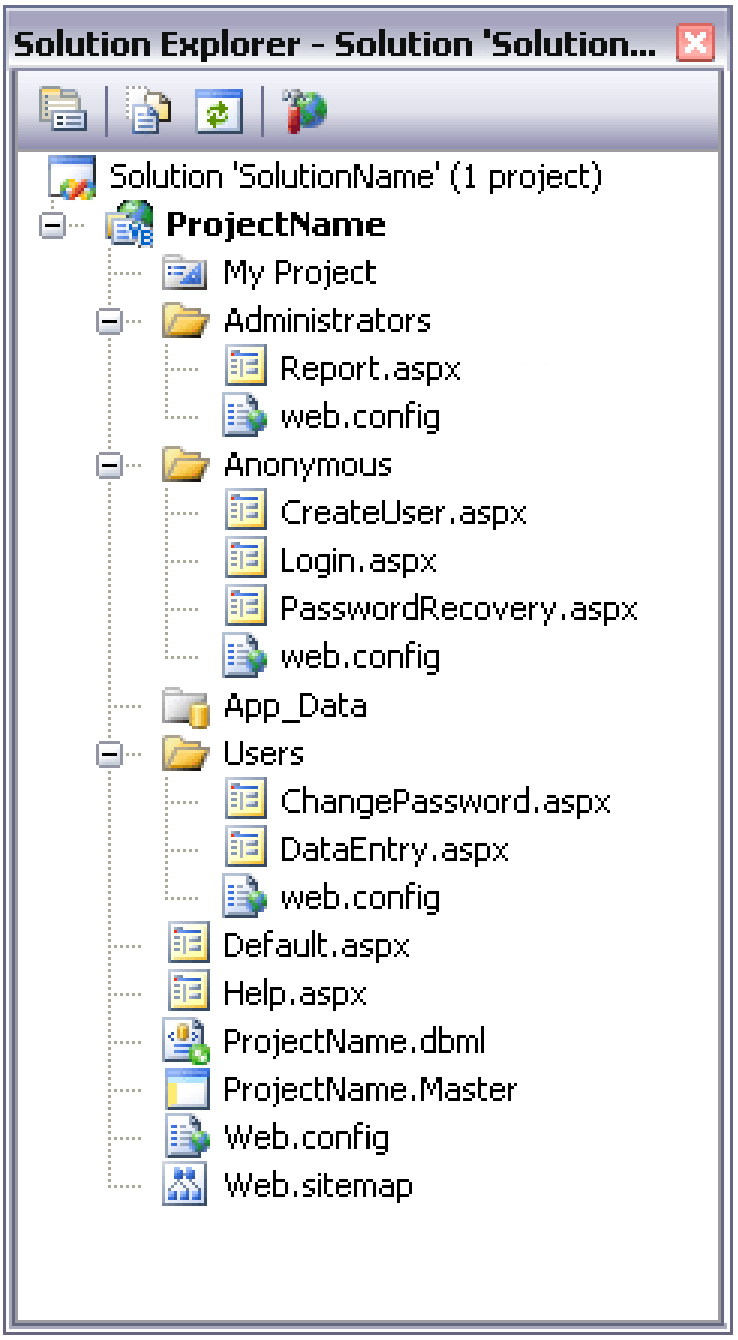
I've got two roles:

- Users
- Administrators

I want pages to be viewable by four different groups:

- Everyone *(Default, Help)*
- Anonymous *(CreateUser, Login, PasswordRecovery)*
- Users *(ChangePassword, DataEntry)*
- Administrators *(Report)*

Expanding on the example in the ASP.NET HOW DO I Video Series: Membership and Roles, I've put those page files into such folders:



And I used the ASP.NET Web Site Administration Tool to set up access rules for each folder.

It works but seems kludgy to me and it creates issues when Login.aspx is not at the root and with the ReturnUrl parameter of Login.aspx.

Is there a better way to do this? Is there perhaps a simple way I can set permissions at the page level rather than at the folder level?

asp.net forms-authentication

Share

Improve this question

Follow

edited May 23, 2017 at 12:19

Community Bot

1 • 1

asked Aug 28, 2008 at 20:05

Zack Peterson

57.3k • 80 • 210 • 280

3 Answers

Sorted by: Highest score (default)

1

A couple solutions off the top of my head.

1. You could set up restrictions for each page in your web.config file. This would allow you to have whatever folder hierarchy you wish to use. However, it will require that you keep the web.config file up to date whenever you add additional pages. The nice part of having the folder structure determine accessibility is that you don't have to think about it when you add in new pages.
2. Have your pages inherit from custom classes (i.e. EveryonePage, UserPage, AdminPage, etc.) and put a role check in the Page_Load routine.

Share Improve this answer

Follow

answered Aug 28, 2008 at 20:26

Kevin Pang
41.4k ● 38 ● 122 ● 173

1

One solution I've used in the past is this:

1. Create a base page called 'SecurePage' or something to that effect.
2. Add a property 'AllowedUserRoles' to the base page that is a generic list of user roles List or List where int is the role id.
3. In the Page_Load event of any page extending SecurePage you add each allowed user role to the AllowedUserroles property.
4. In the base page override OnLoad() and check if the current user has one of the roles listed in AllowedUserRoles.

This allows each page to be customized without you having to put tons of stuff in your web.config to control each page.

Share Improve this answer

Follow

answered Aug 28, 2008 at 20:31

JC

1

In the master page I define a public property that toggles security checking, defaulted to true. I also declare a string that is a ; delimited list of roles needed for that page.

in the page load of my master page I do the following

```
if (_secure)
{
  if (Request.IsAuthenticated)
  {
    if (_role.Length > 0)
    {
      if (PortalSecurity.IsInRoles(_role))
      {
        return;
      }
      else
      {
        accessDenied = true;
      }
    }
    else
    {
      return;
    }
  }
}

//do whatever you wanna do to people who dont have acc
page or whatever
```

also you'll have to put

at the top of your pages so you can access the extended properties of your master page

Share Improve this answer answered Aug 28, 2008 at 20:51

Follow

dfasdljkhfaskldjhfasklhf

1,162 2 10 18