check for login in every page

▲

4

▼

★

I've multiple web forms in my project. It is simple project. User authentication is done by checking valid username and password and role from database. In the Page_Load event for every web page I've added this code :

```
protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                if (Session["username"] == null & Session["role"] == null)
                {
                    Response.Redirect("WebLogin.aspx", false);
                    Context.ApplicationInstance.CompleteRequest();
                }
            }
            catch (Exception ex) { }
        }
```

so that if someone tries to go to a specific web page via URL, it will check for username and role. If it is null then redirect to login page.

Please can you suggest any other way of doing this, more efficient. It looks very unprofessional to add this code in every code behind page.

c#   asp.net   webforms

edited Dec 1 '16 at 7:21

iceDragon
575   1   12   23

asked Dec 1 '16 at 5:52
user4221591
878   3   16   39

2 Answers

Create new class PageBase that inherits Page class and override
the onload method for it then add you login check code

4

```
public abstract class PageBase : Page
{
    protected override void OnLoad(EventArgs e)
    {
        if (Session["username"] == null & Session["role"] == null)
        {
            Response.Redirect("WebLogin.aspx", false);
            Context.ApplicationInstance.CompleteRequest();
        }
    }
}
```
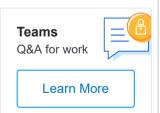
now for each of your forms just change the page class inheritance
from this:

```
public partial class HomePage : Page
```

to PageBase like this:

```
public partial class HomePage : PageBase
```

This will make your code more organized and clean

PUBLIC

🌐 **Stack Overflow**

Tags

Users

Jobs

Teams
Q&A for work

Learn More

answered Dec 1 '16 at 6:40

iceDragon

575 1 12 23

---

you can use asp.net forms authentication.

0

```
protected void Page_Load(object sender, System.EventArgs e)
{
    if (Context.User.Identity.IsAuthenticated)
    {
        // User is logged in, continue
    }
    else
    {
        //Invalid login...
        Session.Clear();
        Response.Redirect("~/Login.aspx");
    }
}
```

https://www.asp.net/web-forms/overview/older-versions-security/introduction/an-overview-of-forms-authentication-cs

answered Dec 1 '16 at 6:11

Kapila Perera

598 1 8 19