ASP.NET Cookie Login

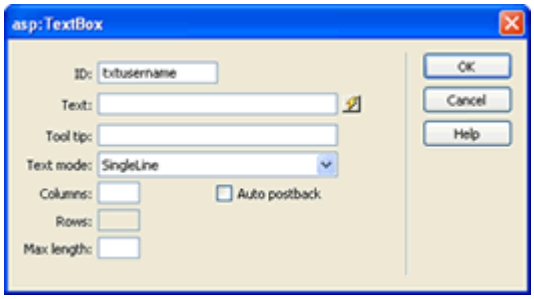
1. *Create dmx login form, dataset and validate it*

Note: The code in this article is for documents that use the ASP.NET/VB server model.

Step 1

1. Create a new asp.net/vb page and name it `login.aspx`.
2. Create a login form with 2 asp:textboxes and 1 asp:button. Use the asp.net web server controls from *Insert > ASP.NET Objects*. Make sure the controls are within the form tag (dmx should do this by default.)

For the username textbox:
ID = `txtusername`
Text mode = `SingleLine`

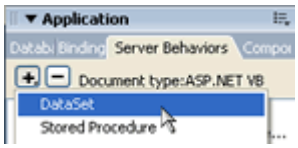


For the password textbox:
ID = `txtpassword`
Text mode = `Password`

For the button:
ID = `btn_login`
Text = `Login`

Step 2

Now let's create a DataSet to retrieve the users stored in the users table. We have to filter this dataset by the username and password that will be submitted by the login form we created for the user above.
1. *Application Panel > Server Behaviors > + > DataSet*

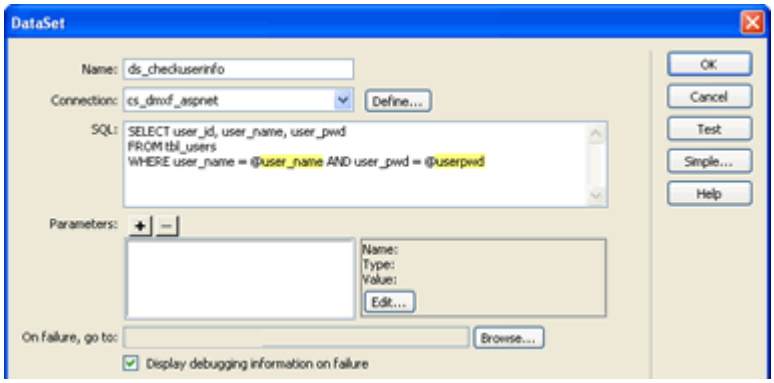


2. Name your dataset `ds_checkuserinfo`.

3. For the SQL select your user id, username and user password columns from the users table. Filter by the username and user password columns. The highlighted text in the image below are the names of 2 parameters that will reflect the values of the 2 textboxes.

Here is my SQL. Obviously, change the names of the columns and table to reflect your own, but do not change the parameter names (that are highlighted in the image and in pink below):

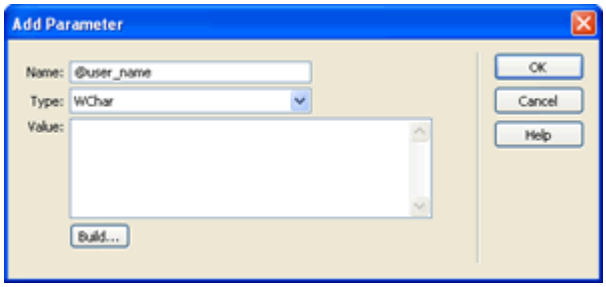
```
SELECT user_id, user_name, user_pwd
FROM tbl_users
WHERE user_name = @user_name AND user_pwd = @userpwd
```



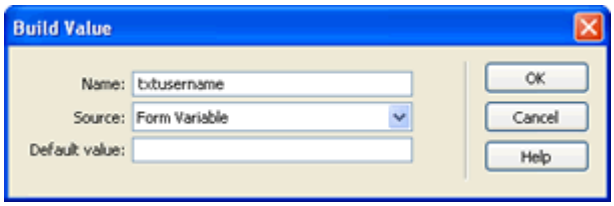
Step 3

Now let's add the parameters which will reflect the values submitted in the 2 textboxes of the form, one which is named `txtusername` and the other `txtpassword`.

1. From the Parameters section, click the plus (+) sign.
Name the parameter `@user_name`. Change the Type to `WChar`. Click the `Build` button.



2. Name the build value `txtusername` and get the source from the Form Variable and click `OK`. (This is using the value of whatever is submitted in the `txtusername` textbox as the value of the parameter.)



Step 4

Now we want to add the second parameter, which will be the password.

1. Do the same as in Step 3 but in the Add parameter dialog box change the name of the parameter to `@userpwd` and in the build dialog box change the name to `txtpassword`. Click *OK* to close the Add Parameter dialog box.
2. Click OK to close the `DataSet` dialog box.

Step 5

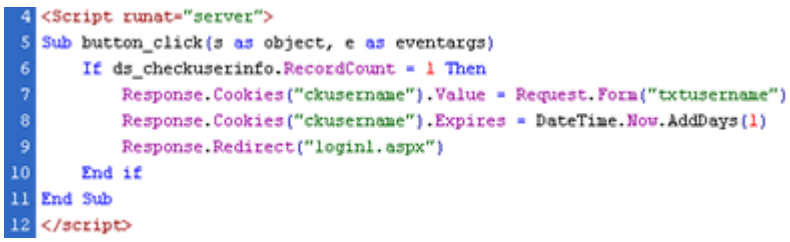
Now let's create the source code that will actually validate the DataSet. The source code has to find out whether a record was found in that DataSet we created and create a cookie if so, and if not, just reload the page.

1. Switch to *Code* view and add the following code above the `<MM:dataset` code.

```
<Script runat="server">

Sub button_click(s as object, e as eventargs)
      If ds_checkuserinfo.RecordCount = 1 Then
            Response.Cookies("ckusername").Value = Request.Form("txtusername")
            Response.Cookies("ckusername").Expires = DateTime.Now.AddDays(1)
            Response.Redirect("login1.aspx")
      End if
End Sub

</script>
```



Here's the **C# code**:

```
<script runat="server">
void button_click(Object s, EventArgs e)
{
if (ds_checkuserinfo.RecordCount == 1)
{
Response.Cookies["ckusername"].Value = txtusername.Text;
Response.Cookies["ckusername"].Expires = DateTime.Now.AddDays(1);
Response.Redirect("login1.aspx");
}
}
</script>
```

```
<Script runat="server">
void button_click(Object s, EventArgs e)
{
    if (ds_checkuserinfo.RecordCount == 1)
    {
        Response.Cookies["ckusername"].Value = txtusername.Text;
        Response.Cookies["ckusername"].Expires = DateTime.Now.AddDays(1);
        Response.Redirect("login1.aspx");
    }
}
</script>
```

The code above checks to see whether the dataset named `ds_checkuserinfo` returned a record. If it does, that means the username and password submitted are valid; thus we create a cookie named `ckusername` using the value of the username and redirect to `login1.aspx` page. You can redirect to any page you like.

Step 6

Finally we need to make sure that when the user clicks the login button the code in Step 5 executes.

1. Switch to Design view and highlight the Login button.
2. Switch to Code view and add the following pink attribute to the `asp:button` code:

```
<asp:Button ID="btn_login" Text="Login" onClick="button_click" runat="server" />
```

This will execute the Sub Routine named `button_click` when the button is clicked.

2. Protect Page

Step 1

Let's create the code for a page that you don't want to users to access unless they are logged in.

1. Switch to *Code* view and place the following code right underneath the page directive.

```
<Script runat="server">

Sub Page_Load(s as object, e as eventargs)

      Dim objckusername As HttpCookie
      objckusername = Request.Cookies("ckusername")

            If objckusername Is Nothing Then
                  Response.Redirect("login.aspx")
            Else
                  lblckusername.Text = Server.HtmlEncode(objckusername.Value)
            End If

End Sub

</script>
```

```
<Script runat="server">

Sub Page_Load(s as object, e as eventargs)

Dim objckusername As HttpCookie
objckusername = Request.Cookies("ckusername")

    If objckusername Is Nothing Then

        Response.Redirect("login.aspx")

    Else

        lblckusername.Text = Server.HtmlEncode(objckusername.Value)

    End If

End Sub

</script>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

In C#:

```
<script runat="server">
void Page_Load(Object s, EventArgs e)
{

HttpCookie objckusername = Request.Cookies["ckusername"];

if (objckusername == null)
{
Response.Redirect("login.aspx");
}
else
{
lblckusername.Text = Server.HtmlEncode(objckusername.Value);
}

}
</script>
```

```
<Script runat="server">
void Page_Load(Object s, EventArgs e)
{
    HttpCookie objckusername = Request.Cookies["ckusername"];

        if (objckusername == null)
        {
            Response.Redirect("login.aspx");
            }
        else
        {
            lblckusername.Text = Server.HtmlEncode(objckusername.Value);
        }
}
</script>
```

Note

The code above checks to see if the cookie named `ckusername` is empty; if it is, we redirect the user to the login page, and if it's not, then we stuff the value of the cookie `ckusername` into a variable so we could use it to display on our page.

2. You can now display the value of the cookie anywhere on the page with this code:

```
<asp:label id="lblckusername" runat="server" />
```

3. Create Logout

Step 1

Let's create the code to logout the user. We need to create a new page that will delete the cookie when accessed.

1. Create a new page and name it `logout.aspx`.
2. Switch to Code view and place the following code right underneath the page directive.

```
<script runat="server">

Sub Page_Load(s as object, e as eventargs)
      Dim objckusername As HttpCookie
            objckusername = Request.Cookies("ckusername")

      If objckusername Is Nothing Then
              'if cookie does not exist user is not logged in
              'and shouldn't be accessing this page anyhow - send user away
            Response.Redirect(Request.ServerVariables("HTTP_REFERER"))
      Else    'if cookie exists user is logged in and wants to logout so kill cookie
            objckusername.Expires = DateTime.Now.AddDays(-1)
            Response.Cookies.Add(objckusername)
            Response.Redirect(Request.ServerVariables("HTTP_REFERER"))
      End If

End Sub

</script>
```

```
<script runat="server">
Sub Page_Load(s as object, e as eventargs)
       Dim objckusername As HttpCookie
           objckusername = Request.Cookies("ckusername")

    If objckusername Is Nothing Then
         'if cookie does not exist, user is not logged in
         'and shouldn't be accessing this page anyhow, send user away
         Response.Redirect(Request.ServerVariables("HTTP_REFERER"))
    Else 'if cookie exists, user is logged in, wants to logout, so kill cookie
         objckusername.Expires = DateTime.Now.AddDays(-1)
         Response.Cookies.Add(objckusername)
         Response.Redirect(Request.ServerVariables("HTTP_REFERER"))
    End If
End Sub
</script>
```

In C#:

```
<script runat="server">
void Page_Load(Object s, EventArgs e)
{
HttpCookie objckusername = Request.Cookies["ckusername"];

if (objckusername == null)
{

/* if cookie does not exist user is not logged in
and shouldn't be accessing this page anyhow - send user away */

Response.Redirect(Request.ServerVariables["HTTP_REFERER"]);
}

else

/*if cookie exists, user is logged in, wants to logout, so kill cookie */

{
objckusername.Expires = DateTime.Now.AddDays(-1);
Response.Cookies.Add(objckusername);
Response.Redirect(Request.ServerVariables["HTTP_REFERER"]);
}
</script>
```

Note

The code above checks to see first if the cookie `ckusername` exists. If it does not, it redirects page back to the previous page. We do this, because if this page is accessed by accident when a user is not logged in, it will throw an error, because it's trying to delete a cookie that doesn't exist. Anyway, after checking for that, the code deletes the cookie `ckusername` and redirects to the previous page.

You're done!

By: Omar Elbaga (www.dmxfire.com)