

How can I create a view that has different displays according to the role the user is in?

Asked 15 years, 11 months ago Modified 8 years, 9 months ago

Viewed 18k times



I want to create a view that has different displays according to the role the user is in.

16



Should I create a different view for different roles or should I check the roles on the View page itself rather than in the actions?



How would I check the role on the view page?



asp.net

asp.net-mvc

roles

Share

Improve this question

Follow

edited Sep 23, 2009 at 23:15



GEOCHET

21.3k ● 15 ● 77 ● 99

asked Jan 3, 2009 at 15:31



devforall

7,327 ● 12 ● 38 ● 42

8 Answers

Sorted by:

Highest score (default)



26



Or should i use check the roles on the Veiw page its self rather than on actions, if so can someone plz show me how do check that on view page

You need to do both. Check roles on actions as a security measure and check roles on views to enable/disable specific controls.

Within your view page the long form of checking a role is

```
HttpContext.Current.User.IsInRole("Administrator")
```

many developers will create page helper methods so you can end up with something more concise for your application like

```
public static bool IsAdmin(this ViewUserControl pg)
{
    return pg.Page.User.IsInRole("Administrator")
}
```

then in your view you can just use *this.IsAdmin()*

To keep your view clutter down look into using partial views

```
<% if (IsAdmin())
{
    Html.RenderPartial("AdminPanel");
}
```

```
}  
else  
{  
    Html.RenderPartial("UserPanel");  
}  
%>
```

Share Improve this answer

edited Jan 3, 2009 at 16:46

Follow

answered Jan 3, 2009 at 16:24



Todd Smith

17.3k ● 11 ● 62 ● 82

i have done the same but i am not able to find IsAdmin in view of controller..why is it so?? – [Neel](#) Jun 12, 2013 at 9:48

It's a helper method you add yourself. – [Todd Smith](#) Jun 13, 2013 at 16:48

Or Maybe Yo have Forgotten `static` statement in declaring function ! – [Moslem7026](#) Aug 9, 2013 at 20:45

This solution breaks as soon as you use caching, and you should be using caching. – [Chris Moschini](#) Dec 12, 2017 at 18:33



6



If the display changes based on the role -- and the change is small -- then I would do the check in the view. If certain views are restricted based on the role, then I would do the check in the controller. If the views are completely different (this would be hard to imagine), then separate views per role may be appropriate.



You may want to abstract out certain role-specific view components into partial views to simplify your view logic - basically you only have to check to include the partial or not based on the role.

Also, other than to check for "IsAuthenticated", I would move the role checking logic to the controller and pass (as data) to the view information on which elements to include/exclude based on role. This keeps the actual business logic from bleeding into your view.

Share Improve this answer

answered Jan 3, 2009 at 16:09

Follow



[tvanfosson](#)

532k ● 102 ● 699 ● 798

In my project I need to maintain branch information. A system administrator should be able to see/maintain all branch information. A branch administrator should only see/maintain their own branch information. Two separate views I think.

– [David Clarke](#) Mar 17, 2011 at 22:19



3



If you are using MVC the whole point of development is to keep the logic out of the view and in the controller. It seems to me like you'd be better off on a WebForms development track than an MVC track.

All that being said, I do an Admin check on a lot of my pages by using a check like this:



```
<% if ((bool)ViewData["Admin"]) { %>
    <!-- Show admin controls here -->
```

```
<% } %>
```

But if you are attempting to build actual logic into the View then you need to figure out what you can push back to the controller to do the work and have the view be as dumb as possible, acting on flags sent to it.

Share Improve this answer

answered Jan 3, 2009 at 15:53

Follow



[thaBadDawg](#)

5,220 ● 6 ● 36 ● 44



2

without researching the exact mechanism asp.net mvc uses for roles i would scream no for putting any of your business logic in the view which is what you are doing if you are checking roles in the view



Share Improve this answer

answered Jan 3, 2009 at 15:46

Follow



[IAmCodeMonkey](#)

1,568 ● 1 ● 11 ● 11

◀ Generally I would agree with this, but I'm not so dogmatic that I would insist on generating a completely different view just to change add/remove a particular element based on the role of the user. – [tvanfosson](#) Jan 3, 2009 at 16:12 ▶



2

Yeah that was something that was bothering me as well ... but at the same time it seems ridiculous to load whole different view for such a small change.



btw how did you set this up in your controller. Right now, my controller looks something like the code below, which I don't think is correct.



```
[Authorize(Roles = "Admin, Member")]  
public ActionResult RegistrationInformation()  
{  
  
    return View();  
}
```

Share Improve this answer

edited Mar 24, 2016 at 21:10

Follow



Laurel

6,163 ● 14 ● 33 ● 59

answered Jan 3, 2009 at 16:28



devforall

7,327 ● 12 ● 38 ● 42

That is correct but I plan on implementing a custom Authorize class that proves enumerated roles such as Roles.Admin | Roles.Member. That way I get compile time checks if a role still exists. – [Todd Smith](#) Jan 3, 2009 at 17:13



0



I'm not that familiar with ASP.NET MVC (yet) but can't you do some kind of conditional filter in the View? If the Controller passes the role to the View, then you should be able to do a conditional filter and display a certain block of code if the user is an admin. If you want to display a totally separate page, then you'd have a multiple Views, otherwise you can use one and do some conditional.



In Ruby on Rails it would be something like (sorry, I don't know ASP.NET MVC really yet):

```
<% if @user.admin? # is the user an admin %>
  <h3>Admin Tools</h3>
<% end %>
<p>Regular site content</p>
```

In Rails you would load the extra content from partials; ASP.NET MVC has something similar but I forget what it's called. Maybe look into that?

Sorry I can't be of more help -- like I said I haven't really gotten to play with ASP.NET MVC.

[Share](#) [Improve this answer](#)

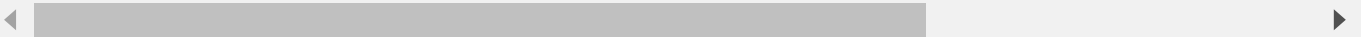
answered Jan 3, 2009 at 15:53

[Follow](#)



[Wayne Molina](#)

19.6k ● 26 ● 102 ● 164





0



I have base model which from all others models extend. In this model i have loaded the user's roles. Its based on `httpcontext.user.isinrole()` method. All views are strong typed expecting the base model type. So i can always check in all views something like `Model.CurrentUser.IsInRoles(Role1 | Role2)`. Not only in views of course, but in hole application.

Share Improve this answer

answered Aug 13, 2009 at 19:02

Follow



0



I like to have full control over this in the view, and I find that:

```
<% if (User.IsInRole("Super User")) { %>
    <h1>Hello world!</h1>
<% } %>
```

Works for most scenarios. It also allows you to easily do conditional formatting for other roles, e.g "Content Manager", "Registered", etc.

I do like Todd Smith's answer, because you might change the name of the Admin role, and that will require only one change, whereas, if you put the "Super User" or "Administrator" string directly in the view, you will have to change it wherever you've used the value.

Share Improve this answer

answered Mar 23, 2010 at 15:20

Follow



Roger Rogers

535 ● 1 ● 4 ● 15

