How to add claims in ASP.NET Identity

51

25

I am trying to find a document or example of how you would add custom claims to the user identity in MVC 5 using ASP.NET Identity. The example should show where to insert the claims in the OWIN security pipeline and how to persist them in a cookie using forms authentication.

asp.net-mvc-5   asp.net-identity

asked Dec 4 '13 at 19:16

Kevin Junghans
15.5k   4   39   57

4 Answers

39

Perhaps the [following article](#) can help:

```
var claims = new List<Claim>();
claims.Add(new Claim(ClaimTypes.Name, "Brock"));
claims.Add(new Claim(ClaimTypes.Email, "brockallen@gmail.com"));
var id = new ClaimsIdentity(claims,DefaultAuthenticationTypes.ApplicationCookie);

var ctx = Request.GetOwinContext();
var authenticationManager = ctx.Authentication;
authenticationManager.SignIn(id);
```

answered Dec 5 '13 at 13:10

Vlince
4,290   6   38   58

59

The correct place to add claims, assuming you are using the ASP.NET MVC 5 project template is in `ApplicationUser.cs` . Just search for `Add custom user claims here` . This will lead you to the `GenerateUserIdentityAsync` method. This is the method that is called when the ASP.NET Identity system has retrieved an ApplicationUser object and needs to turn that into a ClaimsIdentity. You will see this line of code:

```
// Note the authenticationType must match the one defined in
CookieAuthenticationOptions.AuthenticationType
var userIdentity = await manager.CreateIdentityAsync(this,
DefaultAuthenticationTypes.ApplicationCookie);
```

After that is the comment:

```
// Add custom user claims here
```

And finally, it returns the identity:

```
return userIdentity;
```

So if you wanted to add a custom claim, your `GenerateUserIdentityAsync` might look something like:

```
// Note the authenticationType must match the one defined in
CookieAuthenticationOptions.AuthenticationType
var userIdentity = await manager.CreateIdentityAsync(this,
DefaultAuthenticationTypes.ApplicationCookie);

// Add custom user claims here
userIdentity.AddClaim(new Claim("myCustomClaim", "value of claim"));

return userIdentity;
```

answered Mar 2 '15 at 16:32

dprothero
2,277   1   14   24

Model or the ViewBag. But technically, it will work directly in the Razor view as I've written it here. – dprothero Apr 5 '15 at 3:53 ✎

4   In VS2015 you will find the `GenerateUserIdentityAsync()` method in the `IdentityModels.cs` file within the `App_Start` directory. – MATT BAKER Jul
     22 '15 at 14:52

3   @MatthewT.Baker: I think you mean that IdentityModels.cs is in the Models directory, not App_Start? Also, I think this solution is incomplete. The
     documentation from Microsoft on adding custom claims (and I think the comments so far) do not explain the different effects of using
     userIdentity.AddClaim (which adds the claim to the cookies, but doesn't seem to change the AspNetUserClaims table), and manager.AddClaim(userId,
     claim), which adds the claim to the AspNetUserClaims table (NB it adds a new record even if an existing record with the same values exists), but not the
     cookies. – patrickjlee May 21 '16 at 22:39

If you want to add custom claims at the time of registration then this code will work:

4

```
var user = new ApplicationUser
{
    UserName = model.UserName,
    Email = model.Email
};
var result = await UserManager.CreateAsync(user, model.Password);
// Associate the role with the new user
await UserManager.AddToRoleAsync(user.Id, model.UserRole);
// Create customized claim
await UserManager.AddClaimAsync(user.Id, new Claim("newCustomClaim",
"claimValue"));
if (result.Succeeded)
{...etc
```

answered Sep 26 '17 at 10:00

Uwe Köhler
51   5

---

you can do the following in WEB API C #

```
        identity.AddClaim(new Claim(ClaimTypes.MobilePhone, user.Celular));
        identity.AddClaim(new Claim("FullName", user.FullName));
        identity.AddClaim(new Claim("Empresa", user.Empresa));
        identity.AddClaim(new Claim("ConnectionStringsName",
  user.ConnectionStringsName));
```

....

answered Apr 2 at 17:13

Maurico Bello
36   8

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up   OR SIGN IN WITH   G Google   Facebook ✕