عنوان: **مدیریت سفارشی سطوح دسترسی کاربران در MVC**
نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۱/۱۰/۰۸ ۲۳:۱۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, ASP.Net MVC

فرض کنید در پروژه‌ی جاری خودتون قصد دارید یک سیستم مدیریت سطوح دسترسی کاربران رو با انعطاف بالا پیاده سازی کنید . مثلا سیستم شما دارای صفحات مختلفی هستش که هر گروه کاربری اجازه دسترسی به هر صفحه رو نداره ... هدف اینه که شما این گروه‌های کاربری و سطوح دسترسی برای هر گروه رو برای سیستم مشخص کنید .
مثلا فقط کاربرانی که دسترسی admin دارن بتونن به صفحات مدیریتی دسترسی داشته باشن و ....
برای این منظور در دات نت کلاسی با نام Role Provider وجود داره که در ادامه‌ی این مبحث به کار با اون میپردازیم.
مثلا فرض کنید قرار بر اینه که سطوح دسترسی رو از بانک اطلاعاتی استخراج کنیم . کلاس مذکور در فضای نام System.Web.Security قرار گرفته . برای شروع ما نیاز داریم یک نمونه از این کلاس رو پیاده سازی کرده و چند تابع از اون رو بازنویسی کنیم .

پیاده سازی کلاس به این صورته :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Security;

namespace Myproject.Security
{
    public class CustomRoleProvider : RoleProvider
    {

    }
}
```

خب در مرحله‌ی بعد دو تابع از این کلاس رو بازنویسی میکنیم . اول تابع GetRolesForUser که در این مقاله وظیفه‌ی استخراج لیست مجوزها برای هر کاربر رو از بانک اطلاعاتی داره که به شکل زیر پیاده سازی میشه .

```
public override string[] GetRolesForUser(string username)
        {
            using (DatabaseEntities db = new DatabaseEntities())
            {
                User user = db.Users.FirstOrDefault(u => u.UserName.Equals(username,
StringComparison.CurrentCultureIgnoreCase));

                var roles = from ur in user.UserRoles
                            from r in db.Roles
                            where ur.RoleId == r.Id
                            select r.Name;
                if (roles != null)
                    return roles.ToArray();
                else
                    return new string[] {}; ;
            }
        }
```

همونطور که میبینید در تابع بالا از کلاس CustomRoleProvider ما عملیات استخراج لیست مجوزهای دسترسی مربوط به هر کاربر رو از بانک اطلاعاتی انجام دادیم . توجه داشته باشین که شما میتونید این واکشی رو از هر محیط دیگه ای جز بانک اطلاعاتی هم بسته به نوع کارتون انجام بدید .

تابع بعد IsUserInRole نام داره که با بازنویسی اون مشخص میکنیم که ایا یک کاربر دارای مجوز لازم برای دسترسی هست یا نه . اون رو به شکل زیر بازنویسی میکنیم.

```
public override bool IsUserInRole(string username, string roleName)
        {
```

```
            using (DatabaseEntities db = new DatabaseEntities())
            {
                User user = db.Users.FirstOrDefault(u => u.UserName.Equals(username,
StringComparison.CurrentCultureIgnoreCase));

                var roles = from ur in user.UserRoles
                            from r in db.Roles
                            where ur.RoleId == r.Id
                            select r.Name;
                if (user != null)
                    return roles.Any(r => r.Equals(roleName,
StringComparison.CurrentCultureIgnoreCase));
                else
                    return false;
            }
        }
```

همونطور که شاهد هستید در تابع بالا بعد از واکشی لیست مجوزهای ثبت شده برای هر کاربر بررسی انجام میشه که ایا اولا کاربر یک کاربر ثبت شده هست و ثانیا اینکه ایا درخواستی که ارسال کرده برای دسترسی به یک بخش مجوز اون رو داره یا خیر ...

این **نکته** رو یاد آور بشم که این توابع رو میشه به شکل‌های مختلفی پیاده سازی کرد و اونچه که در اینجا نوشته شده **فقط جهت مثال** هستش . مثلا تابع IsUserInRole رو میشه به شکل زیر هم نوشت و این بسته به شرایط کاری داره که قصد انجام اون رو دارید ....

```
public override bool IsUserInRole(string username, string roleName)
        {
  return this.GetRolesForUser(username).Contains(roleName);
        }
```

خب میرسیم به بخش معرفی این Provider در web.config که به صورت زیر انجام میشه ...

```
<system.web>
    ...
    <rolemanager cacherolesincookie="true" defaultprovider="CustomRoleProvider" enabled="true">
        <providers>
            <clear />
            <add name="CustomRoleProvider" type="Myproject.Security.CustomRoleProvider" />
        </providers>
    </rolemanager>
    ...
</system.web>
```

توجه داشته باشید که مجوزهای هر کاربر با معرفی بالا بعد از یک بار واکشی در کوکی ذخیره میشه و دیگه هر بار، بار اضافه برای واکشی از بانک اطلاعاتی به برنامه تحمیل نمیشه ...

حالا به این صورت میتونیم مثلا یک Controller رو محافظت کنیم در برابر درخواست از جانب کاربرانی که سطح دسترسی به اون رو ندارند .

```
using System;
using System.Web.Mvc;

namespace MyProject.Areas.Admin.Controllers
{
    [Authorize(Roles = "Administrators")]
    public class HomeController : Controller
    {
        //
        // GET: /Admin/Home/

        public ActionResult Index()
```

```
        {
            return View();
        }
    }
}
```

توجه داشته باشید که کنترل مجوز برای بررسی وجود مجوز در بخش‌های کوچکتر هم مانند اکشن‌ها و ... میتونه در نظر گرفته بشه .

نظرات خوانندگان

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۱۰/۰۹ ۰:۱۵

فکر می‌کنم خلاصه مطلب شما می‌شه :
نمونه گیری از اینترفیس role provider و پیاده سازی شخصی این کلاس
نظر دوستان رو جلب می‌کنم به این سری مطلب که هرچند قدیمیه ولی واقعا مفیده.
به خود من که خیلی کمک کرد.

نویسنده: سعید
تاریخ: ۱۳۹۱/۱۰/۰۹ ۲:۳۱

ممنون. کلاس‌های users و roles به چه نحوی تعریف شدن؟ به نظر می‌رسه رابطه many-to-many است.

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۱/۱۰/۰۹ ۱۰:۲

سه جدول user و UserRoles و Roles در نظر گرفته شده که جدول UserRoles بین دو جدول دیگه واسط هست . هر کاربر میتونه چندین مجوز داشته باشه و هر مجوز میتونه برای چندین کاربر باشه . ارتباط n به n بین دو جدول Roles و user به واسطه‌ی جدول UserRoles برقراره.
توجه به این نکته حائز اهمیته که این مدل فقط یک مثال ساده صرفا مربوط به این مقالست و هیچ جنبه‌ی دیگه ای نداره و همونطور که تو مقاله هم به صورت ضمنی اشاره شد به هر روش دیگه ای بسته به کارتون میشه پیاده سازیش کرد.

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۱/۱۰/۰۹ ۱۰:۱۳

دوست عزیز RoleProvider یک اینترفیس نیست بلکه یک کلاس مشتق شده از ProviderBase هستش که اونم از کلاس system.object مشتق شده . برای اطلاعات بیشتر توجهتون رو به صفحه‌ی مربوطه در msdn جلب میکنم : RoleProvider

نویسنده: علیرضا
تاریخ: ۱۳۹۱/۱۰/۰۹ ۱۱:۱۵

دوست عزیز سلام،
ممنون بابت آموزش خوبتون امکانش هست یه کم در باره این بخش از صحبت تون و نحوه پیاده سازیش بیشتر توضیح بدید؟

[توجه داشته باشید که کنترل مجوز برای بررسی وجود مجوز در بخش‌های کوچکتر هم مانند اکشن‌ها و ... میتونه در نظر گرفته بشه .]

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۱/۱۰/۰۹ ۱۲:۱۹

شما فیلتر Authorize رو میتونید برای یک اکشن هم در نظر بگیرید مثل مثال زیر . با این کار دیگه کل کنترلر به مجوزی که ذکر میکنید محدود نمیشه و فقط یک اکشن از اون محدود میشه...

```
using System;
using System.Web.Mvc;

namespace MyProject.Areas.Admin.Controllers
{
    public class HomeController : Controller
    {
        //
```

```
        // GET: /Admin/Home/

        public ActionResult Index()
        {
            return View();
        }

        [Authorize(Roles = "Administrators")]
        public ActionResult ViewProfile()
        {
            return View();
        }
    }
}
```

نویسنده: رامین
تاریخ: ۱۳۹۱/۱۰/۰۹ ۱۲:۲۹

ممنون بابت مطلب کامل شما ،
سوالی داشتم آیا در مبحث ASP.NET هم میتوان از این روش استفاده نمود؟

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۱/۱۰/۰۹ ۱۲:۵۸

خواهش میکنم دوست عزیز . بله میشه استفاده کرد . دعوتید به مطالعه‌ی این مبحث در MSDN :

How To: Use Role Manager in ASP.NET

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۱۰/۱۰ ۹:۱۰

خوبه ولی برای انعطاف پذیری بیشتر، من و تیمم با استفاده از Reflection، اسامی متدهایی که خروجی ActionResult دارند رو بازیابی می‌کنیم و در سیستم امنیت و پایه برای مدیر امنیت و برنامه نویس نشان میدیم و اونها می‌تونن دسترسی رو بر این مبنا تنظیم کنن. وجود یک Contoller پایه و یک فیلتر برای اون با override کردن متد OnActionExecuting از الزامات کار هست.

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۱/۱۰/۱۰ ۱۲:۷

جالبه . من در پروژه ای که در حال کار کردن روی اون هستم که یک CMF هست تقریبا همین کار رو انجام میدم . به این شکل که SecurityEngine این سیستم اکشن‌های مشخص شده که نیاز به سطح دسترسی دارن رو بر اساس یک attribute سفارشی شده استخراج میکنند به واسطه‌ی reflection و مدیران سیستم میتونن برای هر اکش roleهای دلخواه رو انتصاب بدن و سطوح هر اکشن رو توسط یک AuthorizeAttribute و با بازنویسی AuthorizeCore از پایگاه داده استخراج میکنم . هر کاربری که سطح دسترسی معین رو داشته باشه میتونه به اون اکشن دسترسی پیدا کنه ... البته پشت این SecurityEngine یک CashManagment هم قرار داره که از بار اظافه‌ی هر بار رفتن به پایگاه داده برای استخراج سطوح جلوگیری میکنه و این پروسه رو مدیریت میکنه

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۱۰/۱۰ ۱۲:۵۵

وجود یک مدیر Cache برای سیستم سطح دسترسی برای من ناملموس هست. اینکه چطور میشه در یک سیستم با حدود 5000 کاربر همزمان، نقش‌ها رو در Cache نگه داشت؟ ساختار ذخیره سازی در حافظه به چه شکل هست؟ ترجیح من این هست که به ازای هر درخواست، پایگاه داده مورد اشاره قراره بگیره.

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۱/۱۰/۱۰ ۱۳:۲۶

نقش‌های کاربران در کش نگهداری نمیشه . اونها در کوکی‌ها ذخیره میشن برای بهینه کردن و کم کردن فشار از روی سیستم در

ازای دسترسی هر با به پایگاه داده . با مشخص کردن cacherolesincookie مربوط به rolemanager در web.config....

و اینکه یک سیستم مجوزهای 5000 کاربر یا حتی 5 میلیون کاربر رو در کش ذخیره کنه برای من هم نه حتی ناملموس بلکه یک اشتباه محض هست چون بعد از یک مدت با برخورد مدیران it سرور و suspend شدن مواجه میشیم ... کار CacheManager که ازش حرف زدم ذخیره کردن مجوزهای اکشن هاست در حافظه که با هر بار Request برای اکشن دیگه نریم و مجوزهای اون رو از پایگاه داده بخونیم . شما بر فرض 100 اکشن مدیریتی دارید که هر کدوم هم 10 مجوز براشون مشخص شده در کل حساب بشه چیز زیادی نمیشه ولی در عوض سرعت بالا رو در قبال نرفتن به پایگاه داده براتون به ارمغان میاره.

اینکه بیان کردید به ازای هر درخواست پایگاه داده مورد اشاره قرار بگیره خیلی رو کارایی سیستم تاثیر منفی میزاره . به استناد گفته‌ی خودتون 5000 کاربر داریم که بر فرض در ثانیه در بدبینانه‌ترین حالت ممکن 100 درخواست ارسال میکنند . یعنی ما باید در هر ثانیه برای استخراج سطوح هر اکشن 100 بار به بانک اطلاعاتی مراجعه کنیم ؟ حال فرض کنید این درخواست‌ها در ثانیه 5000 تا و کاربران ما 5 میلیون تا باشن ... به نظرم این سیستم قبل از پیاده سازی شکستش قطعیه ...

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۱۰/۱۰ ۱۳:۵۹

منظورتون Cache کوکی هست. اگر نقش‌های کاربر تغییر کنه، اِعمال بلافاصله‌ی تغییرات در حین گشت و گذار وی در سایت به چه شکل هست؟

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۱/۱۰/۱۰ ۱۴:۳۲

Cache کوکی برای نگهداری نقش‌های کاربر و Cache سمت سرور برای نقش‌های مشخص شده هر اکشن ...

برای موردی هم که اشاره کردید راه‌های مختلفی میتونه وجود داشته باشه . مثلا شما یک نقش جدید رو برای یک کاربر مشخص کردید . یک کلاس تعریف میکنیم که نگهدارنده‌ی لیست شناسه کاربرانی هست که نقششون در **زمانی که انلاین هستن** تغییر کرده . بعد از اعمال نقش به این کاربران این لیست بروز رسانی و در کش سمت سرور ذخیره سازی میشه . توجه داشته باشید که این لیست فقط زمانی بروز میشه که نقش کاربری تغییر کرده که **انلاین** هست. بعد در زمان درخواست یک اکشن توسط یک کاربر، اون لیست که در کش هست و واکشیش با سرعت بسیار بالا انجام میشه بررسی میشه و اگه نام اون کاربر در اون لیست بود مجددا میتونیم لیست نقش‌های اون کاربر رو از بانک استخراج و کوکی اون رو به روز رسانی کنیم . و در اخر هم نام اون از لیست حذف و کش نگهدارنده‌ی لیست به روز میشه ....

این روش میتونه روش خوبی باشه به این دلیل که مگه ممکنه نقش چند کاربر در زمانی که انلاین هستن تغییر کنه ؟ به طور حتم این تعداد خیلی کم هستن پس کلاس نگهدارنده‌ی شناسه‌ی کاربران مربوطه دارای حجم بسیار کمی هست و فظای خیلی کمی رو از حافظه سرور میگیره ولی در عوض سیستم یک سیستم جاندار و منعطف میشه ... در آخر ذکر این نکته که این روش صرفا یک ایده بود که در زمان نوشتن این متن تحلیل شد و با زمان گذاشتن روی اون و ایده‌های مشابه میشه به نتایج عالی رسید . سیستم من از همچین مکانیزمی به صورت خیلی کاملتر و با جزئیات بالا استفاده میکنه

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۱۰/۱۰ ۱۶:۵۳

مرسی از توضیحاتت.

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۱۰/۱۰ ۱۹:۵۹

گاهی اوقات مشکل از اینجا است که ما به صرف شنیده‌ها و پس زمینه‌های فکری، پیچیده‌ترین روش‌ها رو پیاده سازی می‌کنیم تا

اون پس زمینه‌ی فکری رو سرکوب کنیم. پایگاه‌های داده‌ی امروزی بسیار قدرتمند و پخته هستند و اگر قرار هست با یک Connection اضافه کل سیستم زیر سوال بره، بهتره بساط اون پایگاه داده جمع بشه.
من فکر می‌کنم باید حالت‌های دیگه ای رو هم بررسی کنی. مثلاً اینکه تغییر در نقش کاربر بوده یا مجوزهای نقش. اگر کاربر در حین پر کردن یک فرم AJAXیی مجوزش گرفته شد چه واکنشی باید رخ بده؟ آیا تمامی این بررسی‌ها و پیاده سازی اونها، به ایجاد یک ارتباط ساده و واکشی مجوز جاری کاربر برای درخواست برتری دارند؟
اگر امکانش رو داری، یک بررسی زمانی بین اون سیستم مدیریت Cache (با در نظر گرفتن تمامی حالت ها) و ایجاد ارتباط ساده با پایگاه داده برای بررسی داشتن مجوز کاربر انجام بده.

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۱/۱۰/۱۰ ۲۱:۴۲

اقای راد من هیچوقت قصد شعار دادن بر مبنای تحلیلات ذهنیم رو نداشتم و نخواهم داشت .
تمام حالت هایی که تا الان اشاره کردید قابل پوشش هست و در سیستم من در نظر گرفته شده و همچنان مورد تست قرار میگیره و ضعف هاش برطرف میشه . پیاده سازی یک بار انجام میشه ولی استفاده از اون بعد از پیاده سازی در هر کدوم از پروژه هایی که لازم باشه به سادگی ممکن میشه (به نظرم برنامه نویسی شیء گرا زیباست) . اگر توسعه دهندگان همون پایگاه‌های داده‌ی قدرتمند که ذکر کردید همچین طرز فکری داشتن که : (آیا تمامی این بررسی‌ها و پیاده سازی اونها، به ایجاد یک ارتباط ساده و واکشی مجوز جاری کاربر برای درخواست برتری دارند؟) ، هیچوقت قدرتمند و پخته نمیشدند ... این موضوع بر هیچکس پوشیده نیست که دسترسی مکرر به پایگاه داده برای برنامه هایی که ارزشمند هستند و سرعت و انعطاف در اونها مهمه یک نقطه‌ی ضعف هست . شاید همینجور طرز فکر باشه که باعث شده سیستم هایی که در بعضی از اماکن و سازمان هایی دولتی ما استفاده میشه بعد از جا افتادن و زیاد شدن داده‌ها واقعا کار کردن باهاشون عذاب آور و کسل کننده باشه...

انشاالله در چندین ماه اینده بعد از کامل شدن سیستم و انتشار اون میتونید بررسیش کنید اگر مایل بودید. کیت‌های توسعه‌ی اون هم به صورت سورس باز منتشر خواهد شد و در همین سایت هم معرفی میشن. به نظر من ارزشش رو داره که روی جزئیات این سیستم کار کنم و امیدوارم بتونه جای تامل داشته باشه . در اخر من رو ببخشید برادر من هیچوقت قصد جسارت نداشتم به شما و دیگر دوستان . این سایت محیط مقدسی هست چون واقعا مطالب خوبی در اون قرار میگیره امیدوارم بتونم از اطلاعات شما و دیگر دوستان استفاده کنم . یا حق

نویسنده: ابوالفضل
تاریخ: ۱۳۹۱/۱۰/۱۲ ۱۳:۴

سلام
من از سیستم معمولی وب فرم برای تشخیص هویت در mvc 4 استفاده کردم. اما به یه مشکل جالب برخوردم.
اپلیکیشین وب من روی شبکه داخلی کار میکنه و هر کس با نام کاربری خودش وارد میشه و کار میکنه.
بعد از لوگین ریدایرکت میکنم و در صفحه مقصد در هنگام User.Identity.IsAuthenticated همیشه false برمیگردونه و نمی‌تونم وارد شم. با هیچکدوم از مرورگرها وارد نمیشه ولی با **فایرفاکس** میتونم وارد شم!
من بدین صورت کد لوگین رو نوشتم:

```
if (Membership.ValidateUser(UserName, Password))
            {
                FormsAuthentication.RedirectFromLoginPage(UserName, true);
                return true;
            }
```

دوستان نظرشون چیه؟ مشکل از چی می‌تونه باشه؟ فایرفاکس چه خاصیتی داره که میتونه لوگین کنه؟ اما با IE و Chrome و safari امتحان کردم نشد.
در وب کانفیگ هم اینجوری تنظیم کردم

```
<authentication mode="Forms" >
      <forms loginUrl="~/Account/Login"
```

```
                protection="All"
                timeout="30"
                name="myAppCookie"
                path="/"
                requireSSL="false"
                slidingExpiration="true"
                cookieless="UseCookies"
                enableCrossAppRedirects="false" />
    </authentication>

    <membership >
      <providers>
        <clear />
        <add name="AspNetSqlMembershipProvider" type="System.Web.Security.SqlMembershipProvider"
            connectionStringName="ApplicationServices"  enablePasswordRetrieval="false"
            enablePasswordReset="true" requiresQuestionAndAnswer="false"
            requiresUniqueEmail="false" maxInvalidPasswordAttempts="5" minRequiredPasswordLength="6"
            minRequiredNonalphanumericCharacters="0" passwordAttemptWindow="10" applicationName="myApp
" />
      </providers>
    </membership>

    <profile>
      <providers>
        <clear />
        <add name="AspNetSqlProfileProvider" type="System.Web.Profile.SqlProfileProvider"
                  connectionStringName="ApplicationServices" applicationName="myApp " />
      </providers>
    </profile>

    <roleManager enabled="true">
      <providers>
        <clear />
        <add name="AspNetSqlRoleProvider" type="System.Web.Security.SqlRoleProvider"
                connectionStringName="ApplicationServices" applicationName="myApp "/>

        <add name="AspNetWindowsTokenRoleProvider"
             type="System.Web.Security.WindowsTokenRoleProvider" applicationName="myApp "  />
      </providers>
    </roleManager>
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۱۲ ۱۳:۲۶

- اولین بررسی: مراجعه کنید به قسمت 18 سری MVC به متد public ActionResult LogOn ، یک FormsAuthentication.SetAuthCookie هم باید پیش از Redirect اضافه شود.

نویسنده: ابوالفضل
تاریخ: ۱۳۹۱/۱۰/۱۲ ۱۴:۴۶

با تشکر آقای نصیری. حل شد. مشکل همون بود. کلی سرچ کرده بودم و اون رو هم امتحان کرده بودم، منتها در هنگام کپی روی سرور نهایی برای تست، فایل خروجی تغییر یافته رو اشتباهی کپی می‌کردم. الان فهمیدم قضیه این بوده! :)
ولی باز هم برام سوال هست که چرا فایرفاکس لوگین میکرد ولی بقیه نه؟!

نویسنده: مهتدی حسن‌پور
تاریخ: ۱۳۹۱/۱۱/۱۴ ۱۲:۲۵

با سلام جناب راد
امکانش هست کمی بیشتر درمورد روش‌تون توضیح بدید؟ اگه لینکی هم در این رابطه دارید ممنون می‌شم در اختیارم بذارید.

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۲۴ ۱۰:۱۹

با سلام
این دستور رو من متوجه نشدم میشه یه توضیح مختصری بدید

```
StringComparison.CurrentCultureIgnoreCase
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۴ ۱۱:۱۸

- در مورد تاثیر Culture به مطلب « از متد DateTime.ToString بدون پارامتر استفاده نکنید! » مراجعه کنید.
- IgnoreCase سبب میشه مقایسه انجام شده، حساس به حروف بزرگ و کوچک نباشد.

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۱/۱۲/۱۸ ۲۰:۲

ممنون بابت مطلب مفیدی که در اختیار ما قرار دادید.
گفتید که مجوزها بعد از یک بار واکشی در کوکی ذخیره میشه، این به خاطر وجود مقدار true برای خصوصیت cacheRolesInCookie هست درسته؟ اما من یک RoleProvider سفارشی تهیه کردم که به صورت زیر اونو به وب کانفیگ اضافه کردم:

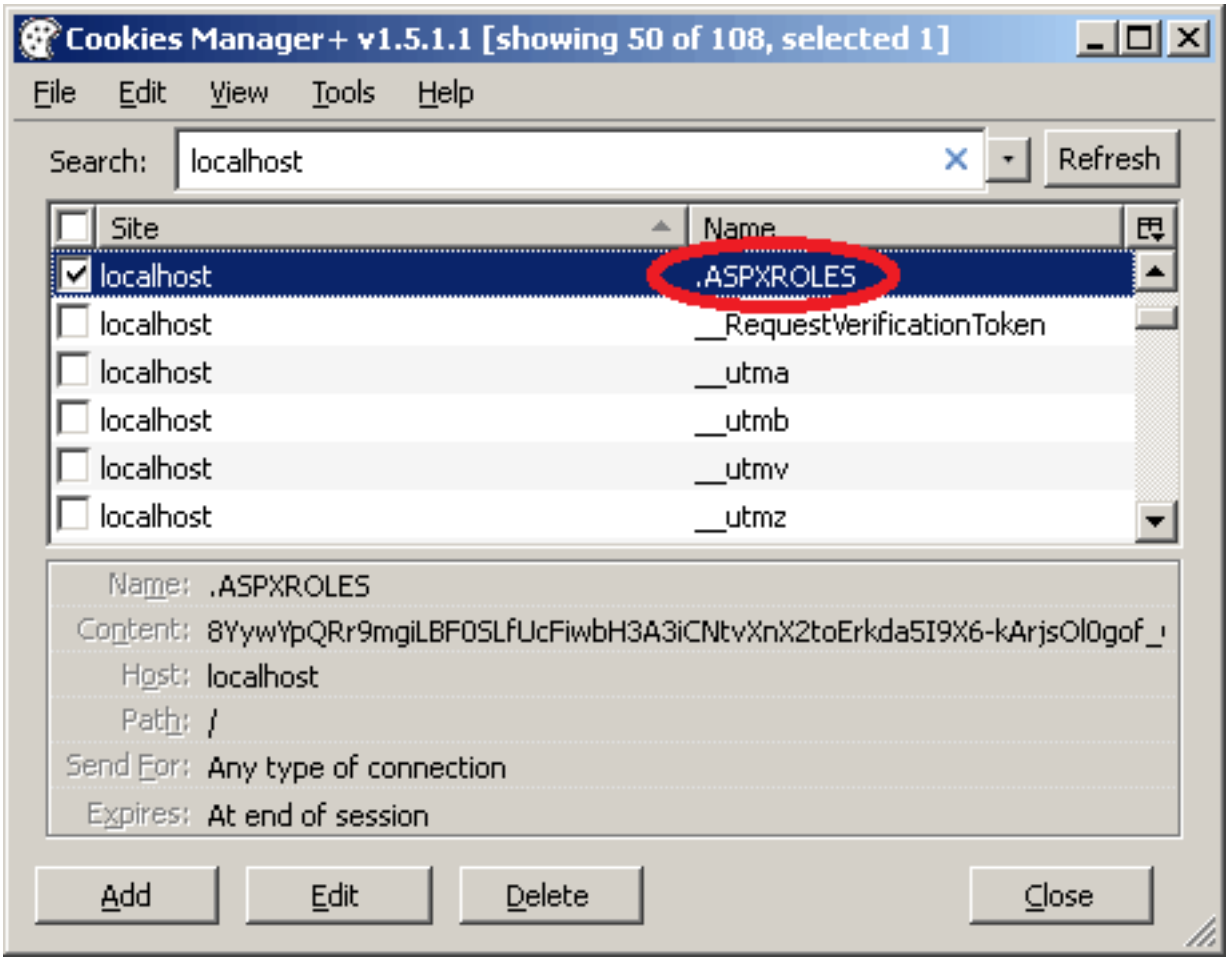
```
<roleManager cacheRolesInCookie="true" enabled="true" defaultProvider="CustomRoleProvider">
      <providers>
        <clear />
        <add name="CustomRoleProvider"
             type="MvcApp.UserInterface.Models.CustomRoleProvider"
             connectionStringName="MvcAppDb"
             applicationName="/" />
      </providers>

</roleManager>
```

اما با مراجعه به کوکی‌های مروگر چیزی نمی‌بینم، در ضمن یک Break Point هم سر متد GetRolesForUser گذاشتم که هر وقت قصد دیدن یک اکشن برای یک Role رو مشخص دارم این متد اجرا میشه یعنی از کوکی نقش رو نمی‌خونه.
سوال بعدیم اینه که این نقش چطوری توی مرورگر کاربر ذخیره میشه؟ آیا رمز گزاری شده هست یا نه به صورت Plain Text ذخیره میشه؟ اگه رمزگزاری میشه با چه الگوریتمی این کار انجام میشه و آیا میشه اونو Decrypt کرد؟
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۱۸ ۲۱:۴۷

- این کوکی رو اگر خواستید مشاهده کنید از افزونه Cookies manager استفاده کنید. چنین نام و محتوای رمزنگاری شده‌ای داره:



البته این نام پیش‌فرض است. اگر نیاز به تعیین نام دیگری بود به این صورت می‌شود عمل کرد:

```
<roleManager
    enabled="true"
    cacheRolesInCookie="true"
    defaultProvider="..."
    cookieName=".ASPXROLES"
    cookiePath="/"
    cookieTimeout="30"
    cookieRequireSSL="false"
    cookieSlidingExpiration="true"
    createPersistentCookie="false"
    cookieProtection="All">
    <providers>
      <!-- .... -->
    </providers>
</roleManager>
```

- این کوکی فقط پس از اولین فراخوانی متدهای IsInRole یا GetRoles تولید می‌شود و نه پیش از آن.
- اگر از دات نت 4 و نیم استفاده می‌کنید، برای حالت کش نشدن این نقش‌ها اخیرا یک patch ارائه شده : (^). مورد چهارم آن.

```
Assume that you set the value of the cachedRolesInCookie property to true in your web application.
Your application serializes the RolePrincipal object into the cookie, and then sends it in response.
In this situation, the role cookie value is empty in the application's following request.
```

نویسنده: سعید
تاریخ: ۱۳۹۲/۰۱/۰۶ ۱:۴

سلام
اگر کاربر بتونه متن Role رو ببینه و همچنین تغییرش بده... آیا این مشکل پیش نمیاد که شخصی بیاد و متن کوکی مدیریت رو

برداره و زمانی که لوگین کرد متن کوکی خودش رو تغییر بده. اینجوری چون همه رول‌ها از کوکی خونده میشه پس می‌تونه به این شیوه به رول‌ها کلک بزنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۰۶ ۱:۶

این اطلاعات رمزنگاری شده هستند. کلید آن‌ها هم در سرور قرار دارد و به کلاینت ارسال نمی‌شود.

نویسنده: ali
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۰:۵۹

دوست عزیز چند ماه گذشته امکانش هست پروژه ای که برای سطح دسترسی نوشتید به صورت سورس باز بگذارید؟
اگر هم به صورت مطلب جدید بگذارید خیلی ممنون میشم.تا بتونیم بازخوردهای خوبی داشته باشیم.

ممنون

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۲/۰۲/۲۹ ۲۰:۶

این پروژه صرفا واسه سطح دسترسی نبود یک بخش از CMF شخصی بنده بود با نام نئوکس، فعلا قصد انتشار عمومی اون رو ندارم چون کارای شخصی خودم رو در حال حاضر دارم با اون انجام میدم و دلیل دیگه اینکه وقت پشتیبانی ازش رو در یک محیط عمومی بعد از انتشار مثلا در Gtihub فعلا متاسفانه ندارم.
فعلا بازده عالی ای داشته این CMF از همه جهات ، اگر تصمیم به انتشار عمومیش گرفتم حتما داخل همین سایت اطلاع رسانی میشه .
برای مثال یکی از کارایی که باهاش انجام شده سایت شرکت خود بنده هست ، اینجا
یا سایت شرکت رایان صنعت (سهامی خاص) با رتبه‌ی یک در چند کلمه‌ی کلیدی به لطف SEO Engine این CMF در اینجا

نویسنده: شاهین
تاریخ: ۱۳۹۲/۰۳/۱۲ ۲۳:۵

سلام
من وقتی CustomRoleProvider رو از RoleProvider ارث بری میکنم، به جز دو تابع GetRolesForUser و IsUserInRole که override کردیم واسه بقیه توابع کلاس RoleProvider خطای زیر رو میده:

```
Error2'student_mvc.Security.CustomRoleProvider' does not implement inherited abstract member
'System.Web.Security.RoleProvider.GetAllRoles()
```

یا مثلا

```
Error3'student_mvc.Security.CustomRoleProvider' does not implement inherited abstract member
'System.Web.Security.RoleProvider.GetUsersInRole(string)
```

مشکل و راه حل چیست.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۲ ۲۳:۱۳

اون‌ها رو باید دستی اضافه کنی. یک نمونه‌اش در قسمت 18 سری MVC مطرح شده.

نویسنده: وحید

تاریخ: ۱۳۹۲/۱۰/۱۶ ۱۳:۵۴

بنظر شما اگر بجای احراز هویت پیش فرض در MVC یا همان FormAuthentication پس از لاگین کاربر اطلاعات موجودیت آن را در session ای بریزیم و بعد در جاهای مختلف برنامه از آن استفاده کنیم مثلا هر جا که لازم بود type کاربر را بدست آورد که مثلا مشتری یا نماینده یا مدیر است دیگر لازم نیست متد GetUserById را صدازد و بار بر روی سیستم را افزایش داد آیا روش بنده صحیح است از لحاظ performance و امنیت؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۶ ۱۴:۳۴

اتفاقا سربار سشن بیشتر است از Forms Authentication. اطلاعات سشن به صورت پیش فرض در حافظه سرور ذخیره می‌شود اما اطلاعات تیکت Forms Authentication در یک کوکی رمزنگاری شده در مرورگر کاربر ذخیره خواهد شد. در این حالت مصرف حافظه کمتری را در سمت سرور خواهید داشت و ضمنا با ری‌سایکل شدن برنامه در IIS، تمام لاگین‌های کاربران از دست نخواهد رفت (اصطلاحا تمام سشن‌های لاگین نمی‌پرند) و مجبور به لاگین مجدد نخواهند شد؛ چون کوکی رمزنگاری شده جهت اعتبارسنجی بعدی، در مرورگر کاربر ذخیره شده و نه در حافظه سرور.

نویسنده: وحید
تاریخ: ۱۳۹۲/۱۰/۱۶ ۱۷:۰

باتشکر از شما .
پس شما میگویید هر زمان که ما خواستیم ببینیم کاربری نمایندست یا مدیر در اکشنی که تعداد آن اکشن‌ها هم کم نیست می‌بایست GetUserById را استفاده نمود؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۶ ۱۷:۲۵

دو مطلب هست. یکی اینکه اگر نیاز به ID کاربر داشتید، این مورد همیشه در User.Identity.Name موجود هست و زمانیکه در حین لاگین انتساب داده شد، در کوکی رمزنگاری شده آن قرار می‌گیره. یعنی واکشی از دیتابیس نداره. ضمنا RoleProvider استفاده شده هم زمانیکه در وب کانفیگ تنظیم میشه، مقدار cacherolesincookie=true داره (در مقاله هست). یعنی این هم کش میشه و هربار از دیتابیس واکشی نمیشه.

نویسنده: وحید
تاریخ: ۱۳۹۲/۱۰/۱۶ ۲۳:۲۷

ممنون از شما آیا با cacherolesincookie=true فقط roleها کش میشوند ؟ مثلا زمانی که کاربری لاگین کرده و در هدر پیغام خوش آمدید کاربر گرامی آقای...می شود این نام و نام خانوادگی را باید هر دفعه بازای هر request از دیتابیس بیاریم یا میتونیم از حالت شبیه این cacherolesincookie=true استفاده کرد با تشکر

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۶ ۲۳:۳۸

برای این حالت‌های خاص می‌تونید از CacheManager قسمت 19 سری MVC استفاده کنید. برای بررسی حالت Forms Authentication هم قسمت 18 سری MVC به مباحث مقدماتی اون پرداخته.

نویسنده: رضا گرمارودی
تاریخ: ۱۳۹۲/۱۱/۲۰ ۹:۲۲

سلام
من کل مطالب Mvc 18 و بخش کوکی‌ها را مطالعه کردم اما به دلیل نظرات مختلف متوجه نشدم راه اصولی چیه؟
برای هر کاربر یکسری اطلاعات وجود دارد که میشه در کوکی‌ها ذخیره کرد مثل نام کاربری یا هر چیز دیگه ای که حتی اگر کاربر

آنها را تغییر بده مهم نیستند و صرفا جنبه نمایش در صفحات را دارند اما یکسری اطلاعات هست که خیلی مهمه هستند مثل این که این کاربر مدیر هست یا خیر. این اطلاعات یا باید هر بار که نیاز هست از دیتابیس یا هر منبع دیگه ای واکشی بشه ویا در جایی ذخیره بشه که هر وقت خواستیم به اون دسترسی داشته باشیم.
پیش‌تر این نوع اطلاعات و در Session ذخیره می‌کردیم که از دید کاربر به دور بود ، اما برای پردازش موازی و انجام چک لیست تهیه شده بهتره که از Sessionها استفاده نشه،
خب حالا ذخیره این اطلاعات در کوکی‌ها درسته؟ حتی اگر کد بشن باز خطرناکه ، چرا که یک کاربر اگر بتونه کوکی که مدیر بودن یا نبودن کاربر را مشخص می‌کند را تغییر بده می‌تونه به همه بخش‌ها دسترسی پیدا کنه!
از طرفی صرفا مدیر بودن یا نبودن یک کاربر مطرح نیست ، اطلاعات زیادی هستند که مهم هستند و ذخیره اون‌ها در Session میتونه منابع سرور و به خودش مشغول کنه!
روش درست چیه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۰ ۹:۲۸

- راه حل‌های مبتنی بر سشن، از Classics ASP دهه نود به ارث رسیده‌اند. عملا با پیشرفت‌هایی که حاصل شده نیازی به بسیاری از آن‌ها نیست. مصرف حافظه بالایی دارند و همچنین با ری‌استارت شدن برنامه در سرور، تمام سشن‌ها از بین خواهند رفت. این مشکلات در Forms Authentication وجود ندارند.
- قدمت Forms Authentication به ASP.NET *1.x* بر می‌گردد. می‌توانید در این مورد در سایت‌های دیگر نیز بیشتر تحقیق کنید که آیا مشکل حادی از سال 2001 تا الان گزارش شده یا خیر.
- کلید رمزنگاری این کوکی‌ها در سمت سرور قرار دارد و تنها یک راه برای دسترسی به آن‌ها هست؛ دسترسی به سرور. در این حالت عملا کل سیستم مورد حمله قرار گرفته و یک کوکی شاید اهمیت خاصی نداشته باشد.
- ضمنا طول مدت زمان معتبر بودن اطلاعات Forms Authentication و دائمی بودن و نبودن کوکی‌های آن قابل تنظیم است (بحث شده در مطلب فوق).

نویسنده: رضا گرمارودی
تاریخ: ۱۳۹۲/۱۱/۲۰ ۱۰:۴۵

ممنون از پاسختون اما Authenticate در مند SetAuthCookie تنها نام یوزر و ذخیره می‌کند، برای ذخیره سایر اطلاعات چه کار باید کرد؟

اگر باید از متدهای HttpContext استفاده بشود جهت رمزنگاری کوکی‌ها چه الگوریتمی به اندازه Authentication مطمئن است؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۰ ۱۱:۲۸

- سایر اطلاعات مانند نقش‌های یک کاربر (مدیر است یا نویسنده مثلا) در مطلب جاری به صورت یک role provider مکمل پیاده سازی شده‌است.
+ بیشتر از این نیازی نیست اطلاعاتی را در کوکی‌ها یا جای دیگری ذخیره کنید. اطلاعاتی فراتر از این (مانند صفحه پروفایل یک شخص در سایت)، بسیار شخصی بوده و هر زمانیکه نیاز باشد، باید مستقلا از دیتابیس واکشی شوند.

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۲/۱۱/۲۰ ۱۳:۴۳

بنده در حال توسعه‌ی یه سیستم ساده Membership متن باز هستم که توانایی تعریف گروه و تعیین سطوح دسترسی به صورت پویا را دارد. کار خیلی زیادی ازش باقی نمانده و در صورت اتمام حتما به اشتراک می‌گذارم.

نویسنده: رضا گرمارودی
تاریخ: ۱۳۹۲/۱۱/۲۳ ۱۲:۱۸

سلام؛ من هم مشکل ایشون دارم و رول‌ها کش نمیشه و هر بار متد GetRolesForUser اجرا میشود. اون هم نه یکبار بلکه به تعداد زیاد و بار زیادی به دیتابیس وارد می‌کنه. لینک شما هم پیغام زیر و میدهد:

An update is available for the .NET Framework 4.5 in Windows 7 SP1, Windows Server 2008 R2 SP1, Windows Server 2008 SP2, and Windows Vista SP2: January 2013

اما ویندوز من 8 هست و پیغام سایت برای ویندوزهای 2008و7وvista هستش. با گشتن هم دیدم بعضی‌ها خودشون متد را دستی کش کردن

```
public override string[] GetRolesForUser(string username)
        {
            var cacheKey = string.Format("{0}:{1}", username, ApplicationName);
            var cache = HttpContext.Current.Cache;
            var roles = cache[cacheKey] as string[];
            if (null == roles)
            {
                using (var db = new Uas3Context())
                {
                    var u = new EfStudentService(db);
                    roles=u.GetUserRoles(username);
                    cache.Insert(cacheKey, roles, null, Cache.NoAbsoluteExpiration,
Cache.NoSlidingExpiration);
                }
            }
            return roles;
        }
```

اما این متد و من کش کردم باقی متدها چی؟ کل متدها را همین طوری کش کنم؟ این کد هم به نظرم ناقصه . چون با تغییر دیتابیس به روز نمیشه و صرفا درصورت وجود کش اطلاعات می‌خونه. اصلا چرا در دات نت 4.5 این مشکل هست و چرا بر روی ویندوز 8 این پیغام داده میشود؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۳ ۱۲:۴۴

- مشکل طولانی بودن حاصل BinaryFormatter serialization برطرف شده (نزدیک به *یکسال قبل* در January 2013). این مشکل سبب می‌شده تا حاصل RolePrincipal.ToEncryptedTicket بسیار طولانی شده و بیشتر از حد مجاز اندازه قابل ذخیره سازی در یک کوکی شود.
- نسخه‌ی ویندوز 8 و ویندوز سرور 2012 آن از اینجا قابل دریافت است؛ نسخه‌ی ویندوز 7 و ویندوز سرور 2008 از اینجا .
+ آپدیت ویندوز را روشن کنید تا آخرین به روز رسانی‌ها و نگارش‌های دات نت نصب شده را به صورت خودکار دریافت کنید.
+ بررسی کنید آیا ویژگی cacheRolesInCookie=true rolemanager در وب کانفیگ تنظیم شده یا خیر. اگر خیر، چیزی کش نخواهد شد.

نویسنده: ارکا
تاریخ: ۱۳۹۲/۱۲/۱۵ ۲۱:۳۷

سلام،
الان دو، سه روزه که میخوام شخصی سازی Membership رو یاد بگیرم، اما واقعا گیج شدم،
میدونم که باید از کلاس‌های MembershipUser,Membership Provider و RoleProvider ارث بری بکنم و توابع مورد نظرم رو Override، اما اینکه دیتابیس چجوری باید باشه، توابعی که نوع خروجیش (تعداد و نوع پارامترهاش) اونی نیس که لازم دارم، چجوری باید پیاده سازی کنم.

امکان راهنمایی وجود داره؟
- من Web Form کار میکنم.
ممنون

نویسنده: محمد شهریاری
تاریخ: ۱۳۹۲/۱۲/۱۶ ۸:۳۰

مبحث مربوط به مدیریت کاربران و گروه کاربران و پروفایل از طریق سه کلاس پایه System.Web.Security.MembershipProvider
System.Web.Security.RoleProvider
System.Web.Profile.ProfileProvider
همونطور که از کلاس‌ها مشخص هست به صورت Provider مدیریت می‌شود. این به این معنی است که اگر شما بخواید Provider خودتون با هر Datasource جدید یا منطق جدید داشته باشید مجاز هستید نسخه جدید از این Providerها رو پیاده سازی و با تنظیمات مربوطه در Web.Config استفاده کنید . در این قسمت شما مجاز به تغییر ورودی یا خروجی متدهای abstract وclassهای گفته شده نیستید .

پیاده سازی‌های مختلفی از این سه کلاس به صورت ODBC , EF , .. وجود داره که لینک زیر مربوط به پیاده سازی odbc است .
پیاده سازی MembershipProvider
پیاده سازی RoleProvider
پیاده سازی ProfileProvider

موفق باشید

نویسنده: ارکا
تاریخ: ۱۳۹۲/۱۲/۱۶ ۹:۲۸

ممنون بابت راهنمایی،
تقریبا تو پیاده سازی مجدد مشکلی ندارم، بیشتر مشکلم با امضای توابع هستش (نوع خروجی و پارامترهاش).

یعنی راهی نیست که بتونم توابعی با نوع خروجی و پارامترهای دلخواه داشته باشم و در عین حال از سیستم ممبرشیب هم استفاده کنم؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۲/۱۶ ۱۰:۵۳

وجود اینترفیس و قرارداد در کدها برای این است که هم طراح و هم استفاده کننده تکلیف خودشان را بدانند. اگر قرار باشد این اینترفیس‌ها به میل شما هر روز تغییر کنند که دیگر به آن قرارداد گفته نمی‌شود. ضمنا برای پاسخگویی به این نوع سؤالات تمامی ناپذیر، سیستم جدیدی رو طراحی کردند به نام ASP.NET Identity . این سیستم از بنیان سورس باز هست. در اینجا شما هر طور که دوست داشتید، تمام اینترفیس‌ها و کدها رو تغییر بدید. سورس رو که دارید. وابسته هم نیست به بانک اطلاعاتی خاصی.

نویسنده: امیر ارسلان
تاریخ: ۱۳۹۳/۰۵/۱۰ ۱۲:۴۲

ممنون از تلاش و سایت خوبتون. اما کاش یه نمونه میزاشتین من خیلی تلاش کردم اما نتونستم جواب درستی بگیرم. اگه امکان داره یه نمونه بزازین ممنون میشم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۱۰ ۱۲:۴۶

- از پروژه سورس باز « Iris Membership برای احراز هویت کاربران در ASP.NET MVC به صورت پویا » ایده بگیرید.
- همچنین در پروژه‌ی سورس باز « سیستم مدیریت محتوای IRIS » نیز پیاده سازی کاملی از این مبحث وجود دارد.

نویسنده: امیر ارسلان
تاریخ: ۱۳۹۳/۰۵/۱۰ ۲۰:۵۷

ممنون. لینک دوم من نتونستم روی لوکال نصب کنم اما Iris membership دیدم کمی درک کردم اما نمیدونم database چطور به پروژه Add کنم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۱۰ ۲۳:۱۶

پیشنیاز درک این پروژه‌ها EF Code first است.

نویسنده: فایز مهربانی
تاریخ: ۱۳۹۳/۰۷/۲۶ ۱۳:۴۲

سلام وقت بخیر...
ببخشید در مورد Claim Authorization چه مطالبی وجود داره ؟ رفرنس فارسی وجود داره ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۲۶ ۱۴:۴۴

« مروری بر Claim »