عنوان: ASP.NET MVC #18
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۳۱ ۱۰:۴۵:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

اعتبار سنجی کاربران در ASP.NET MVC

دو مکانیزم اعتبارسنجی کاربران به صورت توکار در ASP.NET MVC در دسترس هستند: Forms authentication و Windows authentication.

در حالت Forms authentication، برنامه موظف به نمایش فرم لاگین به کاربرها و سپس بررسی اطلاعات وارده توسط آن‌ها است. برخلاف آن، Windows authentication حالت یکپارچه با اعتبار سنجی ویندوز است. برای مثال زمانیکه کاربری به یک دومین ویندوزی وارد می‌شود، از همان اطلاعات ورود او به شبکه داخلی، به صورت خودکار و یکپارچه جهت استفاده از برنامه کمک گرفته خواهد شد و بیشترین کاربرد آن در برنامه‌های نوشته شده برای اینترانت‌های داخلی شرکت‌ها است. به این ترتیب کاربران یک بار به دومین وارد شده و سپس برای استفاده از برنامه‌های مختلف ASP.NET، نیازی به ارائه نام کاربری و کلمه عبور نخواهند داشت. Forms authentication بیشتر برای برنامه‌هایی که از طریق اینترنت به صورت عمومی و از طریق انواع و اقسام سیستم عامل‌ها قابل دسترسی هستند، توصیه می‌شود (و البته منعی هم برای استفاده در حالت اینترانت ندارد).

ضمنا باید به معنای این دو کلمه هم دقت داشت: هدف از Authentication این است که مشخص گردد هم اکنون چه کاربری به سایت وارد شده است. Authorization، سطح دسترسی کاربر وارد شده به سیستم و اعمالی را که مجاز است انجام دهد، مشخص می‌کند.

فیلتر Authorize در ASP.NET MVC

یکی دیگر از فیلترهای امنیتی ASP.NET MVC به نام Authorize، کار محدود ساختن دسترسی به متدهای کنترلرها را انجام می‌دهد. زمانیکه اکشن متدی به این فیلتر یا ویژگی مزین می‌شود، به این معنا است که کاربران اعتبارسنجی نشده، امکان دسترسی به آن‌را نخواهند داشت. فیلتر Authorize همواره قبل از تمامی فیلترهای تعریف شده دیگر اجرا می‌شود.

فیلتر Authorize با پیاده سازی اینترفیس System.Web.Mvc.IAuthorizationFilter توسط کلاس System.Web.Mvc.AuthorizeAttribute در دسترس می‌باشد. این کلاس علاوه بر پیاده سازی اینترفیس یاد شده، دارای دو خاصیت مهم زیر نیز می‌باشد:

```
public string Roles { get; set; } // comma-separated list of role names
public string Users { get; set; } // comma-separated list of usernames
```

زمانیکه فیلتر Authorize به تنهایی بکارگرفته می‌شود، هر کاربر اعتبار سنجی شده‌ای در سیستم قادر خواهد بود به اکشن متد مورد نظر دسترسی پیدا کند. اما اگر همانند مثال زیر، از خواص Roles و یا Users نیز استفاده گردد، تنها کاربران اعتبار سنجی شده مشخصی قادر به دسترسی به یک کنترلر یا متدی در آن خواهند شد:

```
[Authorize(Roles="Admins")]
public class AdminController : Controller
{
    [Authorize(Users="Vahid")]
    public ActionResult DoSomethingSecure()
    {
    }
}
```

در این مثال، تنها کاربرانی با نقش Admins قادر به دسترسی به کنترلر جاری Admin خواهند بود. همچنین در بین این کاربران ویژه، تنها کاربری به نام Vahid قادر است متد DoSomethingSecure را فراخوانی و اجرا کند.

اکنون سؤال اینجا است که فیلتر Authorize چگونه از دو مکانیزم اعتبار سنجی یاد شده استفاده می‌کند؟ برای پاسخ به این سؤال، فایل web.config برنامه را باز نموده و به قسمت authentication آن دقت کنید:

```
<authentication mode="Forms">
      <forms loginUrl="~/Account/LogOn" timeout="2880" />
</authentication>
```

به صورت پیش فرض، برنامه‌های ایجاد شده توسط VS.NET جهت استفاده از حالت Forms یا همان Forms authentication تنظیم شده‌اند. در اینجا کلیه کاربران اعتبار سنجی نشده، به کنترلری به نام Account و متد LogOn در آن هدایت می‌شوند.
برای تغییر آن به حالت اعتبار سنجی یکپارچه با ویندوز، فقط کافی است مقدار mode را به Windows تغییر داد و تنظیمات forms آن‌را نیز حذف کرد.

یک نکته: اعمال تنظیمات اعتبار سنجی اجباری به تمام صفحات سایت

تنظیم زیر نیز در فایل وب کانفیگ برنامه، همان کار افزودن ویژگی Authorize را انجام می‌دهد با این تفاوت که تمام صفحات سایت را به صورت خودکار تحت پوشش قرار خواهد داد (البته منهای loginUrl ایی که در تنظیمات فوق مشاهده نمودید):

```
<authorization>
     <deny users="?" />
</authorization>
```

در این حالت دسترسی به تمام آدرس‌های سایت تحت تاثیر قرار می‌گیرند، منجمله دسترسی به تصاویر و فایل‌های CSS و غیره.
برای اینکه این موارد را برای مثال در حین نمایش صفحه لاگین نیز نمایش دهیم، باید تنظیم زیر را پیش از تگ system.web به فایل وب کانفیگ برنامه اضافه کرد:

```
<!-- we don't want to stop anyone seeing the css and images -->
<location path="Content">
    <system.web>
      <authorization>
        <allow users="*" />
      </authorization>
    </system.web>
</location>
```

در اینجا پوشه Content از سیستم اعتبارسنجی اجباری خارج می‌شود و تمام کاربران به آن دسترسی خواهند داشت.
به علاوه امکان امن ساختن تنها قسمتی از سایت نیز میسر است؛ برای مثال:

```
<location path="secure">
  <system.web>
    <authorization>
      <allow roles="Administrators" />
      <deny users="*" />
    </authorization>
  </system.web>
</location>
```

در اینجا مسیری به نام secure، نیاز به اعتبارسنجی اجباری دارد. به علاوه تنها کاربرانی در نقش Administrators به آن دسترسی خواهند داشت.

نکته: به تنظیمات انجام شده در فایل Web.Config دقت داشته باشید

همانطور که می‌شود دسترسی به یک مسیر را توسط تگ location بازگذاشت، امکان بستن آن هم فراهم است (بجای allow از deny استفاده شود). همچنین در ASP.NET MVC به سادگی می‌توان تنظیمات مسیریابی را در فایل global.asax.cs تغییر داد. برای مثال اینبار مسیر دسترسی به صفحات امن سایت، Admin خواهد بود نه Secure. در این حالت چون از فیلتر Authorize استفاده نشده و همچنین فایل web.config نیز تغییر نکرده، این صفحات بدون محافظت رها خواهند شد.

بنابراین اگر از تگ location برای امن سازی قسمتی از سایت استفاده می‌کنید، حتما باید پس از تغییرات مسیریابی، فایل web.config را هم به روز کرد تا به مسیر جدید اشاره کند.

به همین جهت در ASP.NET MVC بهتر است که صریحا از فیلتر Authorize بر روی کنترلرها (جهت اعمال به تمام متدهای آن) یا بر روی متدهای خاصی از کنترلرها استفاده کرد.

امکان تعریف AuthorizeAttribute در فایل global.asax.cs و متد RegisterGlobalFilters آن به صورت سراسری نیز وجود دارد. اما در این حالت حتی صفحه لاگین سایت هم دیگر در دسترس نخواهد بود. برای رفع این مشکل در ASP.NET MVC 4 فیلتر دیگری به نام AllowAnonymousAttribute معرفی شده است تا بتوان قسمت‌هایی از سایت را مانند صفحه لاگین، از سیستم اعتبارسنجی اجباری خارج کرد تا حداقل کاربر بتواند نام کاربری و کلمه عبور خودش را وارد نماید:

```
[System.Web.Mvc.AllowAnonymous]
public ActionResult Login()
{
    return View();
}
```

بنابراین در ASP.NET MVC 4.0، فیلتر AuthorizeAttribute را سراسری تعریف کنید. سپس در کنترلر لاگین برنامه از فیلتر AllowAnonymous استفاده نمائید.

البته نوشتن فیلتر سفارشی AllowAnonymousAttribute در ASP.NET MVC 3.0 نیز میسر است. برای مثال:

```
public class LogonAuthorize : AuthorizeAttribute {
     public override void OnAuthorization(AuthorizationContext filterContext) {
         if (!(filterContext.Controller is AccountController))
             base.OnAuthorization(filterContext);
     }
}
```

در این فیلتر سفارشی، اگر کنترلر جاری از نوع AccountController باشد، از سیستم اعتبار سنجی اجباری خارج خواهد شد. مابقی کنترلرها همانند سابق پردازش می‌شوند. به این معنا که اکنون می‌توان LogonAuthorize را به صورت یک فیلتر سراسری در فایل global.asax.cs معرفی کرد تا به تمام کنترلرها، منهای کنترلر Account اعمال شود.

مثالی جهت بررسی حالت Windows Authentication

یک پروژه جدید خالی ASP.NET MVC را آغاز کنید. سپس یک کنترلر جدید را به نام Home نیز به آن اضافه کنید. در ادامه متد Index آن‌را با ویژگی Authorize، مزین نمائید. همچنین بر روی نام این متد کلیک راست کرده و یک View خالی را برای آن ایجاد کنید:

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

```
namespace MvcApplication15.Controllers
{
    public class HomeController : Controller
    {
        [Authorize]
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

محتوای View متناظر با متد Index را هم به شکل زیر تغییر دهید تا نام کاربر وارد شده به سیستم را نمایش دهد:

```
@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>
Current user: @User.Identity.Name
```

به علاوه در فایل Web.config برنامه، حالت اعتبار سنجی را به ویندوز تغییر دهید:

```
<authentication mode="Windows" />
```

اکنون اگر برنامه را اجرا کنید و وب سرور آزمایشی انتخابی هم IIS Express باشد، پیغام HTTP Error 401.0 - Unauthorized نمایش داده می‌شود. علت هم اینجا است که Windows Authentication به صورت پیش فرض در این وب سرور غیرفعال است. برای فعال سازی آن به مسیر My Documents\IISExpress\config مراجعه کرده و فایل applicationhost.config را باز نمائید. تگ windowsAuthentication را یافته و ویژگی enabled آن‌را که false است به true تنظیم نمائید. اکنون اگر برنامه را مجددا اجرا کنیم، در محل نمایش User.Identity.Name، نام کاربر وارد شده به سیستم نمایش داده خواهد شد.
همانطور که مشاهده می‌کنید در اینجا همه چیز یکپارچه است و حتی نیازی نیست صفحه لاگین خاصی را به کاربر نمایش داد. همینقدر که کاربر توانسته به سیستم ویندوزی وارد شود، بر این اساس هم می‌تواند از برنامه‌های وب موجود در شبکه استفاده کند.

بررسی حالت Forms Authentication

برای کار با Forms Authentication نیاز به محلی برای ذخیره سازی اطلاعات کاربران است. اکثر مقالات را که مطالعه کنید شما را به مباحث membership مطرح شده در زمان ASP.NET 2.0 ارجاع می‌دهند. این روش در ASP.NET MVC هم کار می‌کند؛ اما الزامی به استفاده از آن نیست.

برای بررسی حالت اعتبار سنجی مبتنی بر فرم‌ها، یک برنامه خالی ASP.NET MVC جدید را آغاز کنید. یک کنترلر Home ساده را نیز به آن اضافه نمائید.
سپس نیاز است نکته «تنظیمات اعتبار سنجی اجباری تمام صفحات سایت» را به فایل وب کانفیگ برنامه اعمال نمائید تا نیازی نباشد فیلتر Authorize را در همه جا معرفی کرد. سپس نحوه معرفی پیش فرض Forms authentication تعریف شده در فایل web.config نیز نیاز به اندکی اصلاح دارد:

```
<authentication mode="Forms">
      <!--one month ticket-->
      <forms name=".403MyApp"
             cookieless="UseCookies"
             loginUrl="~/Account/LogOn"
             defaultUrl="~/Home"
             slidingExpiration="true"
             protection="All"
             path="/"
             timeout="43200"/>
</authentication>
```

در اینجا استفاده از کوکی‌ها اجباری شده است. loginUrl به کنترلر و متد لاگین برنامه اشاره می‌کند. defaultUrl مسیری است که کاربر پس از لاگین به صورت خودکار به آن هدایت خواهد شد. همچنین نکته‌ی مهم دیگری را که باید رعایت کرد، name ایی است که در این فایل config عنوان می‌کنید. اگر بر روی یک وب سرور، چندین برنامه وب ASP.Net را در حال اجرا دارید، باید برای هر کدام از این‌ها نامی جداگانه و منحصربفرد انتخاب کنید، در غیراینصورت تداخل رخ داده و گزینه مرا به خاطر بسپار شما کار نخواهد کرد.

کار slidingExpiration که در اینجا تنظیم شده است نیز به صورت زیر می‌باشد:

اگر لاگین موفقیت آمیزی ساعت 5 عصر صورت گیرد و timeout شما به عدد 10 تنظیم شده باشد، این لاگین به صورت خودکار در 5:10 منقضی خواهد شد. اما اگر در این حین در ساعت 5:05 ، کاربر، یکی از صفحات سایت شما را مرور کند، زمان منقضی شدن کوکی ذکر شده به 5:15 تنظیم خواهد شد(مفهوم تنظیم slidingExpiration). لازم به ذکر است که اگر کاربر پیش از نصف زمان منقضی شدن کوکی (مثلا در 5:04)، یکی از صفحات را مرور کند، تغییری در این زمان نهایی منقضی شدن رخ نخواهد داد.

اگر timeout ذکر نشود، زمان منقضی شدن کوکی ماندگار (persistent) مساوی زمان جاری + زمان منقضی شدن سشن کاربر که پیش فرض آن 30 دقیقه است، خواهد بود.

سپس یک مدل را به نام Account به پوشه مدل‌های برنامه با محتوای زیر اضافه نمائید:

```
using System.ComponentModel.DataAnnotations;

namespace MvcApplication15.Models
{
    public class Account
    {
        [Required(ErrorMessage = "Username is required to login.")]
        [StringLength(20)]
        public string Username { get; set; }

        [Required(ErrorMessage = "Password is required to login.")]
        [DataType(DataType.Password)]
        public string Password { get; set; }

        public bool RememberMe { get; set; }
    }
}
```

همچنین مطابق تنظیمات اعتبار سنجی مبتنی بر فرم‌های فایل وب کانفیگ، نیاز به یک AccountController نیز هست:

```
using System.Web.Mvc;
using MvcApplication15.Models;

namespace MvcApplication15.Controllers
{
    public class AccountController : Controller
    {
        [HttpGet]
        public ActionResult LogOn()
        {
            return View();
```

```
        }

        [HttpPost]
        public ActionResult LogOn(Account loginInfo, string returnUrl)
        {
            return View();
        }
    }
}
```

در اینجا در حالت HttpGet فرم لاگین نمایش داده خواهد شد. بنابراین بر روی این متد کلیک راست کرده و گزینه Add view را انتخاب کنید. سپس در صفحه باز شده گزینه Create a strongly typed view را انتخاب کرده و مدل را هم بر روی کلاس Account قرار دهید. قالب scaffolding را هم Create انتخاب کنید. به این ترتیب فرم لاگین برنامه ساخته خواهد شد.
اگر به متد HttpPost فوق دقت کرده باشید، علاوه بر دریافت وهله‌ای از شیء Account، یک رشته را به نام returnUrl نیز تعریف کرده است. علت هم اینجا است که سیستم Forms authentication، صفحه بازگشت را به صورت خودکار به شکل یک کوئری استرینگ به انتهای Url جاری اضافه می‌کند. مثلا:

```
http://localhost/Account/LogOn?ReturnUrl=something
```

بنابراین اگر یکی از پارامترهای متد تعریف شده به نام returnUrl باشد، به صورت خودکار مقدار دهی خواهد شد.

تا اینجا زمانیکه برنامه را اجرا کنیم، ابتدا بر اساس تعاریف مسیریابی پیش فرض برنامه، آدرس کنترلر Home و متد Index آن فراخوانی می‌گردد. اما چون در وب کانفیگ برنامه authorization را فعال کرده‌ایم، برنامه به صورت خودکار به آدرس مشخص شده در loginUrl قسمت تعاریف اعتبارسنجی مبتنی بر فرم‌ها هدایت خواهد شد. یعنی آدرس کنترلر Account و متد LogOn آن درخواست می‌گردد. در این حالت صفحه لاگین نمایان خواهد شد.

مرحله بعد، اعتبار سنجی اطلاعات وارد شده کاربر است. بنابراین نیاز است کنترلر Account را به نحو زیر بازنویسی کرد:

```
using System.Web.Mvc;
using System.Web.Security;
using MvcApplication15.Models;

namespace MvcApplication15.Controllers
{
    public class AccountController : Controller
    {
        [HttpGet]
        public ActionResult LogOn(string returnUrl)
        {
            if (User.Identity.IsAuthenticated) //remember me
            {
                if (shouldRedirect(returnUrl))
                {
                    return Redirect(returnUrl);
                }
                return Redirect(FormsAuthentication.DefaultUrl);
            }

            return View(); // show the login page
        }

        [HttpGet]
        public void LogOut()
        {
            FormsAuthentication.SignOut();
        }

        private bool shouldRedirect(string returnUrl)
        {
```

```
            // it's a security check
            return !string.IsNullOrWhiteSpace(returnUrl) &&
                                Url.IsLocalUrl(returnUrl) &&
                                returnUrl.Length > 1 &&
                                returnUrl.StartsWith("/") &&
                                !returnUrl.StartsWith("//") &&
                                !returnUrl.StartsWith("/\\");
        }

        [HttpPost]
        public ActionResult LogOn(Account loginInfo, string returnUrl)
        {
            if (this.ModelState.IsValid)
            {
                if (loginInfo.Username == "Vahid" && loginInfo.Password == "123")
                {
                    FormsAuthentication.SetAuthCookie(loginInfo.Username, loginInfo.RememberMe);
                    if (shouldRedirect(returnUrl))
                    {
                        return Redirect(returnUrl);
                    }
                    FormsAuthentication.RedirectFromLoginPage(loginInfo.Username,
loginInfo.RememberMe);
                }
            }
            this.ModelState.AddModelError("", "The user name or password provided is incorrect.");
            ViewBag.Error = "Login faild! Make sure you have entered the right user name and
password!";
            return View(loginInfo);
        }
    }
}
```

در اینجا با توجه به گزینه «مرا به خاطر بسپار»، اگر کاربری پیشتر لاگین کرده و کوکی خودکار حاصل از اعتبار سنجی مبتنی بر فرم‌های او نیز معتبر باشد، مقدار User.Identity.IsAuthenticated مساوی true خواهد بود. بنابراین نیاز است در متد LogOn از نوع HttpGet به این مساله دقت داشت و کاربر اعتبار سنجی شده را به صفحه پیش‌فرض تعیین شده در فایل web.config برنامه یا returnUrl هدایت کرد.

در متد LogOn از نوع HttpPost، کار اعتبارسنجی اطلاعات ارسالی به سرور انجام می‌شود. در اینجا فرصت خواهد بود تا اطلاعات دریافتی، با بانک اطلاعاتی مقایسه شوند. اگر اطلاعات مطابقت داشتند، ابتدا کوکی خودکار FormsAuthentication تنظیم شده و سپس به کمک متد RedirectFromLoginPage کاربر را به صفحه پیش فرض سیستم هدایت می‌کنیم. یا اگر returnUrl ایی وجود داشت، آن‌را پردازش خواهیم کرد.

برای پیاده سازی خروج از سیستم هم تنها کافی است متد FormsAuthentication.SignOut فراخوانی شود تا تمام اطلاعات سشن و کوکی‌های مرتبط، به صورت خودکار حذف گردند.

تا اینجا فیلتر Authorize بدون پارامتر و همچنین در حالت مشخص سازی صریح کاربران به نحو زیر را پوشش دادیم:

```
[Authorize(Users="Vahid")]
```

اما هنوز حالت استفاده از Roles در فیلتر Authorize باقی مانده است. برای فعال سازی خودکار بررسی نقش‌های کاربران نیاز است یک Role provider سفارشی را با پیاده سازی کلاس RoleProvider، طراحی کنیم. برای مثال:

```
using System;
using System.Web.Security;

namespace MvcApplication15.Helper
{
    public class CustomRoleProvider : RoleProvider
    {
        public override bool IsUserInRole(string username, string roleName)
```

```
        {
            if (username.ToLowerInvariant() == "ali" && roleName.ToLowerInvariant() == "User")
                return true;
            // blabla ...
            return false;
        }

        public override string[] GetRolesForUser(string username)
        {
            if (username.ToLowerInvariant() == "ali")
            {
                return new[] { "User", "Helpdesk" };
            }

            if(username.ToLowerInvariant()=="vahid")
            {
                return new [] { "Admin" };
            }

            return new string[] { };
        }

        public override void AddUsersToRoles(string[] usernames, string[] roleNames)
        {
            throw new NotImplementedException();
        }

        public override string ApplicationName
        {
            get
            {
                throw new NotImplementedException();
            }
            set
            {
                throw new NotImplementedException();
            }
        }

        public override void CreateRole(string roleName)
        {
            throw new NotImplementedException();
        }

        public override bool DeleteRole(string roleName, bool throwOnPopulatedRole)
        {
            throw new NotImplementedException();
        }

        public override string[] FindUsersInRole(string roleName, string usernameToMatch)
        {
            throw new NotImplementedException();
        }

        public override string[] GetAllRoles()
        {
            throw new NotImplementedException();
        }

        public override string[] GetUsersInRole(string roleName)
        {
            throw new NotImplementedException();
        }

        public override void RemoveUsersFromRoles(string[] usernames, string[] roleNames)
        {
            throw new NotImplementedException();
        }

        public override bool RoleExists(string roleName)
        {
            throw new NotImplementedException();
        }
    }
}
```

در اینجا حداقل دو متد IsUserInRole و GetRolesForUser باید پیاده سازی شوند و مابقی اختیاری هستند.

بدیهی است در یک برنامه واقعی این اطلاعات باید از یک بانک اطلاعاتی خوانده شوند؛ برای نمونه به ازای هر کاربر تعدادی نقش وجود دارد. به ازای هر نقش نیز تعدادی کاربر تعریف شده است (یک رابطه many-to-many باید تعریف شود).
در مرحله بعد باید این Role provider سفارشی را در فایل وب کانفیگ برنامه در قسمت system.web آن تعریف و ثبت کنیم:

```
<roleManager>
      <providers>
        <clear />
        <add name="CustomRoleProvider" type="MvcApplication15.Helper.CustomRoleProvider"/>
      </providers>
    </roleManager>
```

همین مقدار برای راه اندازی بررسی نقش‌ها در ASP.NET MVC کفایت می‌کند. اکنون امکان تعریف نقش‌ها، حین بکارگیری فیلتر Authorize میسر است:

```
[Authorize(Roles = "Admin")]
public class HomeController : Controller
```

نظرات خوانندگان

نویسنده: Info
تاریخ: ۱۳۹۱/۰۱/۳۱ ۱۵:۵۴:۲۱

با سلام و درود و تحیتواقعا دست مریزاد جناب مهندسدر فیلتر [("Authorize(Roles="Admin")] آیا میشود را بصورت دینامیک تعیین کرد به نحوی که مثلا از بانک بشود خواندباتشکر

نویسنده: Msafdel
تاریخ: ۱۳۹۱/۰۱/۳۱ ۱۷:۱۵:۳۰

ممنون آقای نصیری. این سری آموزشها خیلی عالی بود و استفاده کردم.

نویسنده: سعید شیرزادیان
تاریخ: ۱۳۹۱/۰۱/۳۱ ۲۲:۰۲:۰۳

استاد نصیری عزیز سلام
مطالب در رابطه با Mvc بسیار جالب و مفید می باشد من گام به گام از مرحله اول تا 11 را تمرین کردم. اگر امکان دارد فراخوانی اطلاعات از دیتابیس و استفاده از آنها نیز اشاره ای بکنید.

با تشکر - شیرزادیان

نویسنده: Naser Tahery
تاریخ: ۱۳۹۱/۰۲/۰۱ ۰۰:۰۹:۴۶

ممنون. نمیدونم بحث رمزنگاری به این مطلب مربوط میشه یانه.(چون بحث امنیت است) با توجه به تجربه و نظر شما کدوم الگوریتم رمزنگاری برای اینکریپت کردن پسورد مناسب تره؟

نویسنده: مجتبی حسینی
تاریخ: ۱۳۹۱/۰۲/۰۱ ۰۱:۴۷:۰۹

با سلام و درود و تحیت
واقعا دست مریزاد جناب مهندس
در فیلتر [("Authorize(Roles="Admin")] آیا میشودRole را بصورت دینامیک و پویا تعیین کرد به نحوی که مثلا از بانک خوانده شود
باتشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۱ ۰۹:۱۵:۲۲

ترکیبی باید باشه. من از ترکیب SHA1 و MD5 و سپس معکوس کردن نتیجه استفاده می‌کنم. یعنی مثل بچه‌های خوب نباید از SHA1 معمولی استفاده کنید. برنامه برای شکستن این‌ها زیاد است (بروت فورس). اما زمانیکه کمی این رو پیچوندید، دیگه برنامه شکستن خودکار با الگوریتم‌های پردازش موازی براش نیست.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۱ ۰۹:۱۷:۳۶

Authorize زمانیکه به این نحو استفاده می‌شود تابع قوانین مثلا زبان سی شارپ است و نمی‌شود پارامتر آن‌را پویا تعریف کرد. اما می‌شود با ارث بری AuthorizeAttribute : CmsAuthorizeAttribute و ایجاد یک فیلتر سفارشی اینکار رو انجام داد. بعد در متد public override virtual bool AuthorizeCore فرصت خواهید داشت با بانک اطلاعاتی کار کنید.

نویسنده: Salehi
تاریخ: ۱۳۹۱/۰۲/۰۱ ۱۲:۵۹:۵۴

چطور میشه اسم controller، area و action رو به این تابع فرستاد؟ جستجو کردم ولی چیزی پیدا نکردم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۱ ۱۳:۲۶:۵۴

- از کلاس AuthorizeAttribute ارث بری کنید. بعد داخل آن یک خاصیت به نام مثلا public string AreaName تعریف کنید. این ویژگی سفارشی اکنون می‌تواند از پارامتر AreaName هم استفاده کند و استفاده داخلی از آن با تحریف متد AuthorizeCore میسر خواهد شد.
- اگر متد OnAuthorization را تحریف کنید، به filterContext.Controller دسترسی خواهید داشت.
- ضمن اینکه شما در سازنده این کلاس فیلتر سفارشی، فرصت مقدار دهی خواصی مانند Roles را بر اساس اطلاعات بانک اطلاعاتی خواهید داشت. یعنی به این شکل هم می‌شود آن‌را پویا تعریف کرد.
- توسط HttpContextBase httpContext متدهای تحریف شده به اطلاعات کاربر جاری می‌شود دسترسی یافت (httpContext.User.Identity.Name).

نویسنده: Salehi
تاریخ: ۱۳۹۱/۰۲/۰۱ ۲۰:۱۸:۴۱

این واقعا خیلی عالی بود. متشکرم.
پویا کردن AuthorizeAttribute واسه من جای سوال داشت که شما برطرفش کردید.
به نظرم استفاده از Attribute ها واقعا تو mvc خیلی کاربرد داره و خیلی کارها رو راحت میکنه. ولی متاسفانه تو اکثر کتابها فقط به معرفی اونهایی که موجود هستن پرداخته میشه و در مورد توسعه و شخصی سازی اونها صحبتی نمیشه. البته تو مباحث دیگه هم به همین صورته .

نویسنده: محمود رمضانی
تاریخ: ۱۳۹۱/۰۴/۱۷ ۲۲:۲۸

```
[HttpGet]
        public ActionResult LogOn(string returnUrl)
        {
            if (User.Identity.IsAuthenticated) //remember me
            {
                if (ShouldRedirect(returnUrl))
                {
                    return Redirect(returnUrl);
                }
                return Redirect(FormsAuthentication.DefaultUrl);
            }

            return View(); // show the login page
        }
```

بعد از پیاده سازی CustomRoleProvider وقتی که از این فیلتر استفاده می‌کنم :

```
[Authorize(Roles = "Admin")]
```

می ره به اکشن Logon اونجا

```
IsAuthenticated=true
```

هست و همین طور متد

```
ShouldRedirect(returnUrl)
```

هم مقدار true رو بر می‌گردونه و نتیجه اینکه دوباره بر می‌گرده به /User/Create و اونجا هم دوباره برمی گرده به همین اکشن و این Loop تکرار می‌شه.

این کد برای من اینجوری کار می‌کنه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۷ ۲۲:۴۴

علتش می‌تونه عدم دریافت نقش‌های کاربر لاگین کرده به سیستم باشد. اگر در قسمت ایجاد کاربر، نقش مورد نیاز، مدیریتی است و مدیر (شخص لاگین کرده) به این صفحه هدایت شده و باز به لاگین بر می‌گردد، یعنی شخص لاگین شده دارای این نقش نیست. یا اگر این دسترسی تعریف شده، قسمت پروایدرنقش‌های سفارشی، درست عمل نکرده.
ضمن اینکه به نظر این redirectها نیاز به یک شمارنده هم دارد که در یک چنین مواردی زیاده از حد تکرار نشود.

نویسنده: محمود رمضانی
تاریخ: ۱۳۹۱/۰۴/۱۷ ۲۳:۱۹

کد Web.config رو اینجوری تغییر دادم الان مشکل وقتی پیش میاد که کاربری که Admin نیست قبلا Login کرده و می‌خواد به اون صفحه دسترسی پیدا کنه.در بقیه موارد درست کار می‌کنه

```
<roleManager cacheRolesInCookie="true" defaultProvider="CustomRoleProvider" enabled="true">
      <providers>
        <clear />
        <add name="CustomRoleProvider" type="MvcApplication.Lib.CustomRoleProvider" />
      </providers>
    </roleManager>
```

نویسنده: محمود رمضانی
تاریخ: ۱۳۹۱/۰۴/۱۷ ۲۳:۳۴

و اون مشکل هم بدین صورت حل شد

```
public class CustomAuthorizeAttribute : AuthorizeAttribute
    {

        protected override void HandleUnauthorizedRequest(AuthorizationContext filterContext)
        {
            if (filterContext.HttpContext.Request.IsAuthenticated)
            {
                filterContext.Result = new HttpStatusCodeResult(403);
            }
            else
            {
                base.HandleUnauthorizedRequest(filterContext);
            }
        }
    }
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۸ ۰:۲۷

ممنون. روش خوبیه. پیشنهاد من این است که بجای 403 از روش زیر استفاده شود:

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

namespace SecurityModule
{
    [AttributeUsage(AttributeTargets.Class | AttributeTargets.Method, Inherited = true, AllowMultiple =
true)]
```

```
    public class SiteAuthorizeAttribute : AuthorizeAttribute
    {
        protected override void HandleUnauthorizedRequest(AuthorizationContext filterContext)
        {
            if (filterContext.HttpContext.Request.IsAuthenticated)
            {
                throw new UnauthorizedAccessException(); //to avoid multiple redirects
            }
            else
            {
                base.HandleUnauthorizedRequest(filterContext);
            }
        }
    }
}
```

به این ترتیب ریز جزئیات سعی در دسترسی غیرمجاز، توسط ELMAH ثبت خواهد شد.

نویسنده: محمود رمضانی
تاریخ: ۱۳۹۱/۰۴/۱۸ ۲۳:۴۷

ممنون

نویسنده: دانشجو
تاریخ: ۱۳۹۱/۰۶/۰۱ ۱۲:۴۶

سلام
بنده هم با مشکل اجرای چندگانه redirect مواجه هستم، از فیلتر فوق چه طور می‌توان جهت رفع آن مشکل استفاده کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۰۱ ۱۳:۲۶

توضیح داده شد. throw new Exception (در SiteAuthorizeAttribute یاد شده در نظر فوق) سبب می‌شود تا این redirectها متوقف شود و همچنین دسترسی غیرمجاز نیز لاگ شود در سیستم.

نویسنده: دانشجو
تاریخ: ۱۳۹۱/۰۶/۰۱ ۱۵:۱۲

متوجه کاربرد فیلتر فوق هستم، مشکل در نحوه به کارگیری آن است!
مسئله اینجاست که زمانیکه کاربر Authenticate شده صفحه ای که به آن دسترسی ندارد را درخواست میکند، فیلتر فوق و متد HandleUnauthorizedRequest اصلا اجرا نمی‌شود؛
آیا SiteAuthorizeAttribute باید در GlobalFilterCollection اضافه شود؟ یا...

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۰۱ ۱۶:۰

احتمالا **Role provider سفارش ی** شما درست ثبت نشده و کار نمی‌کند. در این مورد در انتهای متن قسمت جاری بحث شده.

نویسنده: Naser Tahery
تاریخ: ۱۳۹۱/۰۶/۲۶ ۰:۴۲

سلام آقای نصیری
با توجه به این نوع پیاده سازی[لایه دسترسی به داده‌ها توسط service layer] (+) اگر خواسته باشیم نقش‌های یک کاربر را

بدست بیاوریم، باید از لایه‌ی سرویس استفاده کنیم؟ یعنی شبیه به تصویر اول در این کامنت (+) لازم است که متغیر هایی را از نوع اینترفیس‌های لایه سرویس تعریف و بعد استفاده کنیم؟
چون در صورت استفاده از لایه سرویس ،مشکلاتی در کوئری گرفتنم به وجود میومد. یا بهتره بگم طرز استفاده از اونها رو نمیدونم.
آیا این کد قابل قبوله؟

```
public class CustomRoleProvider : System.Web.Security.RoleProvider
    {
        public override bool IsUserInRole(string username, string roleName)
        {
            //if (username.ToLowerInvariant() == "ali" && roleName.ToLowerInvariant() == "User")
            //    return true;
            // blabla ...
            return true;
        }

        public override string[] GetRolesForUser(string username)
        {
            using (var context = new PublishingContext())
            {
                var user = context.Users.Where(x => x.Username == username).FirstOrDefault();

                var roles = from ur in user.Roles
                            from r in context.Roles
                            where ur.Id == r.Id
                            select r.Role; //نام نقش
                if (roles != null)
                {
                    return roles.ToArray();
                }
            }

            return new string[] {};
        }
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۶ ۱:۲

CustomRoleProvider چون مستقیما و راسا توسط خود ASP.NET وهله سازی می‌شود، وارد پروسه متداول Controller Factory و تزریق وابستگی‌های ما نخواهد شد. به همین جهت در اینجا مجبور هستیم از الگوی service locator استفاده کنیم. چیزی مثل این:

```
public class CustomRoleProvider : RoleProvider
{
        public override bool IsUserInRole(string username, string roleName)
        {
            var rolesService = ObjectFactory.GetInstance<IRolesService>();
            return rolesService.IsUserInRole(username, roleName);
        }
//...
}
```

نویسنده: Naser Tahery
تاریخ: ۱۳۹۱/۰۶/۲۹ ۱۴:۷

ببخشید؛
با این تفاسیر متد IsUserInRole باید در داخل کلاس EfRole در لایه سرویس، پوشه EFServices به شکل زیر پیاده سازی بشه؟
یعنی میشه به طور مستقیم از شی context استفاده کرد؟

```
public class EfRole : EfGenericService<Role>, IRole
    {
        public EfRole(IUnitOfWork uow) : base(uow)
        {
        }

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

```
        {
            using (var context = new PublishingContext())
            {
                var user = context.Users.Where(x => x.Username.Equals(username,
StringComparison.CurrentCultureIgnoreCase)).FirstOrDefault();

                var roles = from ur in user.Rolls
                            from r in context.Rolls
                            where ur.Id == r.Id
                            select r.Rol;
                if (user != null)
                    return roles.Any(x => x.Equals(roleName,
StringComparison.CurrentCultureIgnoreCase));
                else
                    return false;
            }
        }
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۹ ۱۵:۶

نه به این شکل. نیازی به وهله سازی PublishingContext نیست چون الگوی واحد کار را نقض می‌کند؛ چون هر وهله سازی Context یعنی یک تراکنش جدا. از uow استفاده کنید. کار تزریق وابستگی‌ها توسط StructureMap در حالت service locator هم در لایه‌های زیرین انجام می‌شود. بنابراین کار مانند قبل است. فقط در کلاس CustomRoleProvider برخلاف معمول باید از ObjectFactory.GetInstance استفاده کرد. مابقی مسایل تفاوتی نمی‌کند.

نویسنده: رضا بزرگی
تاریخ: ۱۳۹۱/۰۷/۰۶ ۲۱:۴۵

برای یکپارچه‌سازی membership توکار ASP.NET با برنامه‌ی خودمون چه‌کار باید کرد؟
آیا استفاده از رابطه‌ی 1:1 با ویژگی ForeignKey برای این کار کفایت میکند؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۰۶ ۲۲:۱۳

از راهکار جدید مایکروسافت به نام Simple membership استفاده کنید.

نویسنده: رضا بزرگی
تاریخ: ۱۳۹۱/۰۷/۰۷ ۲۰:۲۱

ممنونم بابت معرفی این مورد.

این مهم در این سری آموزشی‌های سایت نیست. چون زمان نگارش اصلا SimpleMembership وجود نداشته.
به‌طور خلاصه توضیحاتی بدم:
استفاده از این پروایدر به صورت توکار فقط در تمپلیت Internet Application استفاده شده. خاصیت این سیستم این‌است که سختی آن سیستم قدیمی که از ASP.NET 2.0 باب شد را ندارد و واقعا Simple است.
موقع استفاده در web.config اصلی برنامه در زیر گره‌ی system.web موارد زیر را اضافه کنید:

```
<roleManager enabled="true" defaultProvider="SimpleRoleProvider">
  <providers>
    <clear/>
    <add name="SimpleRoleProvider" type="WebMatrix.WebData.SimpleRoleProvider, WebMatrix.WebData"/>
  </providers>
</roleManager>
<membership defaultProvider="SimpleMembershipProvider">
  <providers>
    <clear/>
    <add name="SimpleMembershipProvider" type="WebMatrix.WebData.SimpleMembershipProvider,
WebMatrix.WebData" />
```

```
  </providers>
</membership>
```

SimpleMembership به شما اجازه میدهد با برنامه‌ی خودتان در یک پایگاه داده ذخیره کنید. و در واقع یکپارچه کنید. برای این کار در AcountModel.cs و در کلاس UserProfile ویژگی‌های دلخواه دیگری را اضافه کنید (مثلا ایمیل، آدرس یا وب‌سایت). پس از آن باید در کلاس مدل Register فیلدهای مربوطه را اضافه کنید و همچنین data anotationهای مورد نیاز را اضافه کنید. این تغییرات را در رجیستر View هم انجام داده تا کاربر جاهای خالی را پر کند.

حال در کنترلر Account در متد Register با خاصیت HttpPost باید WebSecurity.CreateUserAndAccount را به‌درستی مقدار دهی کرد. چندین overload دارد. در overloadی که دارای قسمت object است میتوانیم از یک Anonymus Type استفاده کرده و اطلاعات مورد نیازمان را ثبت کنیم. مثلا:

```
WebSecurity.CreateUserAndAccount(
                model.UserName,
                model.Password,
                new { Mobile = model.Mobile },
                false);
```

برای اطلاعات بیشتر همراه با مثال: ^

نویسنده: donya
تاریخ: ۱۳۹۱/۰۷/۱۷ ۱۷:۵۴

سلام؛
User.Identity.IsAuthenticated همیشه مقدار false رو بر میگردونه دلیلش چیه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۱۷ ۱۸:۱۴

- بررسی کنید آیا authentication mode در فایل کانفیگ برنامه به Forms تغییر کرده یا نه (پیش فرض آن ویندوزی است نه Forms).
- همچنین مطابق روشی که در متن ذکر شد (متد LogOn) نیاز خواهد بود تا کوکی لازم و RedirectFromLoginPage صحیحی اعمال شود.

نویسنده: mina
تاریخ: ۱۳۹۱/۰۷/۲۶ ۹:۵۷

```
<location path="content">
   <system.web>
      <authorization>
         <allow users="*" />
      </authorization>
   </system.web>
</location>


<authentication mode="forms">
      <forms  loginurl="~/account/logon" slidingexpiration="true" timeout="2"/>
</authentication>
<authorization>
    <!--<allow roles="?"/>-->
  <deny users="?"/>
</authorization>
```

سلام آقای نصیری
ممنون از مطالب خوبتون

من تو این قسمت مشکل دارم تا اینجا درست کار میکنه که کسانی که authenticate نکردن برن به صفحه لاگین اما cssها رو با اینکه location هم گذاشتم نمیاره
بالای system.web هم می‌زنم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۶ ۱۰:۵۹

- در این تنظیمات فرض بر این است که پوشه content محل قرارگیری فایل‌های css است و همچنین این پوشه در ریشه سایت قرار دارد.
- در MVC نیازی به استفاده از تنظیم authorization -> deny users در وب کانفیگ نیست. روش مرجح استفاده از فیلتر Authorize است که توضیح داده شد.
- و بهتر است فایل‌های css و js از سیستم مسیریابی MVC حذف شوند:

```
// in RegisterRoutes -> Global.asax.cs
routes.IgnoreRoute("{*js}", new { js = @".*\.js(/.*)?"});
routes.IgnoreRoute("{*css}", new { css = @".*\.css(/.*)?"});
```

نویسنده: علی حق جو
تاریخ: ۱۳۹۲/۰۳/۱۵ ۲۰:۵۶

با سلام.
من یک Area برای Admin سایت در نظرگرفتم و می‌خواهم فیلتر Authorize را به کل آن area اعمال کنم. آیا امکانش وجود دارد؟
با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۵ ۲۱:۱۷

چنین کاری در مورد کل یک Area توصیه نمی‌شود ؛ چون Area بر اساس مسیریابی کار می‌کند و نه کنترلرها . روش توصیه شده تهیه یک base controller کلاس دارای فیلتر Authorize است و اعمال این کلاس پایه به کنترلرهای یک Area.

نویسنده: محسن.د
تاریخ: ۱۳۹۲/۰۳/۱۸ ۲۰:۵۶

سلام جناب نصیری

در این مقاله (پاراگراف مربوط به روش‌های اشتباه) هش ترکیبی رو روش مناسبی ندونسته به این دلیل که :
1- این امکان وجود داره که هش ایجاد شده غیرایمن باشه
2- اگر هدف ، آهسته کردن فرایند شکستن هش باشه ، روش‌های استانداردی مثل key stretching وجود داره .

ممنون میشم نظرتون رو در مورد این دلایل که ذکر کرده بفرمایید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۸ ۲۱:۴۱

- مقاله عنوان کرده اگر کار شما سورس باز است ترکیب فایده‌ای نداره چون مشخص است چکار کردید. (البته این رو برای ساده‌تر کردن کار خودش عنوان کرده!)
- key stretching یا هر روش دیگری، بحث و هدف اصلی من طرح یک نکته بود: از الگوریتم‌های هش عمومی مستقیما استفاده نکنید. روش پیچ و تاب آن باشد بر اساس روشی که انتخاب می‌کنید.
- من از روش ترکیبی استفاده می‌کنم. حتما هم استفاده می‌کنم. چون دقیقا می‌دونم اون کسانیکه این هش‌ها رو زمانیکه از دیتابیس شما بیرون کشیدند، چه ابزارهایی در اختیار دارند.

- حتی زمانیکه کار شما سورس باز است نیز از روش ترکیبی استفاده کنید. پردازش‌های موازی سنگین و استفاده از GPU برای الگوریتم‌های معمولی و متداول تهیه شده. حتی بانک‌های اطلاعاتی بزرگی که برای نگهداری هش‌های آماده تهیه شدن برمبنای هش‌های عمومی هستند. گیرم حتی اگر کار شما سورس باز باشد، تبدیل الگوریتم‌های ترکیبی آن به الگوریتم‌های بهینه قابل اجرای بر روی GPU کار هرکسی نیست.

استفاده مستقیم از الگوریتم‌های هش‌های متداول در یک پروژه عمومی با تعداد کاربر بالا = عین حماقت

نویسنده: احمد احمدی
تاریخ: ۱۳۹۲/۰۴/۰۷ ۰:۱۲

من دسترسی به کل محتوای سایت همانطور که فرمودید محدود کردم و به درستی بروی کنترلرها اعمال میشه اما بروی محتوای استاتیک (پوشه Content,Scripts و ...) اعمال نمیشه.
میدونم که نباید برای این محتوا سطح دسترسی تعیین کرد اما میخوام بدونم مشکل از کجاست و چرا این درخواست‌ها مستقیم توسط IIS مدیریت میشن و کنترل به برنامه منتقل نمیشه.
اصلا چرا نمیشه درخواست به این محتوا رو به یک کنترلر هدایت کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۰۷ ۰:۳۸

- برای اینکه عموما محتوای فایل‌های js و css پویا نیستند که نیازی به مدیریت آن‌ها توسط یک موتور پویا مانند ASP.NET باشد.
- همچنین اگر این پوشه‌ها مدیریت شوند، حتی اگر کل برنامه شما محافظت شده باشد، نیاز است برای صفحه لاگین حداقل سایت کامل و درست نمایش داده شود. یعنی باید محتوای استاتیک سایت بدون اعتبارسنجی هم قابل دسترسی باشد.
- درخواست به تمام فایل‌ها رو میشه در IIS یا حتی در ASP.NET مدیریت کرد. IIS یک قسمت نگاشت برای این مسایل دارد که چه فایلی به موتور ASP.NET نگاشت شود یا خیر. در IIS7 اگر حالت integrated pipline باشد، به صورت پیش فرض در وب کانفیگ runAllManagedModulesForAllRequests وجود دارد (یعنی در این حالت تمام فایل‌ها از موتور ASP.NET رد خواهند شد). در IIS6 است که یا باید دستی تنظیمات را تغییر داد یا کدنویسی کرد. ضمن اینکه مدیریت این مسایل توسط IIS سربار کمتری داره و مسایل اعمال کش و فشرده سازی و غیره رو میشه خارج از پروسه برنامه توسط IIS مدیریت کرد.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۲/۰۴/۲۴ ۱۵:۱۰

با سلام.
چگونه میتوان برای قسمت admin سایت از یک فرم لاگین و برای قسمت‌های دیگر از فرم لاگین دیگر استفاده کرد. تنظیمات آن در وب کانفیگ چگونه است؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۲۴ ۱۶:۴۳

بر اساس نقش‌های کاربران پس از لاگین، کدنویسی کنید:

```
if (User.IsInRole("Admins"))
  return Redirect("~/Admins/Default");
else if (User.IsInRole("Editors"))
  return Redirect("~/Editors/Default");
else //...
```

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۲/۰۴/۲۴ ۱۷:۴۴

سپاس از پاسخ شما.
ولی منظورم ، داشتن چند فرم لاگین بود. مثلاً من یک سیستم دارم که کاربران خاص علاوه بر نام کاربری و کلمه عبور ، چند فیلد

اضافی دیگر را نیز باید وارد کنند تا وارد سیستم مربوطه خود شوند.برخی دیگر از کاربران فقط نیاز به یک فرم لاگین با ورودی‌های نام کاربری ، پسوورد دارند. با تشکر فراوان.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۲۴ ۱۸:۲۰

در سؤال قبل تنظیمات وب کانفیگ رو می‌خواستید برای چند لاگین. در وب کانفیگ، کار خاصی انجام نمیشه. اونجا فقط مشخص میشه که طول عمر کوکی لاگین چند روز باشه یا پس از لاگین، کاربر به چه صفحه‌ای هدایت شود. به همین جهت عنوان کردم که چطور می‌توان کاربر را به صفحات دیگری با کدنویسی هدایت کرد.
در این سؤال دوم عنوان کردید که کاربران وارد سیستم می‌شوند. حالا من چندتا زیر سیستم دارم. می‌خواهم برای هر زیر سیستم بر اساس «نقش‌های» کاربران (واژه علمی «کاربران خاصی» که عنوان کردید) بتوانند به زیر سیستم خودشون وارد شوند. باید فیلتر AuthorizeAttribute را سفارشی کنید بر اساس Roleهای مشخص سیستم. اگر زیر سیستمی باید صرفا برای کاربران برای مثال Editor قابل دسترسی باشد، در این کلاس و فیلتر سفارشی مشتق شده از AuthorizeAttribute، اول باید چک کنید که کوکی سفارشی خاص حاصل از ورود موفقیت آمیز به صفحه لاگین دوم، تنظیم شده یا خیر (یا در ساده‌ترین حالت از سشن استفاده کنید). اگر خیر، بر اساس Role مشخص صفحه جاری، به یک صفحه لاگین ثانویه هدایت شود تا کاربر بتواند کوکی یا سشن لازم را پس از لاگین دوم تولید کند.

نویسنده: سهیل
تاریخ: ۱۳۹۲/۰۴/۲۸ ۰:۶

با سلام
در برنامه‌های ویندوزی که من کار کردم مجوزهای یک نقش قابل تغییر است، مثلا نقش مدیر می‌تواند نقشی را تعریف و برای آن مجوزهای را تعریف کند، آیا در mvc نیز می‌توان چنین کاری را پیاده سازی کرد؟
آنگونه که من فهمیدم ظاهرا با نوشتن نام یک نقش روی یک اکشن یک کار هاردکد انجام داده ایم و به طور داینامیک قابل تغییر نیست!

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۲۸ ۰:۲۸

این مورد بستگی به طراحی و آنالیز برنامه دارد. در جایی مانند صفحه تنظیمات SMTP Server برنامه، سطح دسترسی، فقط مدیریتی است و پویایی آن معنایی ندارد. در جای دیگری قرار است مطالبی ارسال شوند، اینجا دسترسی به دو نقش ادیتور و ادمین که شامل عده‌ای خواهند بود مفهوم دارد.
اما اگر سایتی پویا طراحی شده و از روز اول طرح دقیقی ندارد، مثلا در آن صفحات مختلفی به صورت پویا اضافه می‌شوند و قرار است به هر صفحه‌ای نقش(های) خاصی انتساب داده شوند که از روز اول پیش بینی و طراحی نشده، فقط کافی است همین فیلتر AuthorizeAttribute را با ارث بری از آن سفارشی سازی کرد تا بر اساس آدرس صفحه جاری (filterContext.HttpContext.Request.Url)، نقش‌های انتساب داده شده را از بانک اطلاعاتی بخواند و تصمیم گیری کند که آیا کاربر لاگین شده به سیستم (filterContext.HttpContext.User) می‌تواند به این صفحه دسترسی داشته باشد یا خیر و هدایتش کند (filterContext.HttpContext.Response.Redirect) به صفحه لاگین برنامه (FormsAuthentication.LoginUrl). یا حتی در اینجا بر اساس نام کنترلر و اکشن متد جاری هم می‌توان تصمیم گیری کرد:

```
filterContext.ActionDescriptor.ControllerDescriptor.ControllerName
filterContext.ActionDescriptor.ActionName
```

نویسنده: سهیل
تاریخ: ۱۳۹۲/۰۴/۲۸ ۲۳:۳۷

من از AllowAnonymous در کنترل و Home استفاده کردم اما متعجبم که فقط روی کنترل Account کار می‌کند و در کنترل Home بازهم نیاز به اعتبارسنجی است!

```
[AllowAnonymous]
```

```
    public class HomeController : Controller
    {
       [HttpGet]
        public ActionResult Index()
        {
            return View();
        }
    }

 [AllowAnonymous]
    public class SecurityController : Controller
    {
        [HttpGet]
        public ActionResult LogOn(string returnUrl)
        {
            if (User.Identity.IsAuthenticated) //remember me
            {
                if (CanRedirect(returnUrl))
                {
                    return Redirect(returnUrl);
                }
                return Redirect(FormsAuthentication.DefaultUrl);
            }

            return View(); // show the login page
        }
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۲۹ ۱۹:۴۱

امکان دیباگ کارهای شخصی، به همراه سفارشی سازی‌های خاص آن‌ها و قسمت‌های متعدد تاثیرگذار بر یکدیگر در آن‌ها، از راه دور وجود ندارد.

نویسنده: سهیل
تاریخ: ۱۳۹۲/۰۴/۳۰ ۱۲:۳۸

من با توجه به این گفته شما « نیاز است نکته «تنظیمات اعتبار سنجی اجباری تمام صفحات سایت» را به فایل وب کانفیگ برنامه اعمال نمائید تا نیازی نباشد فیلتر Authorize را در همه جا معرفی کرد. » کد زیر را به فایل کانفیگ (بعلاوه تنظیمات Form Authentication) اضافه کردم:

```
<authorization>
     <deny users="?" />
</authorization>
```

و فیلتر Authorize *را نیز* در Global معرفی کردم، که الان اینگونه فهمیدم اگر فیلتر را در فایل Global معرفی کردیم دیگر لازم به کد فوق نیست. چون با برداشتن آن برنامه به درستی کار کرد.

نویسنده: م کریمی
تاریخ: ۱۳۹۲/۰۴/۳۱ ۹:۵۲

با سلام؛ ما یک سیستمی داریم که 30 نوع نقش داره و هر نقش هم به یکسری کنترلرها و یا اکشن‌های متفاوتی دسترسی دارند . و همچنین تغییرات دسترسی‌ها به شدت بالاست . به طور مثال امروز 1 نقش به کنترلر 1 دسترسی دارد و 2 ساعت دیگه به کنترلر 2 دسترسی دارد و یا اکشن جدیدی به آن اضافه می‌شود و برنامه نویس نمیتواند مدام بالای کنترلرها و اکشن‌ها SiteAuth را بنویسد و کم کند و یا بردارد و .....
اگر هم بشود بعد از متدی کنترل این‌ها خیلی سخت می‌شه .. می‌خواستم نظر شما رو هم در این باره بدونم ممنون میشوم اگر راهنمایی کنید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۳۱ ۱۰:۲

امکان سفارشی سازی فیلتر Authorization وجود دارد . چند نمونه پیاده سازی آن برای ایده گرفتن:

Dynamic Controller/Action Authorization in ASP.NET MVC

MVC Dynamic Authorization

ASP.NET MVC Custom Authorize Attribute with Roles Parser

نویسنده: محمدرضا برنتی
تاریخ: ۱۳۹۲/۱۰/۲۷ ۱۹:۳۴

طبق مطلب بالا با اجرا شدن کد زیر باید ارتباط با اکشن هایی که شامل فیلتر [() Authorize] هستند برقرار شده و redirect به صفحه لاگین رخ ندهد اما همچنان پس از لاگین هم redirect انجام می‌شود ...

```
[HttpPost]
        public virtual ActionResult LogOn(Accounts acc)
        {
            FormsAuthentication.SetAuthCookie(acc.Username, acc.RememberMe);
            return View();
        }
```

لطفا اگر امکانش هست یک لینک دانلود نمونه پروژه لاگین رو قرار بدید . ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۲۷ ۱۹:۵۷

- اون قطعه کد کافی نیست. تنظیمات دیگری هم در متن ذکر شده که باید مطالعه کنید.
- نمونه پروژه‌ای که از روش Forms Authentication استفاده می‌کند: « سیستم مدیریت محتوای IRIS »

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۱۱/۰۲ ۲۳:۵۷

در این حالت پیاده سازی کلاس RolesProvider باید به چه صورتی باشد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۳ ۰:۳۱

کدام قسمت کلاس CustomRoleProvider : RoleProvider نوشته شده نیاز به توضیح بیشتری دارد؟

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۱۱/۰۳ ۸:۴۴

ببخشید، من منظورم کلاس RoleService است، اینترفیس IRoleService را به این صورت تعریف کرده ام :

```
public interface IRoleService
    {
        bool IsUserInRole(string username, string roleName);
        string[] GetRolesForUser(string username);
        void AddUsersToRoles(string[] usernames, string[] roleNames);
        string ApplicationName { get; set; }
        void CreateRole(string roleName);
        bool DeleteRole(string roleName, bool throwOnPopulatedRole);
        string[] FindUsersInRole(string roleName, string usernameToMatch);
        string[] GetAllRoles();
```

```
        string[] GetUsersInRole(string roleName);
        void RemoveUsersFromRoles(string[] usernames, string[] roleNames);
        bool RoleExists(string roleName);
    }
```

کلاس CustomRoleProvider هم که همونطور که فرمودید باید پیاده سازی کنیم، حالا من منظورم کلاس RoleService است :

```
public class RoleService: IRoleService
{
        private readonly IDbSet<Role> _role;
        private readonly IUnitOfWork _uow;
        public RoleService(IUnitOfWork uow)
        {
            _uow = uow;
            _role = uow.Set<Role>();
        }
        public bool IsUserInRole(string username, string roleName)
        {
            throw new NotImplementedException();
        }
        public string[] GetRolesForUser(string username)
        {
            throw new NotImplementedException();
        }
//...
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۳ ۹:۹

- نیازی نیست تمام متدهای RoleProvider دات نت پیاده سازی شوند. برای یک برنامه پیاده سازی دو متد IsUserInRole، GetRolesForUser کافی است.

- سپس دو کلاس Role و User را باید تعریف کنید. این دو رابطه many-to-many با هم دارند؛ یعنی هر کدام با یک ICollection به دیگری ارتباط پیدا می‌کنند. سپس این دو کلاس را در کلاس Context برنامه مطابق معمول توسط DbSetها در معرض دید EF قرار می‌دهید. مابقی آن کارکردن معمولی با این دو جدول اضافه شده به برنامه است:

```
public class EfRolesService : IRolesService
    {
        readonly IUnitOfWork _uow;
        readonly IDbSet<Role> _roles;
        public EfRolesService(IUnitOfWork uow)
        {
            _uow = uow;
            _roles = _uow.Set<Role>();
        }

        public IList<Role> FindUserRoles(int userId)
        {
            var query = from role in _roles
                        from user in role.Users
                        where user.Id == userId
                        select role;

            return query.OrderBy(x => x.Name).ToList();
        }

        public string[] GetRolesForUser(int userId)
        {
            var roles = FindUserRoles(userId);
            if (roles == null || !roles.Any())
            {
                return new string[] { };
            }

            return roles.Select(x => x.Name).ToArray();
        }

        public bool IsUserInRole(int userId, string roleName)
        {
```

```
            var query = from role in _roles
                        where role.Name == roleName
                        from user in role.Users
                        where user.Id == userId
                        select role;
            var userRole = query.FirstOrDefault();
            return userRole != null;
        }
    }
```

و در این حالت CustomRoleProvider به صورت زیر خواهد بود. در این روش فرض شده حین لاگین، user.Id در FormsAuthentication.SetAuthCookie تنظیم می‌شود ؛ یعنی userName در این RoleProvider به id آن تنظیم شده:

```
public class CustomRoleProvider : RoleProvider
    {
        public override bool IsUserInRole(string username, string roleName)
        {
            // Since the role provider, in this case the CustomRoleProvider is instantiated by
            // the ASP.NET framework the best solution is to use the service locator pattern.
            // The service locator pattern is normally considered to be an anti-pattern but
            // sometimes you have to be pragmatic and accept the limitation on the framework
            // that is being used (in this case the ASP.NET framework).

            var rolesService = ObjectFactory.GetInstance<IRolesService>();
            return rolesService.IsUserInRole(username.ToInt(), roleName);
        }

        public override string[] GetRolesForUser(string username)
        {
            var rolesService = ObjectFactory.GetInstance<IRolesService>();
            return rolesService.GetRolesForUser(username.ToInt());
        }
// مابقی نیازی نیست پیاده سازی شوند
```

نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۱۱/۰۳ ۹:۲۷

با سلام؛

User.Identity.Name وقتی که RedirectToAction به همون کنترلی که هستیم انجام میدیم کار نمیکنه؟

تو این اکشن

```
public ActionResult DeleteMessage( int id )
        {
            _messageService.DeleteMessage(id , User.Identity.Name);
            return RedirectToAction(MVC.User.SendMessages());
        }
```

وقتی به اکشن مقصد میره

```
public virtual ActionResult SendMessages()
{
    string s = User.Identity.Name;// Error : Object reference not set to an instance of an object.
    return View(_messageService.SendMessageListUser(User.Identity.Name));
}
```

تو جایی که کامنت کردم ارور میده و User.Identity.Name رو نال بر میگردونه! بقیه جاها از User.Identity.Name استفاده میکنم و هیچ مشکلی نیست!

با تشکر

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۱۱/۰۳ ۹:۴۵

ممنون، عالیه

فقط این ToInt به چه صورت username را به int تبدیل میکنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۳ ۱۰:۵

یک متد الحاقی است به این شکل:

```
public static int ToInt(this string data)
        {
            if (string.IsNullOrWhiteSpace(data)) return 0;
            int result;
            return int.TryParse(data, out result) ? result : 0;
        }
```

فقط هنگام لاگین، به userName خود id کاربر انتساب داده شده:

```
FormsAuthentication.SetAuthCookie(user.Id.ToString(CultureInfo.InvariantCulture), ...
// ...
FormsAuthentication.RedirectFromLoginPage(user.Id.ToString(CultureInfo.InvariantCulture), ...
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۳ ۱۰:۱۸

User.Identity.Name فقط زمانی مقدار دهی می‌شود که علاوه بر تنظیم authentication mode=Forms در وب کانفیگ، در حین لاگین دو مورد ذیل نیز وجود داشته باشند:

```
FormsAuthentication.SetAuthCookie(...
// ...
FormsAuthentication.RedirectFromLoginPage(...
```

ضمنا این رشته در MVC اگر کاربر لاگین نکرده باشد، صرفا string.Empty خواهد بود و نه null (با یک برنامه ساده و جدید آن‌را امتحان کنید). بنابراین مشکل از قسمت دیگری از کدهای شما ناشی می‌شود. stack trace را نیاز است دقیقا بررسی کنید.

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۱۱/۰۳ ۱۶:۴۰

خیلی ممنون،

User.Identity.Name مقدار صفر رو برمیگردونه، این موارد رو هم چک کردم :

- authentication mode=Forms

```
FormsAuthentication.SetAuthCookie(user.Id.ToString(CultureInfo.InvariantCulture), user.RememberMe);
FormsAuthentication.RedirectFromLoginPage(user.Id.ToString(CultureInfo.InvariantCulture),
user.RememberMe);
```

RoleProvider سفارشی رو هم در web.config ثبت کردم به این صورت :

```
<roleManager enabled="true" defaultProvider="CustomRoleProvider"
      cacheRolesInCookie="true">
        <providers>
          <clear />
          <add name="CustomRoleProvider"
               type="MyApp.Web.Helper.CustomRoleProvider"/>
        </providers>
      </roleManager>
```

یک مورد دیگر اینکه در حالت [("Authorize(Roles="Admins")] و یا [("Authorize(Users="user1")] حالت loop که اینجا گفته شده رخ میدهد (ظاهراً کاربران و نقش‌ها رو به درستی از دیتابیس دریافت نمی‌کند)، به نظر شما مشکل از کجا می‌تواند باشد؟(می

دونم امکان دیباگ از راه دور وجود ندارد، ولی گفتم شاید موردی رو از قلم انداخته باشم).

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۳ ۱۶:۵۳

- آیا فیلتر Authorize را با تگ authorization و location در وب کانفیگ با هم بکار برده‌اید؟ فقط از فیلتر Authorize در ASP.NET MVC استفاده کنید کافی است.
- بررسی کنید مقدار User.Identity.Name پس از لاگین چیست؟ (در یک صفحه معمولی)
- یک breakpoint داخل public override bool IsUserInRole قرار دهید و ببینید اصلا صدا زده می‌شود؟ چه پارامترهایی را دریافت می‌کند؟

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۱۱/۰۳ ۱۷:۵۳

- اگر از فیلتر Authrozie استفاده کنم البته به صورت ["Authorize(Roles="Admins"] حالت loop پیش میاد بنابراین برای رفع این مشکل از روش که خودتون گفتین استفاده کردم یعنی ["SiteAuthorize(Roles="Admins"] ، در این حالت خطای Attempted to perform an unauthorized operation. را دریافت میکنم.
- مقدار User.Identity.Name در صورت استفاده از فیلتر Authrozie یا SiteAuthorize (بدون تعیین کاربر و یا نقش) در یک صفحه معمولی مقدار صفر را برمیگرداند.
- breakpoint هم گذاشتم از صدا زده نمی‌شود.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۳ ۱۸:۲۷

اگر breakpoint شما صدا زده نمی‌شود، یکبار کوکی‌های برنامه را کلا پاک کنید. مجددا تست کنید. اگر IsUserInRole صدا زده نشد، یعنی مسیر قسمت type تعریف شده در role manager، قابل یافت شدن توسط برنامه نبوده.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۶ ۰:۴۸

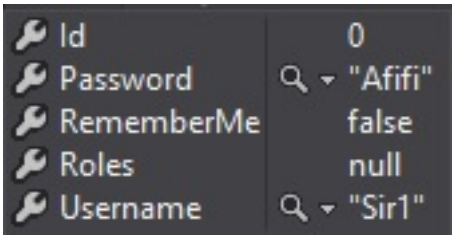
یک نکته‌ی تکمیلی

می‌توان CustomRoleProvider را کلا حذف کرد و بجای آن از FormsAuthenticationTicket که نقش‌های کاربر را هم قبول می‌کند، استفاده کرد. فقط کافی است این مقدار دهی در حین لاگین کاربر انجام شود. یک مثال:

Authorizing Users with Role-Based Security

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۱۱/۰۷ ۱۱:۳۷

قسمت type مشکلی نداره، breakpoint رو وقتی داخل متد GetRolesForUser قرار میدم فراخوانی میشه ولی چون مقدار Id صفر درنظر گرفته شده نمی‌تونه Role موردنظر رو برای این Id توی دیتابیس پیدا کنه، موقع لاگین فقط UserName و Password توسط سیستم Binding برای متد Login فرستاده میشه :



وقتی در فرم لاگین هم به صورت دستی Id رو ارسال میکنم، باز هم پیام Attempted to perform an unauthorized operation. دریافت میکنم. آیا تغییری دیگری در View لاگین نیاز هست اعمال بشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۷ ۱۴:۲۴

در حین لاگین، شما شیء User خاص خودتان را از دیتابیس واکشی می‌کنید. نام کاربری و کلمه عبور، توسط کاربر وارد شده، یک جستجوی LINQ متداول است برای یافتن وهله‌ای از شیء User. حالا این شیء یافت شده در صورت نال نبودن (ورود صحیح کلمه عبور و نام کاربری)، قابل استفاده است. Id آن‌را (که صفر هم نخواهد بود) به متدهای تنظیم کوکی و ری‌دایرکت FormsAuthentication انتساب دهید.

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۱۱/۰۷ ۱۵:۲۹

ممنون، تا اینجاش رو مشکلی ندارم و به این صورت که گفتید پیاده سازی کردم :

```
[HttpPost]
        [AllowAnonymous]
        [ValidateAntiForgeryToken]
        public ActionResult LogOn(User user, string returnUrl)
        {
            if (this.ModelState.IsValid)
            {
                if (_userService.IsValid(user))
                {
                    int userID = _userService.GetUser(u => u.Username == user.Username && u.Password ==
user.Password).Id;
                    FormsAuthentication.SetAuthCookie(userID.ToString(CultureInfo.InvariantCulture),
user.RememberMe);
                    if (shouldRedirect(returnUrl))
                    {
                        return Redirect(returnUrl);
                    }

FormsAuthentication.RedirectFromLoginPage(userID.ToString(CultureInfo.InvariantCulture),
user.RememberMe);
                }
            }
            this.ModelState.AddModelError("", "The user name or password provided is incorrect.");
            ViewBag.Error = "Login faild! Make sure you have entered the right user name and
password!";
            return View(user);
        }
```

در این صورت Id هم صفر نیست و مقدار به درستی داخل کوکی ذخیره می‌شود، مشکل اصلی من این اجرای چند گانه است یعنی موقعی که کنترلری را با ویژگی Authorize و تعیین Role و یا User مزین می‌کنم میدهد خطای Attempted to perform an unauthorized operation را میدهد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۷ ۱۵:۴۷

Role Provider شما عمل نمی‌کند. کلا کلاس آن‌را حذف کنید و از روش Authorizing Users with Role-Based Security بعد از لاگین شخص استفاده کنید (روش دومی بدون نیاز به Role Provider سفارشی؛ بر اساس اضافه کردن دستی نقش‌ها به کوکی رمزنگاری شده Roles سایت).

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۱۱/۰۷ ۲۱:۸

روش خیلی خوبیه ممنون، توی مقاله که ذکر کردیدخودش سه روش رو جهت چک کردن Role Membership پیشنهاد داده:
1- استفاده از ویژگی PrincipalPermissionAttribute :

```
[PrincipalPermissionAttribute(SecurityAction.Demand, Role = "MyRole")]
```

2- فراخوانی متد IsInRole
3- از طریق web.config
ولی می‌تونیم از فیلتر Authorize به جای مورد اول استفاده کنیم. مورد بعدی : کد زیر رو باید داخل رویداد Application_Start اضافه کرد :

```
AppDomain.CurrentDomain.SetPrincipalPolicy(PrincipalPolicy.WindowsPrincipal);
```

نویسنده: abbas
تاریخ: ۱۳۹۲/۱۱/۱۸ ۰:۴۳

سلام. با mvc 4 نوشتم ولی Role Provider صدا زده نشد. کلی گشتم ... تگ role manager رو به این صورت نوشتم حل شد:

```
<roleManager cacheRolesInCookie="true" defaultProvider="CustomRoleProvider" enabled="true">
```

شاید برای دوستانی که مثل من مشکل داشتند، کمکی باشه .

نویسنده: اس ام
تاریخ: ۱۳۹۲/۱۲/۲۹ ۵:۱۴

سلام من هم با MVC 5 نوشتم ولی RoleProvider صدا زده نمیشه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۹ ۹:۷

در دات نت 4.5، 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.
- وصله‌ی 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.

نویسنده: اس ام
تاریخ: ۱۳۹۳/۰۱/۰۱ ۳:۴۹

سلام ،سال نو مبارک ممنون از پاسخ شما.
من از vs 2013 update1 و win 8.1 استفاده میکنم و برنامه رو با NetFrameWork 4.5.1. تست کردم.یعنی شما می‌فرمایید که اگه این بسته‌ی ارتقاع رو نصب کنم مشکل حل میشه؟ (مشکل صدا زدن RoleProvider)
اگر هم تو سیستم لوکال خودم حل شه! به سیستم عامل Host که دسترسی ندارم .بخوام اونم به روز کنم.
من در دات نت 4 که با وب فرم کار می‌کردم همیشه از RoleProvider سفارشی استفاده می‌کردم و مشکلی نداشت.با مطالعه مقاله شما علاقه مند شدم تا از این به بعد برنامه هامو با MVC توسعه بدم. ولی چون آموزش شما با MVC4 هست و الان که MVC5 عرضه شده سوالات زیادی ذهنم رو مشغول کرده یکی اینکه آیا احتمال داره طی این زمان که از انتشار این مقاله می‌گذره روش‌ها عوض شده و یا روش بهتری برای MVC ارائه شده باشه؟
بازم ممنون از زحمتتتون.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۰۱ ۹:۱۱

- تمام هاست‌ها به دلایل امنیتی، سیستم عامل و وابستگی‌های آن‌را مرتبا به روز نگه می‌دارند. این مورد، اصل اول رعایت مسایل امنیتی هست.
- MVC5 فقط یک افزونه است برای MVC4 و MVC4 هم یک افزونه است برای MVC3. در MVC5 افزونه‌ای به نام ASP.NET Identity نیز ارائه شده‌است.
- پروژه‌ی سورس باز دیگری نیز در سایت به نام Iris membership برای پوشش مسایل بحث جاری تهیه شده‌است.

نویسنده: رشوند
تاریخ: ۱۳۹۳/۰۱/۲۶ ۱۷:۱۸

سلام؛بنده هم همین مشکل رو دارم.
برای اینکه قسمت‌های مختلف که شما مطمئنا از ان اگاه نیستید و شاید بران تاثیر بگذاره رو نداشته باشید.
یک پروژه تازه رو باز کردم یک راست رفتم سراغ web.config این کد رو در قسمت system.web وارد کردم

```
<authorization>
     <deny users="?" />
</authorization>
```

بعد بلافاصله یک کنترلر به نام Home ایجاد کردم و یک view برای index اد کردم...
قبل از اکشن index هم [AllowAnonymous] رو قرار دادم
ولی بعد از اجرا هنوز هم دسترسی ندارم ...
Access is denied.

آیا دلیل و نکته است که رعایت نکرده ام؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۲۶ ۱۷:۲۶

از تگ authorization در وب کانفیگ برنامه‌های ASP.NET MVC استفاده نکنید. این تنظیم بیشتر مربوط به برنامه‌های وب فرم است تا MVC (در اینجا فقط جهت یادآوری عنوان شده).
در برنامه‌های MVC فیلتر Authorize را به صورت Global تعریف کنید : «... امکان تعریف AuthorizeAttribute در فایل global.asax.cs و متد RegisterGlobalFilters آن به صورت سراسری نیز وجود دارد ...»

نویسنده: رشوند
تاریخ: ۱۳۹۳/۰۱/۲۷ ۱۳:۴۶

آقای نصیری سپاس.
مشکل حل شد, توضیح بدم اگر کسی مشکل من رو داشت دیگه نپرسه!
لینکی که دادید رو قبلا خوانده بودم ولی دقت نکردم در انتها ذکر کرده بودید *تعریف فیلترهای سراسری*
این کد رو متد Application_Start فایل Global.asax.cs اضافه کردم:

```
GlobalFilters.Filters.Add(new System.Web.Mvc.AuthorizeAttribute());
```

در همه جا اگر فرد لاگین کره بود می‌تونه دسترسی داشته باشه.
بعد در صفحه Home هم [AllowAnonymous] رو در ابتدا اضافه کردم تا بدون در نظر گرفتن لاگین اجازه دسترسی رو بده.

نویسنده: ایلیا اکبری فرد

تاریخ: ۱۳۹۳/۰۷/۱۷ ۱۷:۹

با سلام.
چرا مقدار returnUrl در متد login همواره نال است؟

```
[HttpPost]
[ValidateAntiForgeryToken]
[ValidateCaptchaAttribute]
[AllowAnonymous]
public virtual ActionResult Login(LoginWithCaptchaViewModel vm, string returnUrl)
{
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۱۷ ۱۸:۱۶

مقدار کوئری استرینگ returnUrl ، توسط سیستم اعتبارسنجی و redirect خودکار توسط آن بر اساس آدرس loginUrl در وب کانفیگ ، مقدار دهی می‌شود.

نویسنده: سعیدجلالی
تاریخ: ۱۳۹۳/۰۷/۲۴ ۱۰:۳۰

خوب با توجه به ارائه Identity 2 و قرارگیری آن در قالب پیش فرض MVC5 برای Visual Studio 2013 Update 2
آیا امکان انجام همین کار سفارشی سازی براساس جداول موجود در پایگاه داده (نه روش Code First) وجود دارد؟ من حدود یک هفته است که هر کاری انجام دادم به نتیجه نرسیدم در حالی که با آموزش ساده و روان شما خیلی راحت تونستم این کار رو انجام بدهم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۲۴ ۱۰:۴۲

- بررسی اختصاصی و مفصل asp.net identity در گروه مربوطه‌ی آن در سایت، انجام شده.
- برای استفاده از asp.net identity با روش database first دو سری ویدیو در یوتیوب هست: (^ و ^)

نویسنده: برنامه نویس کاشانی
تاریخ: ۱۳۹۳/۰۸/۲۴ ۱۵:۱۰

با عرض سلام و خسته نباشید
میشه به یه اکشن متد گفت که فقط همان کاربری که لاگین کرده به این متد دسترسی داشته باشه مثلا به این شکل :

```
[Authorize(Users = User.Identity.Name)]
البته این کد جواب نمیدهد برای فهم کلام نوشتم
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۲۴ ۱۵:۲۴

همینقدر که بنویسید [Authorize] کافی است.

نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۴/۰۲/۲۳ ۰:۴۱

از اونجا که فقط یک forms auth داریم ، در یک سیستمی که جدول کاربران و مدیران جداگونه باشه و دو صفحه لاگین برای هر دو باشه

کار به چه صورت هست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۲/۲۳ ۰:۴۶

به همین جهت مباحث Roles درنظر گرفته شده‌است. نقش مدیر، نقش نویسنده، نقش ادیتور، نقش کاربر صرفا خواننده و غیره. برای هر کدام یک جدول جدا درست نمی‌کنند. نقش این کاربرها را جداگانه مشخص می‌کنند.

نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۴/۰۲/۲۳ ۱:۱۴

الان من یک برنامه دارم که مشخصات یوزر با ادمین کاملا جداست و وجه اشتراکی ندارند
در نتیجه جداول جداست و فعلا کاربرها ، کاربران موبایل هستند و احتمالا در آینده سرویس وب آن هم در دسترس قرار می‌گیرید.
در این حالت وضعیت چگونه خواهد بود؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۲/۲۳ ۱:۲۵

- کار اضافی انجام دادید. جدول ادمین را حذف و بر اساس نقش‌های کاربرها کار کنید.
- همچنین این سیستم اساسا کاری به طراحی جداول شما ندارد. اصل کار آن در FormsAuthentication.SetAuthCookie انجام می‌شود. در متد ActionResult LogOn نحوه‌ی پیاده سازی لاگین و خواندن اطلاعات آن به اختیار شما است. همچنین نقش‌ها از public class CustomRoleProvider : RoleProvider دریافت می‌شوند. در اینجا مهم نیست که جداول به چه نحوی طراحی شده‌اند. مهم این است که خروجی IsUserInRole آن true هست یا false. مهم نیست که نحوه‌ی تهیه‌ی این true یا false قرار است از چه جدولی یا به چه نحو خاصی باشد.