عنوان: **هدایت خودکار کاربر به صفحه لاگین در حین اعمال Ajax ایی**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۲۴ ۱۳:۱۷
آدرس: www.dotnettips.info
گروه‌ها: MVC, jQuery Ajax, Security

در ASP.NET MVC به کمک فیلتر Authorize می‌توان کاربران را در صورت درخواست دسترسی به کنترلر و یا اکشن متد خاصی در صورت لزوم و عدم اعتبارسنجی کامل، به صفحه لاگین هدایت کرد. این مساله در حین postback کامل به سرور به صورت خودکار رخ داده و کاربر به Login Url ذکر شده در web.config هدایت می‌شود. اما در مورد اعمال Ajax ایی چطور؟ در این حالت خاص، فیلتر Authorize قابلیت هدایت خودکار کاربران را به صفحه لاگین، ندارد. در ادامه نحوه رفع این نقیصه را بررسی خواهیم کرد.

تهیه فیلتر سفارشی SiteAuthorize

برای بررسی اعمال Ajaxایی، نیاز است فیلتر پیش فرض Authorize سفارشی شود:

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

namespace MvcApplication28.Helpers
{
    [AttributeUsage(AttributeTargets.Class | AttributeTargets.Method, Inherited = true, AllowMultiple =
true)]
    public sealed class SiteAuthorizeAttribute : AuthorizeAttribute
    {
        protected override void HandleUnauthorizedRequest(AuthorizationContext filterContext)
        {
            if (filterContext.HttpContext.Request.IsAuthenticated)
            {
                throw new UnauthorizedAccessException(); //to avoid multiple redirects
            }
            else
            {
                handleAjaxRequest(filterContext);
                base.HandleUnauthorizedRequest(filterContext);
            }
        }

        private static void handleAjaxRequest(AuthorizationContext filterContext)
        {
            var ctx = filterContext.HttpContext;
            if (!ctx.Request.IsAjaxRequest())
                return;

            ctx.Response.StatusCode = (int)HttpStatusCode.Forbidden;
            ctx.Response.End();
        }
    }
}
```

در فیلتر فوق بررسی handleAjaxRequest اضافه شده است. در اینجا درخواست‌های اعتبار سنجی نشده از نوع Ajax ایی خاتمه داده شده و سپس StatusCode ممنوع (403) به کلاینت بازگشت داده می‌شود. در این حالت کلاینت تنها کافی است StatusCode یاده شده را مدیریت کند:

```
using System.Web.Mvc;
using MvcApplication28.Helpers;

namespace MvcApplication28.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        [SiteAuthorize]
        [HttpPost]
        public ActionResult SaveData(string data)
```

```
        {
            if(string.IsNullOrWhiteSpace(data))
                return Content("NOk!");

            return Content("Ok!");
        }
    }
}
```

در کد فوق نحوه استفاده از فیلتر جدید SiteAuthorize را ملاحظه می‌کنید. View ارسال کننده اطلاعات به اکشن متد SaveData، در ادامه بررسی می‌شود:

```
@{
    ViewBag.Title = "Index";
    var postUrl = this.Url.Action(actionName: "SaveData", controllerName: "Home");
}
<h2>
    Index</h2>
@using (Html.BeginForm(actionName: "SaveData", controllerName: "Home",
                method: FormMethod.Post, htmlAttributes: new { id = "form1" }))
{
    @Html.TextBox(name: "data")
    <br />
    <span id="btnSave">Save Data</span>
}
@section Scripts
{
    <script type="text/javascript">
        $(document).ready(function () {
            $("#btnSave").click(function (event) {
                $.ajax({
                    type: "POST",
                    url: "@postUrl",
                    data: $("#form1").serialize(),
                    // controller is returning a simple text, not json
                    complete: function (xhr, status) {
                        var data = xhr.responseText;
                        if (xhr.status == 403) {
                            window.location = "/login";
                        }
                    }
                });
            });
        });
    </script>
}
```

تنها نکته جدید کدهای فوق، بررسی xhr.status == 403 است. اگر فیلتر SiteAuthorize کد وضعیت 403 را بازگشت دهد، به کمک مقدار دهی window.location، مرورگر را وادار خواهیم کرد تا صفحه کنترلر login را نمایش دهد. این کد جاوا اسکریپتی، با تمام مرورگرها سازگار است.

نکته تکمیلی:

در متد handleAjaxRequest، می‌توان یک JavaScriptResult را نیز بازگشت داد تا همان کدهای مرتبط با window.location را به صورت خودکار به صفحه تزریق کند:

```
filterContext.Result =  new JavaScriptResult { Script="window.location = '" + redirectToUrl + "'"};
```

البته این روش بسته به نحوه استفاده از jQuery Ajax ممکن است نتایج دلخواهی را حاصل نکند. برای مثال اگر قسمتی از صفحه جاری را پس از دریافت نتایج Ajax ایی از سرور، تغییر می‌دهید، صفحه لاگین در همین قسمت در بین کدهای صفحه درج خواهد شد. اما روش یاد شده در مثال فوق در تمام حالت‌ها کار می‌کند.

نظرات خوانندگان

نویسنده: mahdi1391
تاریخ: ۱۳۹۱/۰۹/۲۴ ۱۴:۲۶

با سلام
ممنون از این مطلب واقعاً کاربردی، عالی بود.
اما موضوع زیر بسیار در محیط عملیاتی اتفاق می‌افتد:
فرض کنید کاربری در یک اداره درحال پر کردن یک فرم از برنامه ما است که چند ده فیلد داره، و طبیعیه که ممکن در این بین به دنیال کاری بره و برگرده و بقیه فرم رو پرکنه و در نهایت دکمه ثبت رو بزنه، مشکلی که پیش میاد اینه که به صفحه لاگین هدایت میشه و وقتی دوباره به اون فرم بر میگرده تمام اطلاعاتی که وارد کرده بود از بین میره و این کاربر به نوعی از برنامه ما متنفر میشه. یکی از راه حل‌های این مشکل این است که به جای هدایت کاربر به صفحه لاگین، با یک jQuery Modal Dialog دوباره نام کاربری و کلمه عبور از کاربر دریافت بشه و اگر صحیح بود Dialog بسته بشه و اگر غلط بود همچنان Modal بمونه.

نویسنده: davmszd
تاریخ: ۱۳۹۱/۰۹/۲۵ ۱۰:۵۸

با درود؛
اگه من بخوام این چک رو برای تقریبا همه اکشن‌ها تو کل پروژه انجام بدم الا چند تا اکشن خاص چه روشی رو پیشنهاد میکنید
من یه کنترلر بیس دارم که تمام کنترلرهای برنامم از اون به ارث رفتن و توی اون متد OnActionExecuting رور override کردم و یه همچین چکی رو دارم انجام میدم .
به نظرتون این کار درسته ؟
راه بهتری وجود داره ؟
شما چه روشی رو پیشنهادمیکنید ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۲۵ ۱۱:۴۷

- می‌تونید یک فیلتر رو به صورت سراسری تعریف کنید. باید در global.asax.cs تعریف شود: (^)
به این ترتیب به همه جا اعمال خواهد شد.

```
public static void RegisterGlobalFilters(GlobalFilterCollection filters)
{
    filters.Add(new SiteAuthorizeAttribute());
}
```

- در MVC4 برای معاف کردن تعدادی اکشن متد خاص از فیلتر سراسری یاد شده فقط کافی است از فیلتر جدید AllowAnonymous استفاده کنید.

نویسنده: شیرزادیان
تاریخ: ۱۳۹۱/۰۹/۲۵ ۲۳:۱۴

سلام؛
بسیار مفید و کارآمد بود. بازهم متشکرم

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۲/۱۱/۲۰ ۲۱:۱۲

من توی خطاهای لاگ شده توسط elmah توی سایتم خطای Server cannot set status after HTTP headers have been sent.
را در اجرای همین قسمت دریافت می‌کنم.
کار به درستی انجام میشه ولی لاگ سایت پر شده از این خطا. اشکال کار از کجای فیلتر فوق است؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۰ ۲۱:۳۲

این فیلتر اشکالی ندارد. احتمالا فیلترهای دیگری در همین لحظه در برنامه شما مشغول به کار هستند که روی Response تاثیر دارند. برای نمونه یکبار ترکیب فشرده سازی خروجی که Response.End داشت به همراه RSS Result ایی که آن هم Response.End داشت سبب بروز خطایی که نوشتید، شده بود. در یکی از این‌ها Response.End حذف شد تا مشکل برطرف شود.

نویسنده: علی محبی
تاریخ: ۱۳۹۲/۱۲/۲۵ ۱۳:۵۱

در حالتی که کاربر وارد شده و Authorize مقدار true دارد و ولی Role مد نظر را ندارد چطوری می‌توان کاربر را به صفحه لاگین هدایت کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۵ ۱۳:۵۵

اگر Role Provider تعریف شده درست ثبت شده باشد و توسط ASP.NET شناسایی شده باشد، فیلتر Authorize امکان ندارد چنین شخصی را مجاز بداند. بنابراین لازم نیست کار خاص اضافه‌تری را انجام دهید. بحث Request.IsAuthenticated متفاوت است و در مورد آن در SiteAuthorizeAttribute فوق، تمهیدات لازم صورت گرفته. البته سطر ctx.Response.StatusCode = (int)HttpStatusCode.Forbidden را هم می‌توانید پیش از صدور استثناء فراخوانی کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۱۷ ۱۴:۳۵

یک نکته‌ی تکمیلی

- برای رفع این مشکل (تداخل Forms authentication و تنظیم StatusCode) اگر از دات نت 4.5 به بعد استفاده می‌کنید، باید SuppressFormsAuthenticationRedirect را نیز پیش از ctx.Response.StatusCode اضافه کنید. به Response.End هم نیازی نخواهد بود.
- اگر از دات نت 4 استفاده می‌کنید، پیاده سازی SuppressFormsAuthenticationRedirect مخصوص آن‌را نیاز خواهید داشت.