عنوان: **ایجاد کپچایی (captcha) سریع و ساده در ASP.NET MVC 5**
نویسنده: مرتضی دلیل
تاریخ: ۱۳۹۳/۰۳/۲۶ ۹:۲۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, Security, CAPTCHA

در این مثال به کمک MVC5 ، یک کپچای ساده و قابل فهم را تولید و استفاده خواهیم کرد. این نوشته بر اساس این مقاله ایجاد شده و جزئیات زیادی برای درک افراد مبتدی به آن افزوده شده است که امیدوارم راهنمای مفیدی برای علاقمندان باشد.

با کلیک راست بر روی پوشه کنترلر، یک کنترلر به منظور ایجاد کپچا بسازید و اکشن متد زیر را در آن کنترلر ایجاد کنید:

```
public class CaptchaController : Controller
    {
        public ActionResult CaptchaImage(string prefix, bool noisy = true)
        {
            var rand = new Random((int)DateTime.Now.Ticks);
            //generate new question
            int a = rand.Next(10, 99);
            int b = rand.Next(0, 9);
            var captcha = string.Format("{0} + {1} = ?", a, b);

            //store answer
            Session["Captcha" + prefix] = a + b;

            //image stream
            FileContentResult img = null;

            using (var mem = new MemoryStream())
            using (var bmp = new Bitmap(130, 30))
            using (var gfx = Graphics.FromImage((Image)bmp))
            {
                gfx.TextRenderingHint = TextRenderingHint.ClearTypeGridFit;
                gfx.SmoothingMode = SmoothingMode.AntiAlias;
                gfx.FillRectangle(Brushes.White, new Rectangle(0, 0, bmp.Width, bmp.Height));

                //add noise
                if (noisy)
                {
                    int i, r, x, y;
                    var pen = new Pen(Color.Yellow);
                    for (i = 1; i < 10; i++)
                    {
                        pen.Color = Color.FromArgb(
                        (rand.Next(0, 255)),
                        (rand.Next(0, 255)),
                        (rand.Next(0, 255)));

                        r = rand.Next(0, (130 / 3));
                        x = rand.Next(0, 130);
                        y = rand.Next(0, 30);

                        gfx.DrawEllipse(pen, x - r, y - r, r, r);
                    }
                }

                //add question
                gfx.DrawString(captcha, new Font("Tahoma", 15), Brushes.Gray, 2, 3);

                //render as Jpeg
                bmp.Save(mem, System.Drawing.Imaging.ImageFormat.Jpeg);
                img = this.File(mem.GetBuffer(), "image/Jpeg");
            }

            return img;
        }
```

همانطور که از کد فوق پیداست، دو مقدار a و b ، به شکل اتفاقی ایجاد می‌شوند و حاصل جمع آنها در یک Session نگهداری خواهد شد. سپس تصویری بر اساس تصویر a+b ایجاد می‌شود (مثل 3+4). این تصویر خروجی این اکشن متد است. به سادگی می‌توانید این اکشن را بر اساس خواسته خود اصلاح کنید؛ مثلا به جای حاصل جمع دو عدد، از کاربرد چند حرف یا عدد که بصورت اتفاقی تولید کرده‌اید، استفاده نمائید.

فرض کنید می‌خواهیم کپچا را هنگام ثبت نام استفاده کنیم.

در فایل AccountViewModels.cs در پوشه مدل‌ها در کلاس RegisterViewModel خاصیت زیر را اضافه کنید:

```
[Required(ErrorMessage = "لطفا {0} را وارد کنید")]
        [Display(Name = "حاصل جمع")]
        public string Captcha { get; set; }
```

حالا در پوشه View/Account به فایل Register.Cshtml خاصیت فوق را اضافه کنید:

```
<div class="form-group">
                        <input type="button" value="" id="refresh" />

                        @Html.LabelFor(model => model.Captcha)

                        <img alt="Captcha" id="imgcpatcha" src="@Url.Action("CaptchaImage","Captcha")"
style="" />
                    </div>
```

وظیفه این بخش، نمایش کپچاست . تگ img دارای آدرسی است که توسط اکشن متدی که در ابتدای این مقاله ایجاد نموده‌ایم تولید می‌شود. این آدرس تصویر کپچاست. یک دکمه هم با شناسه refresh برای به روز رسانی مجدد تصویر در نظر گرفته‌ایم.

حالا کد ایجکسی برای آپدیت کپچا توسط دکمه refresh را به شکل زیر بنویسید (من در پایین ویوی Register ، اسکریپت زیر را قرار دادم):

```
<script type="text/javascript">
    $(function () {
        $('#refresh').click(function () {


            $.ajax({
                url: '@Url.Action("CaptchaImage","Captcha")',
                type: "GET",
                data: null
            })
            .done(function (functionResult) {
                $("#imgcpatcha").attr("src", "/Captcha/CaptchaImage?" + functionResult);
            });

        });
    });
</script>
```

آنچه در url نوشته شده است، شاید اصولی‌ترین شکل فراخوانی یک اکشن متد باشد. این اکشن در ابتدای مقاله تحت کنترلری به نام Captcha معرفی شده بود و خروجی آن آدرس یک فایل تصویری است. نوع ارتباط، Get است و هیچ اطلاعاتی به اکشن متد فرستاده نمیشود، اما اکشن متد ما آدرسی را به ما برمی‌گرداند که تحت نام FunctionResult آن را دریافت کرده و به کمک کد جی کوئری، مقدارش را در ویژگی src تصویر موجود در صفحه جاری جایگزین می‌کنیم. دقت کنید که برای دسترسی به تصویر، لازم است جایگزینی آدرس، در ویژگی src به شکل فوق صورت پذیرد. *

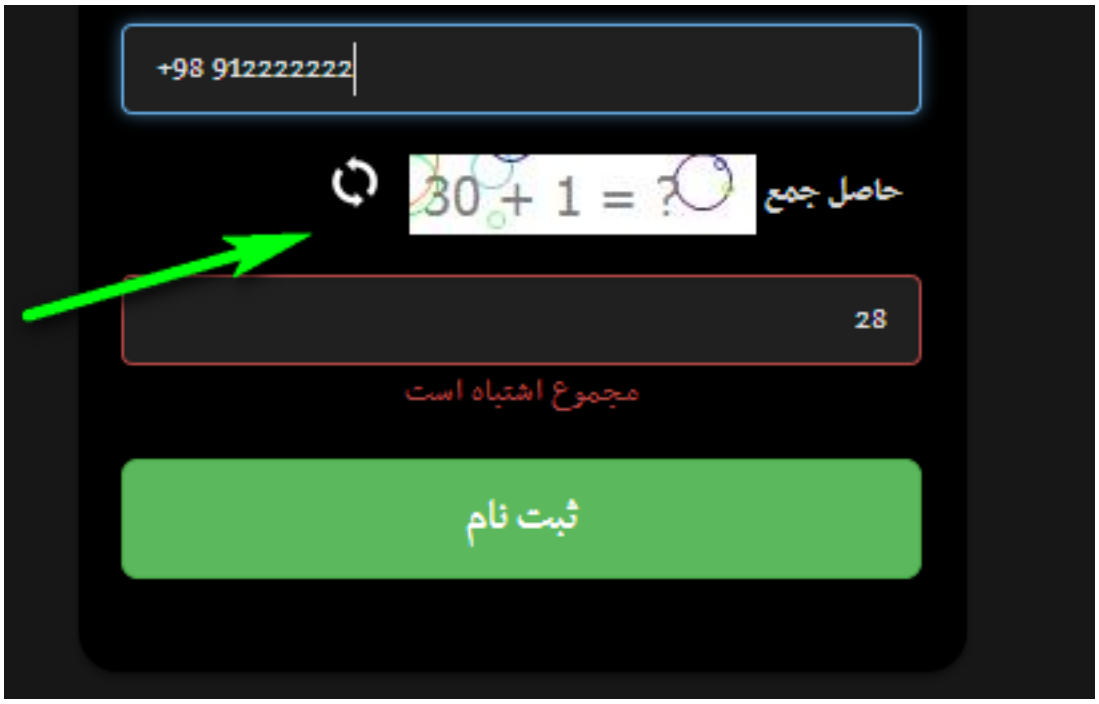
تنها کار باقیمانده اضافه کردن کد زیر به ابتدای اکشن متد Register درون کنترلر Account است.

```
if (Session["Captcha"] == null || Session["Captcha"].ToString() != model.Captcha)
            {
                ModelState.AddModelError("Captcha", "مجموع اشتباه است");
            }
```

واضح است که اینکار پیش از شرط if(ModelState.IsValidate) صورت میگیرد و وظیفه شرط فوق، بررسیِ برابریِ مقدار Session تولید شده در اکشن CaptchaImage (ابتدای این مقاله) با مقدار ورودی کاربر است. (مقداری که از طریق خاصیت تولیدی

خودمان به آن دسترسی داریم) . بدیهی‌است اگر این دو مقدار نابرابر باشند، یک خطا به ModelState اضافه می‌شود و شرط ModelState.IsValid که در اولین خط بعد از کد فوق وجود دارد، برقرار نخواهد بود و پیغام خطا در صفحه ثبت نام نمایش داده خواهد شد.

تصویر زیر نمونه‌ی نتیجه‌ای است که حاصل خواهد شد :



* اصلاح : دقت کنید بدون استفاده از ایجکس هم میتوانید تصویر فوق را آپدیت کنید:

```
$('#refresh').click(function () {

            var d = new Date();
            $("#imgcpatcha").attr("src", "Captcha/CaptchaImage?" + d.getTime());

        });
```

رویداد کلیک را با کد فوق جایگزین کنید؛ دو نکته در اینجا وجود دارد :

یک. استفاده از زمان در انتهای آدرس به خاطر مشکلاتیست که فایرفاکس یا IE با اینگونه آپدیت‌های تصویری دارند. این دو مرورگر (بر خلاف کروم) تصاویر را نگهداری میکنند و آپدیت به روش فوق به مشکل برخورد میکند مگر آنکه آدرس را به کمک اضافه کردن زمان آپدیت کنید تا مرورگر متوجه داستان شود

دو. همانطور که میبینید آدرس تصویر در حقیقت خروجی یک اکشن است. پس نیازی نیست هر بار این اکشن را به کمک ایجکس صدا بزنیم و روش فوق در مرورگرهای مختلف جواب خواهد داد.

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۳/۲۶ ۹:۳۳

ممنون از شما. فقط یک نکته‌ی کوچک در مورد memory stream هست که بهتره درنظر گرفته بشه. در این شیء متدهای ToArray و GetBuffer یکی نیستند. متد GetBuffer حجمی نزدیک به 2 برابر آرایه اصلی رو عموما داره و انتهاش یک سری بایت‌های اضافی هم شاید باشند. اما ToArray اصل دیتا رو بر می‌گردونه.

Note that the buffer contains allocated bytes which might be unused. For example, if the string "test" is written into the MemoryStream object, the length of the buffer returned from GetBuffer is 256, not 4, with 252 bytes unused. To obtain only the data in the buffer, use the ToArray method; however, ToArray creates a copy of the data in memory.