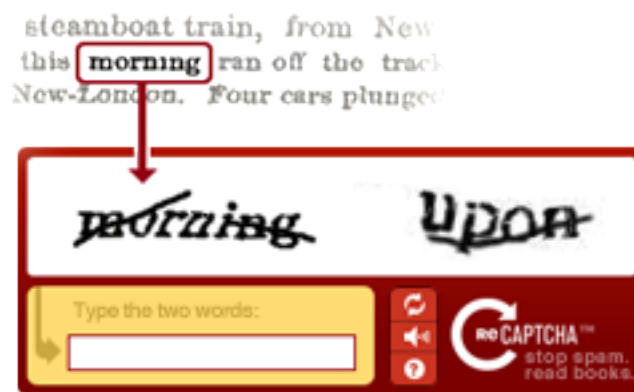


CAPTCHA جهت تشخیص انسان در وب استفاده می‌شود. در حقیقت می‌توان CAPTCHA را مانعی مابین منابع یک سایت و روبات‌های مخرب دانست. گوگل یک CAPTCHA رایگان جهت استفاده توسعه دهندگان وب مهیا کرده است که استفاده از آن را به چند دلیل می‌توان مثبت دانست:

- فراگیر بودن آن و استفاده سایت‌های مختلف از آن (بیش از 200.000 سایت).
- سهولت استفاده برای کاربران.
- سهولت استفاده توسط برنامه نویسان.
- دسترسی پذیری مناسب بدلیل وجود تلفظ و پخش عبارت، بصورت صوتی.
- امنیت بالا. بسیاری از CAPTCHA ها به سادگی قابل شکستن هستند!



حال اجازه دهید نگاهی به نحوه استفاده از reCAPTCHA در ASP.NET داشته باشیم:

- پس از اینکه به حساب کاربری خود در Google وارد شدید به [این صفحه +](#) مراجعه کنید تا یک API Key جدید برای سایت خود ایجاد کنید. برای استفاده از reCAPTCHA اولین پیش نیاز داشتن یک API Key معتبر است.
- [دانلود reCAPTCHA ASP.NET Library](#). این کتابخانه شامل کنترل reCAPTCHA است. پس از دریافت، **Recaptcha.dll** را توسط گزینه **Add Reference** به پروژه خود اضافه کنید.
- جهت استفاده از این کنترل، در بالای صفحه از دستور زیر استفاده کنید:

```
<%@ Register TagPrefix="recaptcha" Namespace="Recaptcha" Assembly="Recaptcha" %>
```

سپس کنترل reCAPTCHA را در محل مناسب قرار دهید:

```
<recaptcha:RecaptchaControl
  ID="recaptcha"
  runat="server"
  PublicKey="your_public_key"
  PrivateKey="your_private_key"
/>
```

4. حال جهت بررسی و اعتبار سنجی فرم باید از متد IsValid کلاس Page استفاده شود. نمونه کد مورد نظر در VB.NET بصورت زیر است:

```
Sub btnSubmit_Click(ByVal sender As Object, ByVal e As EventArgs)
    If Page.IsValid Then
        lblResult.Text = "You Got It!"
        lblResult.ForeColor = Drawing.Color.Green
    Else
        lblResult.Text = "Incorrect"
        lblResult.ForeColor = Drawing.Color.Red
    End If
End Sub
```

نظرات خوانندگان

نویسنده: زهرا
تاریخ: ۹:۱۴ ۱۳۹۱/۰۴/۳۱

با سلام

آیا این dll اپن سورس هست؟ از کجا میشه فهمید که درون این dll کدهای مخرب وجود ندارد؟

نویسنده: وحید نصیری
تاریخ: ۹:۳۶ ۱۳۹۱/۰۴/۳۱

دقت کردید در code.google.com هاست شده؟ یعنی سورس باز است.

نویسنده: وحید نصیری
تاریخ: ۹:۳۸ ۱۳۹۱/۰۴/۳۱

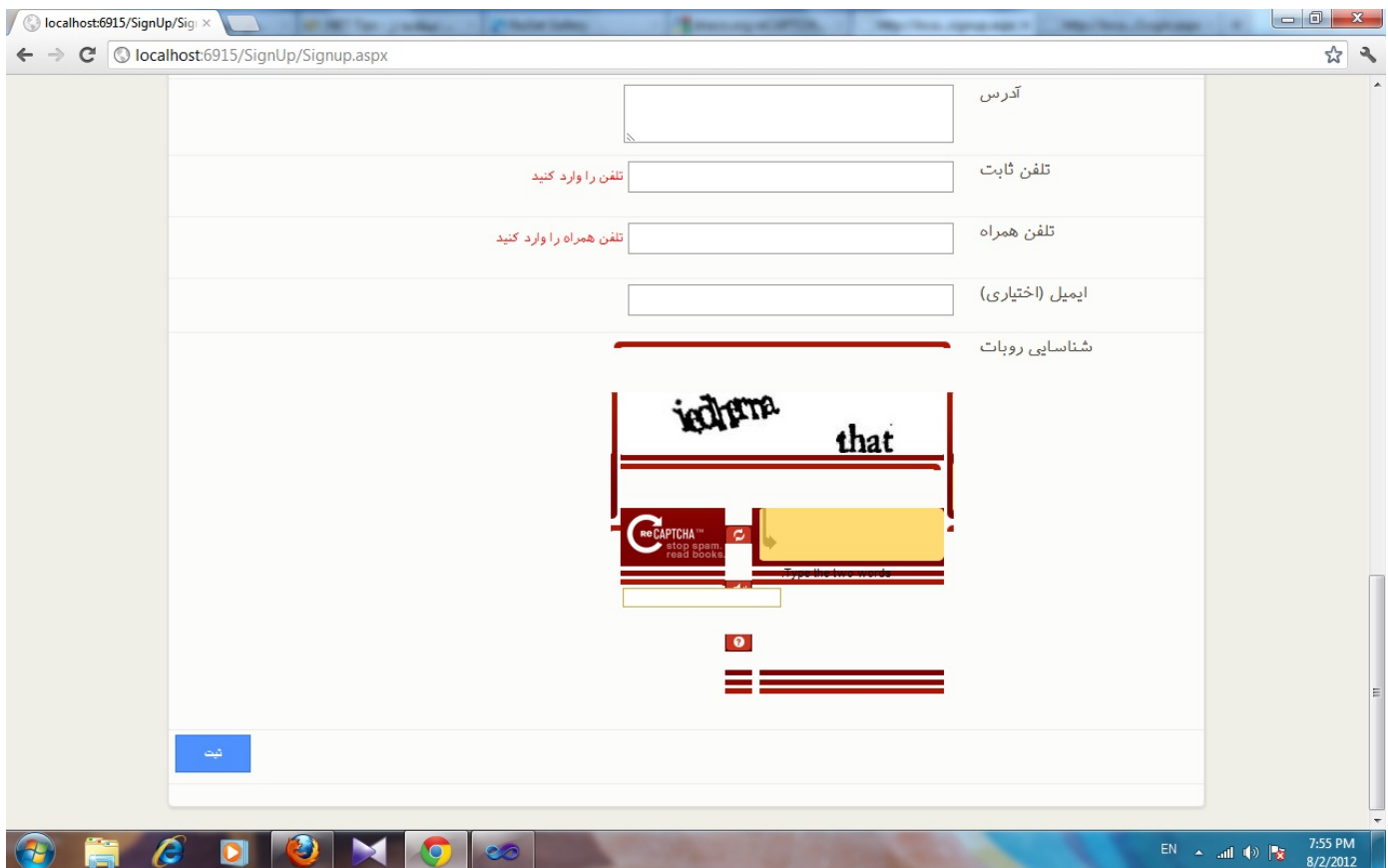
اگر دسترسی به گوگل کد ندارید، می‌شود از طریق [NuGet](#) نیز آنرا دریافت کرد.

نویسنده: علیرضا اسم‌رام
تاریخ: ۹:۴۸ ۱۳۹۱/۰۴/۳۱

سلام. جهت تکمیل نظر آقای نصیری و همچنین پاسخ به شما در مورد کدهای مخرب باید عرض کنم که بیش از 200.000 سایت معتبر در مقاطع زمانی مختلف از این سرویس تا به حال بهره گرفته اند. در این لیست می‌توان به سایت‌های بزرگی چون Twitter، Facebook و eBay اشاره کرد. جای نگرانی نیست.

نویسنده: naser
تاریخ: ۱۹:۵۸ ۱۳۹۱/۰۵/۱۲

سلام. من از این سیستم استفاده کردم. رو تست لوکال، ناقص لود میشه. با کروم و فایرفاکس هم تست کردم. قضیه چیه؟



نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۲ ۲۳:۴۳

این مورد بیشتر به تداخل css کلی سایت با این کنترل مربوط می‌شود. بهترین کار استفاده از firebug است. ابتدا در برگه net آن بررسی کنید پیام 404 ایی مشاهده می‌شود؟ ممکن است چیزی از قلم افتاده باشد. سپس به وسیله firebug به صورت live شروع به ویرایش css المان انتخاب شده کنید تا به نتیجه برسید.

نویسنده: na3er
تاریخ: ۱۳۹۱/۰۵/۱۳ ۱۳:۴۳

ممنون از راهنمایی تون. مشکلم حل شد مینویسم که اگر دوستانم دیگه هم به این مشکل خوردن بدونن از چیه. اولین مشکل از این بود که من به سلول‌های جدولم طول داده بودم که کمی بزرگ باشه (height) که برش داشتم و دومین مورد هم مربوط به تگ body بود که direction=rtl بود. این دو مورد نباید در مورد جایی که این کنترل قرار داره اعمال بشه. ممنون استاد نصیری.

نویسنده: حسین
تاریخ: ۱۳۹۱/۰۵/۱۷ ۲۲:۳۱

سلام
دوستان من نمیتونم فایل دل ال الشو دانلود کنم، تو رو خدا فایلشو بذارین منم بردارم. با اینکه تو لاگین جی میلیم هستم ازم پسرود میخاد که هر چی میزنم میگه اشتباهه ... ! با اینکه من داخل جی میلیم شدم و در تب دیگه مرورگرم دارم میل هامو میخونم.

دوستان منتظرم کمک کنید لطفا. با تشکر از مطالب مفیدتون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۷ ۲۲:۳۶

بالتر توضیح دادم. از [NuGet](#) استفاده کنید. (دسترسی به گوگل کد با IP ایرانی میسر نیست؛ ولی NuGet محدودیتی ندارد)

نویسنده: پژمان
تاریخ: ۱۳۹۳/۰۲/۲۱ ۱۴:۵۲

با عرض سلام.

ببخشید من کارهایی که گفته بودیدو انجام دادم، و در کد پشت باتن enterLogin نوشتم :

```
protected void enterLogin_Click( object sender, EventArgs e )
{
    if(Page.IsValid)
    {
        ScriptManager.RegisterStartupScript(this, GetType(), "showalert", "alert('خوش آمدید!');", true);
    }
    ScriptManager.RegisterStartupScript(this, GetType(), "showalert", "alert('خطا!');", true);
}
```

CAPTCHA نمایش داده میشود اما بررسی نمیکنه که عدد وارد شده درست است یا نه؟ میشه بهم بگید مشکله کار کجاست؟

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۲/۲۱ ۱۶:۵۶

کد شما قسمت `else` نداره یا حداقل اینجا نوشته نشده. بنابراین حتی اگر صفحه معتبر هم باشه، قسمت نمایش خطا اجرا میشه.

عنوان: CAPTCHAfa

نویسنده: بهروز راد

تاریخ: ۱۳۹۱/۰۵/۱۵ ۱۲:۴۶

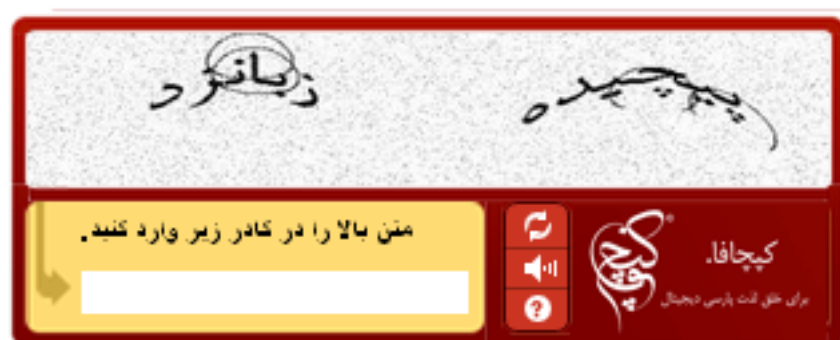
آدرس: www.dotnettips.info

برچسب‌ها: CAPTCHA, reCAPTCHA, CAPTCHAfa, ASP.Net, MVC, ASP.Net MVC

حتماً با CAPTCHA آشنا هستید. فرایندی که در طی آن متنی نمایش داده می‌شود که عمدتاً فقط یک انسان قادر به درک و پاسخگویی به آن است. با این کار از ارسال داده‌های بیهوده توسط ربات‌ها جلوگیری می‌شود. reCAPTCHA ایده‌ای است که با نمایش کلمات واقعی و اسکن شده از کتاب‌های قدیمی، بخشی از مشکلات را حل کرده و از کاربران اینترنت برای شناسایی کلماتی که رایانه توانایی خواندن آن‌ها را ندارد استفاده می‌کند. (شکل زیر)



با وارد کردن درست هر کلمه، بخشی از یک کتاب، روزنامه، و یا مجله‌ی قدیمی در رایانه شناسایی و به فرمت دیجیتال ذخیره می‌شود. به این شکل شما در دیجیتالی کردن متون کاغذی سهیم هستید. پروژه‌ی reCAPTCHA توسط گوگل حمایت می‌شود و در [این آدرس](#) قرار دارد. به تازگی تیمی از دانشکده فنی دانشگاه تهران به همراه انستیتو تکنولوژی ایلینویز شیکاگو، پروژه‌ای بر همین اساس اما برای متون فارسی با عنوان CAPTCHAfa تولید کرده اند (شکل زیر) که در [این آدرس](#) در دسترس است. امیدوارم این پروژه به گونه‌ای تغییر کند که برای دیجیتالی کردن متن‌های پارسی استفاده بشه. در حال حاضر، این پروژه از کلماتی از پیش تعریف شده استفاده می‌کند.



متأسفانه این پروژه در حال حاضر فقط توسط برنامه‌های PHP قابل استفاده است. از این رو بر آن شدم تا اون رو برای برنامه‌های ASP.NET (هم Web Forms و هم MVC) آماده کنم. برای استفاده از CAPTCHAfa نیاز به یک کلید خصوصی و یک کلید عمومی دارید که از [این آدرس](#) قابل دریافت است. کدهای پروژه‌ی Class Library به شرح زیر است.

```
// =====
// Captchafa demo for ASP.NET applications
// by: Behrouz Rad
// =====
namespace Captcha
{
    using System;
    using System.Collections.Generic;
    using System.IO;
    using System.Net;
    using System.Text;
    using System.Web;

    public class Captchafa
    {
        private static readonly string PRIVATE_KEY = "your private key";
        private static readonly string PUBLIC_KEY = "your public key";
        private static readonly string CAPTCHAFA_API_SERVER = "http://www.captchafa.com/api";
        private static readonly string CAPTCHAFA_VERIFY_SERVER =
"http://www.captchafa.com/api/verify/";

        private IDictionary<string, string> CaptchafaData
        {
            get
            {
                HttpContext httpContext = HttpContext.Current;

                string remoteIp = httpContext.Request.ServerVariables["REMOTE_ADDR"];
                string challenge = httpContext.Request.Form["captchafa_challenge_field"];
                string response = httpContext.Request.Form["captchafa_response_field"];

                IDictionary<string, string> data = new Dictionary<string, string>()
                {
                    {"privatekey" , PRIVATE_KEY },
                    {"remoteip" , remoteIp },
                    {"challenge" , challenge },
                    {"response" , response },
                };

                return data;
            }
        }

        public static string CaptchafaGetHtml()
        {
            return string.Format("<script type=\"text/javascript\"
src=\"{0}/?challenge&k={1}\"></script>", CAPTCHAFA_API_SERVER, PUBLIC_KEY);
        }

        public bool IsAnswerCorrect()
        {
            string dataToSend = this.CaptchafaPrepareDataToSend(this.CaptchafaData);

            string result = this.CaptchafaPostResponse(dataToSend);

            return result.StartsWith("true");
        }

        private string CaptchafaPrepareDataToSend(IDictionary<string, string> data)
        {
            string result = string.Empty;
            StringBuilder sb = new StringBuilder();

            foreach (var item in data)
            {
                sb.AppendFormat("{0}={1}&", item.Key, HttpUtility.UrlEncode(item.Value.Replace("@\"",
string.Empty)));
            }

            result = sb.ToString();

            sb = null;

            result = result.Substring(0, result.LastIndexOf("&"));

            return result;
        }

        private string CaptchafaPostResponse(string data)
        {
            StreamReader reader = null;

```

```

Stream dataStream = null;
WebResponse response = null;
string responseFromServer = string.Empty;

try
{
    WebRequest request = WebRequest.Create(CAPTCHAFA_VERIFY_SERVER);

    request.Method = "POST";
    request.ContentType = "application/x-www-form-urlencoded";

    byte[] byteData = Encoding.UTF8.GetBytes(data);

    request.ContentLength = byteData.Length;

    dataStream = request.GetRequestStream();

    dataStream.Write(byteData, 0, byteData.Length);

    dataStream.Close();

    response = request.GetResponse();

    dataStream = response.GetResponseStream();

    reader = new StreamReader(dataStream);

    responseFromServer = reader.ReadToEnd();
}
finally
{
    if (reader != null)
    {
        reader.Close();
    }

    if (dataStream != null)
    {
        dataStream.Close();
    }

    if (response != null)
    {
        response.Close();
    }
}

return responseFromServer;
}
}
}

```

استفاده در پروژه‌های ASP.NET Web Forms

ابتدا ارجاعی به فایل Captchafa.dll ایجاد کنید و سپس در روال Page_Load، کد زیر را قرار دهید. این کار برای تزریق اسکریپت CAPTCHAfa به صفحه استفاده می‌شود.

```

if (!IsPostBack)
{
    litCaptcha.Text = Captchafa.CaptchafaGetHtml();
}

```

litCaptcha، یک کنترل Literal است که اسکریپت تولید شده، به عنوان متن آن معرفی می‌شود. بررسی صحت مقدار وارد شده توسط کاربر (مثلاً در روال Click یک دکمه) به صورت زیر است.

```

Captchafa captchaFa = new Captchafa();

bool isAnswerCorrect = captchaFa.IsAnswerCorrect();

if (isAnswerCorrect)
{
    // پاسخ صحیح است
}
else
{
    // پاسخ صحیح نیست
}

```



```
}
```

استفاده در پروژه‌های ASP.NET MVC

ابتدا ارجاعی به فایل CaptchaFa.dll ایجاد کنید. در ASP.NET MVC بهتره تا فرایند کار رو در یک HTML helper کپسوله کنیم.

```
public static class CaptchaHelper
{
    public static MvcHtmlString CaptchaFa(this HtmlHelper htmlHelper)
    {
        return MvcHtmlString.Create(Captcha.CaptchaFa.CaptchaFaGetHtml());
    }
}
```

بررسی صحت مقدار وارد شده توسط کاربر (پس از ارسال فرم به Server) به صورت زیر است.

```
[HttpPost]
[ActionName("Index")]
public ActionResult CaptchaCheck()
{
    CaptchaFa captchaFa = new CaptchaFa();

    bool isAnswerCorrect = captchaFa.IsAnswerCorrect();

    if (isAnswerCorrect)
    {
        ViewBag.IsAnswerCorrect = true;
    }
    else
    {
        ViewBag.IsAnswerCorrect = false;
    }

    return View();
}
```

و در نهایت، کدهای View (از سینتکس موتور Razor استفاده شده است).

```
@using CaptchaFaDemoMvc.Helpers;

@{
    ViewBag.Title = "Index";
}

<form action="/" method="post">
@Html.CaptchaFa();
<input type="submit" id="btnCaptchaFa" name="btnCaptchaFa" value="آزمایش" />

@{
    bool isAnswerExists = ViewBag.IsAnswerCorrect != null;
}

@if (isAnswerExists)
{
    if ((bool)ViewBag.IsAnswerCorrect == true)
    {
        <span id="lblResult">پاسخ صحیح است</span>
    }
    else
    {
        <span id="lblResult">پاسخ صحیح نیست</span>
    }
}
</form>
```

دموی پروژه رو در [این آدرس](#) قرار دادم. پروژه‌ی نمونه نیز از [این آدرس](#) قابل دریافت است.

پ.ن: به زودی برخی بهبودها رو بر روی این پروژه انجام میدم.

نظرات خوانندگان

نویسنده: امیرحسین جلوداری
تاریخ: ۱۵:۹ ۱۳۹۱/۰۵/۱۵

اول دیدم هر چی تست میکنم میگه صحیح نیست! ... بعد دیدم باید فاصله باشه بین دو کلمه که صحیح است! در صورتی که تو recaptcha اصلی فاصله اهمیت نداره! ... چرا اینجوریه؟!

نویسنده: بهروز راد
تاریخ: ۱۵:۳۸ ۱۳۹۱/۰۵/۱۵

چون در زبان فارسی، حروف وقتی به هم بچسبن، شکلشون تغییر می‌کنه. بنابراین طبیعیه که "مسلسل" با "مسسل سل" فرق کنه.

نویسنده: ایمان نعمتی
تاریخ: ۱۵:۴۰ ۱۳۹۱/۰۵/۱۵

کار جالبی کردی بهروز جان
اما به نظر میرسه خود پروژه خیلی جالب نباشه
اینکه از یه سری کلمات از پیش تعیین شده استفاده میکنه و برای نمایش هم فقط دوتا خط به شکل بیضی روی کلمات میندازه زیاد جالب نیست به نظرم. شبیه به ری کپتچا خواسته کار کنه اما خب چیز خوبی از کار درنیامده

نویسنده: بهروز راد
تاریخ: ۱۵:۵۲ ۱۳۹۱/۰۵/۱۵

برای شروع، کار بسیار ارزشمندیه. ضمن اینکه تنوع کلمات مورد استفاده بسیار بالاست. خیلی مشتاقم که زودتر برای دیجیتالی کردن متون فارسی بشه ازش استفاده کرد، اما نیاز به یک متولی قوی و پشتیبانی خوب داره. مثل کاری که سایت [گنجور](#) انجام میده.

نویسنده: وحید نصیری
تاریخ: ۱۵:۵۳ ۱۳۹۱/۰۵/۱۵

نوشتن مشابه آن با دات نت زیاد سخت نیست. برای نمونه « [بررسی نحوه برنامه نویسی سایت نستعلیق آنلاین](#) » می‌تونه ایده خوبی برای شروع باشه. من این مطلب رو به همراه مطلب « [تبدیل عدد به حروف](#) » کنار هم گذاشتم و یک captcha فارسی ازش تهیه کردم.

نویسنده: ایمان نعمتی
تاریخ: ۱۹:۵۳ ۱۳۹۱/۰۵/۱۵

چه جالب!
امیدوارم استفاده از کپتچای فارسی در سایت‌های فارسی مرسوم بشه

نویسنده: علیرضا اسم‌رام
تاریخ: ۹:۵۶ ۱۳۹۱/۰۵/۱۶

با سلام. جهت حل مشکل کلمات از پیش تعریف شده، می‌توان از Google Translate استفاده کرد.
متأسفانه هنوز فرصت نکردم تا جزییات را دقیق ببینم، اما امیدوارم تلفظ و پخش کلمات بصورت صوتی خوب عمل کند!

نکته دیگر اینکه در مورد پیچیدگی تصویر CAPTCHA، بهترین روش استفاده از نقطه و پاره خط جهت تشویش تصویر است. معمولاً

بیشترین خطاها در OCR مربوط به این دو شکل پایه‌ی هندسی هستند. البته این موضوع بیشتر در مورد کلمات لاتین مطرح است. شاید در مورد کلمات فارسی بیضی‌هایی که دوستان روی متن انداختند پیشچیدگی را افزایش دهد. آقای ایمان نعمتی لطفاً فراموش نکنیم که هدف از تشویش تصویر، سختی کار برای انسان نیست بلکه پیچیده کردن کار روبات است. در مجموعه بسیار موضوع خوبی است. اگر دوستانی علاقه به این موضوع داشته باشند ایده و مطلب بسیار خوبی وجود دارد، هرچند پیاده سازی نشده است اما فکر می‌کنم ارزش انتشار و مطالعه را داشته باشد. البته جهت پیاده سازی آن چالشی بر سر راه نیست و فقط کمی وقت آزاد و البته تخصص نیاز دارد.

این مطلب را در این سایت میشه منتشر کرد؟ (با توجه به اینکه هنوز پیاده سازی ندارد و صرفاً بیان الگوریتم است).

نویسنده: هوشنگ

تاریخ: ۱۱:۱۱ ۱۳۹۱/۰۵/۱۸

"با وارد کردن درست هر کلمه، بخشی از یک کتاب، روزنامه، و یا مجله‌ی قدیمی در رایانه شناسایی و به فرمت دیجیتال ذخیره می‌شود. به این شکل شما در دیجیتالی کردن متون کاغذی سهیم هستید."

این منطقی نیست، با همان شیوه ای که کلمه تایپ شده کاربر با عکس مقایسه میشه، به همان طریق میشه متون کاغذی رو دیجیتالی کرد و نیازی به استفاده از این شیوه برای دیجیتالی کردن نیست

نویسنده: بهروز راد

تاریخ: ۱۲:۴ ۱۳۹۱/۰۵/۱۸

دوست من.

تمامی کلمات در یک متن (مخصوصاً اگر اون متن قدیمی باشه و برخی حروف یک کلمه پاک یا ناخوانا باشند) توسط رایانه و برنامه‌های OCR قابل تشخیص نیست.

مدل‌های CAPTCHA یی مانند reCAPTCHA که بر اساس دیجیتالی کردن متون پیاده سازی شدند، دو کلمه رو به کاربر نشان میدن:

1) کلمه ای که توسط رایانه به درستی تشخیص داده شده

2) کلمه یا بخشی از اون که رایانه نتونسته اون رو به درستی تشخیص بده.

در صورتی که کاربر، کلمه ای که توسط رایانه تونسته تشخیص داده بشه و برای رایانه مشخص هست رو صحیح وارد کنه، فرض بر این گرفته میشه که کلمه‌ی دوم یا بخشی از کلمه‌ی دوم که قابل تشخیص نبوده رو هم تونسته به درستی وارد کنه.

پس از این، میانگینی از تعداد عبارت ورودی کاربران برای کلمه‌ی نامشخص گرفته میشه و عبارتی که بیشترین فراوانی رو داشته به عنوان مورد قابل قبول ثبت میشه.

نویسنده: بهروز راد

تاریخ: ۱۲:۵۰ ۱۳۹۱/۰۵/۲۸

کامپوننت رو آپدیت کردم. نسخه‌ی اولیه بر اساس وفاداری به کدهای PHP و برگردانی از اونها بود. در نسخه‌ی جدید، متد CaptchafaGetHtml به GetHtml تغییر پیدا کرد، مدیریت خطاها اضافه شد و کلیدهای خصوصی و عمومی باید در زمان استفاده به کامپوننت پاس داده شوند. نسخه‌ی جدید را از لینک انتهای مقاله دریافت کنید.

عنوان: بررسی تصویر امنیتی (Captcha) سایت - قسمت اول

نویسنده: مهدی پایروند

تاریخ: ۱۳۹۱/۰۸/۲۸ ۲۱:۳۰

آدرس: www.dotnettips.info

برچسب‌ها: Security, CAPTCHA, CAPTCHAfa, handler

تصویر امنیتی و یا کپچا برای تشخیص و احراز انسان بودن استفاده کننده استفاده میشود و بصورت تصویری که استخراج نوشته‌های درون آن برای روبات‌ها بسیار سخت و یا نشدنی است ایجاد میشود و دارای انواع و اقسام متفاوتی است. در این میان برای استفاده از این امکان نمونه‌هایی در زبانهای مختلف تهیه شده که بسته به سلیقه و نیاز مورد استفاده قرار گرفته شده است. در این مقاله قصد داریم با بررسی تصویر امنیتی که در وبلاگ کنونی استفاده شده آنرا تا حدودی بازسازی کنیم.

تصویر آشنای کاربران سایت برای ورود به قسمت مدیریت که در صفحه مورد نظر از تصویر امنیتی استفاده شده است:



با توجه به آدرس لینک تصویر مشخص است که برای تولید این تصویر از هندلر استفاده شده و با توجه به پارامترهایی که به آن داده شده تصویر مورد نظر را ایجاد میکند و با فرمت مشخصی بر میگردداند:

<http://www.dotnettips.info/CaptchaImage/Show?text=H2yL5iOXIuu0dsBvu%2F405AnXkeacis3dMQ%2FHXA8h4A%2BjwDM0f7%2FRZMHvBG5pXYJ>

&foreColor=%231B0172
&fontSize=12
&fontName=Tahoma

پارامترهای پاس داده شده به ترتیب ذیل برای اهدافی قرار داده شده اند:

نام	مورد استفاده	مقدار
text	متنی که بصورت کد شده حاوی متن مورد استفاده تصویر امنیتی است	H2yL5iOXIuu0dsBvu%2F405AnXkeacis3dMQ%2FHXA8h4A%2BjwDM0f7%2FRZMHvBG5pXYJ
foreColor	رنگ مربوط به نوشته‌های تصویر	%231B0172
fontSize	سایز فونت	12
fontName	نام فونت	Tahoma

بنظر اهم پارامتر مورد نیاز همان متن است که بصورت کد شده در بالا به آن اشاره شد. برای این منظور باید کلاسی تهیه شود که حاوی متدهای:

1- انتخاب یه عدد تصادفی از بین دامنه ای که به آن داده میشود.

برای مثال یه عد از بین 100 تا 9999 که بصورت تصادفی انتخاب میشود

2- تبدیل به معادل حرفی آن

برای مثال از عدد "7062" به حروف آن یعنی "هفت هزار و شصت و دو" استخراج شود

3- کد کردن حرف تولید شده

حرف "هفت هزار و شصت و دو" را کد کرده و بصورت متنی شبیه

" H2yL5iOXIuu0dsBvu%2F405AnXkeacis3dMQ%2FHXA8h4A%2BjwDM0f7%2FRZMHvBG5pXYJ"

تبدیل کند

4- دیکد کردن و استخراج

البته این مورد با توجه به استفاده ای که ما داریم نیازی به پیاده سازی نیست

در کنار تصویر امنیتی از یک فیلد مخفی نیز برای نگهداری مقدار کد شده برای مقایسه با ورودی کاربر استفاده شده که در قسمت بعد ضمن ایجاد نمونه کاربردی بیشتر با قسمتهای مختلف آن آشنا میشویم.

نظرات خوانندگان

نویسنده: ahmadalli

تاریخ: ۲۱:۳۹ ۱۳۹۱/۰۸/۳۰

از نظر تئوری هر کدی قابل دی‌کد شدن هست. بهتر نیست برای کد اصلی از hash استفاده کنیم و hash کدی که کاربر وارد می‌کنه رو با hash اصلی مقایسه کنیم؟

نویسنده: مهدی پایروند

تاریخ: ۲۳:۴۴ ۱۳۹۱/۰۸/۳۰

با توجه به [قسمت دوم](#) از قسمت دیکد کردن استفاده نشده ولی چون از این نوع رمزنگرای استفاده شده در مواقعی ممکن است از دیکد کردن نیز استفاده کرد ولی به هرحال این رمزگذاری رو میشه بدخواه تغییر داد.


```

        {
            s = s + Dahyek[d12 - 10];
        }
        else
        {
            var d2 = d12 / 10;
            if (d2 != 0)
                s = s + Dahgan[d2] + " و ";
            var d1 = d12 % 10;
            if (d1 != 0)
                s = s + Yakan[d1] + " و ";
            s = s.Substring(0, s.Length - 3);
        }
        return s;
    }

    public static string ConvertIntNumberToFarsiAlphabatic(string snum)
    {
        var stotal = "";
        if (snum == "0") return Yakan[0];

        snum = snum.PadLeft(((snum.Length - 1) / 3 + 1) * 3, '0');
        var l = snum.Length / 3 - 1;
        for (var i = 0; i <= l; i++)
        {
            var b = int.Parse(snum.Substring(i * 3, 3));
            if (b != 0)
                stotal = stotal + Getnum3(b) + " " + Basex[l - i] + " و ";
        }
        stotal = stotal.Substring(0, stotal.Length - 3);
        return stotal;
    }
}

```

و برای کد کردن و دیکد کردن یعنی موارد سوم و چهارم مقاله قبلی، متن بدست آمده را که بعنوان قسمتی از آدرس تصویر در ادامه آدرس هندلر معرفی شده می‌آید تبدیل به یک string بی معنی برای بازدیدکننده میکند:

```

using System;
using System.IO;
using System.Security.Cryptography;
using System.Text;

namespace PersianCaptchaHandler
{
    public class Encryptor
    {
        #region constraints
        private static string Password { get { return "Mehdi"; } }
        private static string Salt { get { return "Payervand"; } }
        #endregion

        public static string Encrypt(string clearText)
        {
            // Turn text to bytes
            var clearBytes = Encoding.Unicode.GetBytes(clearText);

            var pdb = new PasswordDeriveBytes(Password, Encoding.Unicode.GetBytes(Salt));

            var ms = new MemoryStream();
            var alg = Rijndael.Create();

            alg.Key = pdb.GetBytes(32);
            alg.IV = pdb.GetBytes(16);

            var cs = new CryptoStream(ms, alg.CreateEncryptor(), CryptoStreamMode.Write);

            cs.Write(clearBytes, 0, clearBytes.Length);
            cs.Close();

            var encryptedData = ms.ToArray();

            return Convert.ToBase64String(encryptedData);
        }

        public static string Decrypt(string cipherText)
        {

```

```
// Convert text to byte
var cipherBytes = Convert.FromBase64String(cipherText);

var pdb = new PasswordDeriveBytes(Password, Encoding.Unicode.GetBytes(Salt));

var ms = new MemoryStream();
var alg = Rijndael.Create();

alg.Key = pdb.GetBytes(32);
alg.IV = pdb.GetBytes(16);

var cs = new CryptoStream(ms, alg.CreateDecryptor(), CryptoStreamMode.Write);

cs.Write(cipherBytes, 0, cipherBytes.Length);
cs.Close();

var decryptedData = ms.ToArray();

return Encoding.Unicode.GetString(decryptedData);
}
}
```

و نیز برای اعتبار سنجی عدد دریافتی از کاربر میتوان از عبارت با قاعده زیر استفاده کرد:

```
// Utils.cs
using System.Text.RegularExpressions;

namespace PersianCaptchaHandler
{
    public class Utils
    {
        private static readonly Regex NumberMatch = new Regex(@"^([0-9]*|\d*\.\d{1}?\d*)$",
        RegexOptions.Compiled);
        public static bool IsNumber(string number2Match)
        {
            return NumberMatch.IsMatch(number2Match);
        }
    }
}
```

برای استفاده نیز کافیت که هندلر مربوط به این کتابخانه را بطریق زیر در وب کانفیگ رجیستر کرد:

```
<add verb="GET" path="/captcha/" type="PersianCaptchaHandler.CaptchaHandler, PersianCaptchaHandler,
Version=1.0.0.0, Culture=neutral" />
```

و در صفحه مورد نظرتان بطریق زیر میتوان از یک تگ تصویر برای نمایش تصویر تولیدی و از یک فیلد مخفی برای نگهداری مقدار کد شده معادل عدد تولیدی که در هنگام مقایسه با عدد ورودی کاربر مورد نیاز است استفاده شود:

```
<!-- ASPX -->
<dl>
    <dt>تصویر امنیتی</dt>
    <dd>
        <asp:Image ID="imgCaptchaText" runat="server" AlternateText="CaptchaImage" />
        <asp:HiddenField ID="hfCaptchaText" runat="server" />
        <asp:ImageButton ID="btnRefreshCaptcha" runat="server" ImageUrl="/img/refresh.png"
            OnClick="btnRefreshCaptcha_Click" />
        <br />
        <asp:TextBox ID="txtCaptcha" runat="server" AutoCompleteType="Disabled"></asp:TextBox>
    </dd>
    <dd>
        <asp:Button ID="btnSubmit" runat="server" Text="ثبت" OnClick="btnSubmit_Click" />
    </dd>
</dl>
<asp:Label ID="lblMessage" runat="server"></asp:Label>
```

در زمان لود صفحه، تصویر امنیتی مقدار دهی میشود و در زمان ورود عدد توسط کاربر با توجه به اینکه کاربر حتما باید عدد وارد کند با عبارت با قاعده این اعتبار سنجی انجام میشود:

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!Page.IsPostBack)
        SetCaptcha();
}

private void SetCaptcha()
{
    lblMessage.Text =
    txtCaptcha.Text = string.Empty;

    var newNumber =
        RandomGenerator.Next(100, 999)
        ;

    var farsiAlphabetic =
    NumberToString.ConvertIntNumberToFarsiAlphabetic(newNumber.ToString());

    hfCaptchaText.Value =
        HttpUtility
        .UrlEncode(
            Encryptor.Encrypt(
                farsiAlphabetic
            )
        );

    txtCaptcha.Text = string.Empty;
    imgCaptchaText.ImageUrl =
        "/captcha/?text=" + hfCaptchaText.Value;
}
```

و بعد از ورود عدد از سمت کاربر از متد تبدیل به حروف استفاده کرده و این مقدار تولیدی با مقدار فیلد مخفی مقایسه میشود:

```
private string GetCaptcha()
{
    var farsiAlphabetic = NumberToString.ConvertIntNumberToFarsiAlphabetic(txtCaptcha.Text);

    var encryptedString =
        HttpUtility
        .UrlEncode(
            Encryptor.Encrypt(
                farsiAlphabetic
            )
        );

    return encryptedString;
}

private bool ValidateUserInputForLogin()
{
    if (!Utils.IsNumber(txtCaptcha.Text))
    {
        lblMessage.Text = "تصویر امنیتی را بطور صحیح وارد نکرده اید";
        return false;
    }

    var strGetCaptcha =
        GetCaptcha();

    var strDecodedVAlue =
        hfCaptchaText.Value;

    if (strDecodedVAlue != strGetCaptcha)
    {
        lblMessage.Text = "کلمه امنیتی اشتباه است";
        SetCaptcha();
        return false;
    }
    return true;
}
```

```
protected void btnSubmit_Click(object sender, EventArgs e)
{
    if (!ValidateUserInputForLogin()) return;
    lblMessage.Text = "کلمه امنیتی درست است";
}

protected void btnRefreshCaptcha_Click(object sender, ImageClickEventArgs e)
{
    SetCaptcha();
}
```

در آخر این [پروژه در کدپلکسی](#) قرار داده شده، و مشتاق نظرات و پیشنهادات شما هستم و نیز [نمونه مثال بالا](#) ضمیمه شده است

نظرات خوانندگان

نویسنده: احمد احمدی
تاریخ: ۱۳:۲۸ ۱۳۹۱/۰۹/۰۱

تشکر از مطلب مفیدتون.
جناب نصیری کلاسی برای [تبدیل عدد به حروف](#) نوشته اند که استفاده از اون کلاس در این پروژه ، زیبایی کار را دو چندان می کند!

نویسنده: مهدی پایروند
تاریخ: ۱۳:۳۸ ۱۳۹۱/۰۹/۰۱

پروژه در [کدپلکس](#)
پروژه در [سایت](#)

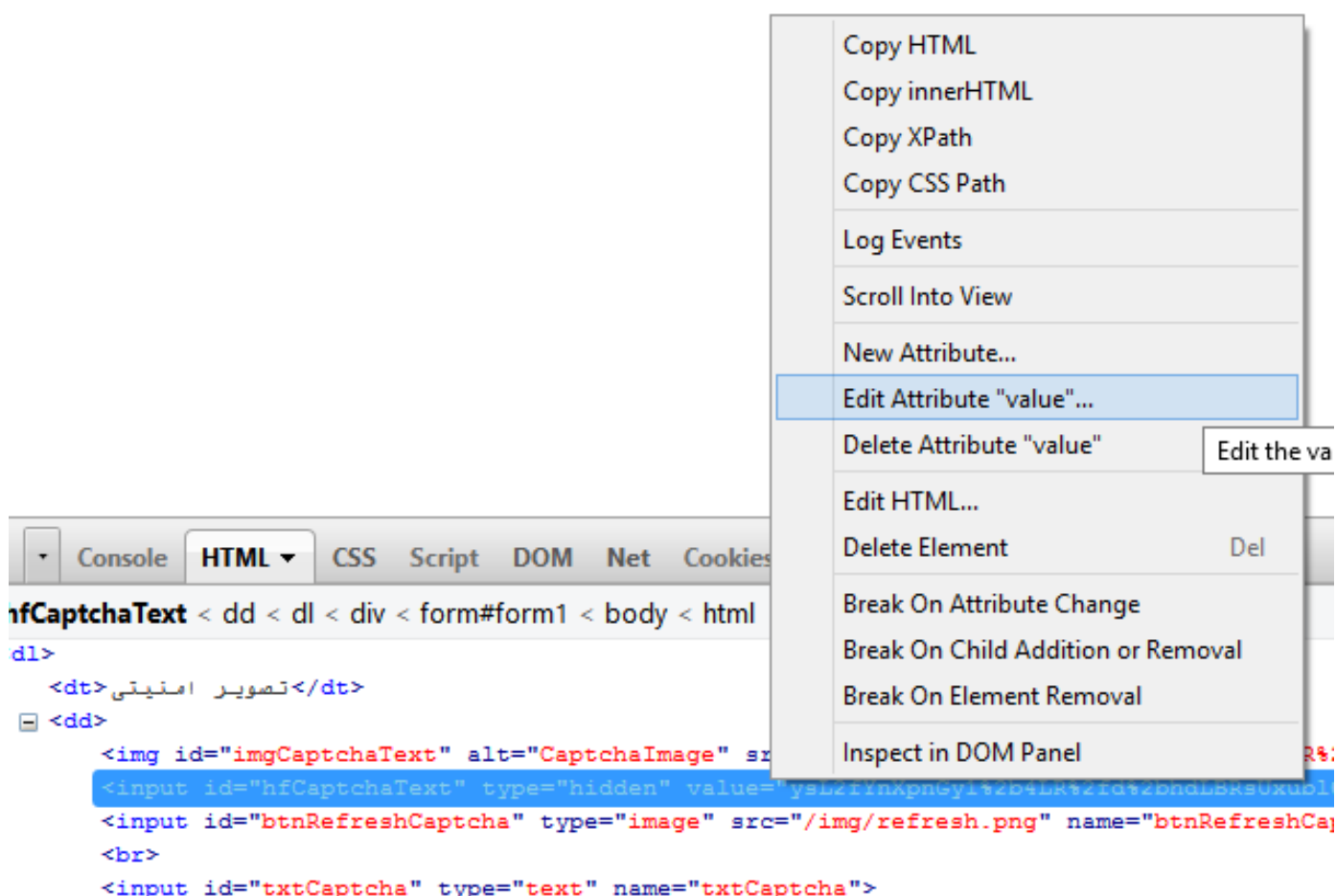
نویسنده: مهدی پایروند
تاریخ: ۱۳:۴۰ ۱۳۹۱/۰۹/۰۱

بسیار ممنون، با اجازه مهندس نصیری این کار انجام میشه.
لطفا درخواست رو در [قسمت مربوط به پروژه در سایت](#) مطرح کنید تا یک تاریخچه مطلوبی برای پروژه ایجاد بشه.

نویسنده: رحمت اله رضایی
تاریخ: ۱۱:۲۹ ۱۳۹۱/۰۹/۰۶

مشکلی که این روش دارد این است که با یک بار بارگذاری صفحه، خواندن عکس توسط انسان و یادداشت متن رمز گذاری شده می توان مقدار اولیه عکس را هر بار با ویرایش متن رمزگذاری شده به سرور ارسال کرد و تایید گرفت! مشکلی که سایت فعلی هم دارد.

برای حل این مشکل چه پیشنهادی دارید؟



نویسنده: مهدی پایروند
تاریخ: ۱۴:۱۹ ۱۳۹۱/۰۹/۰۶

ممنون از نظرتون،
موضوع مورد نظر، قسمتی است که متن تصویر برای این کپچا در آن نگهداری میشود،
1. در صورتی که بخواهیم از تواناییهای سمت سرور کمک گرفت: بنظرم برای اعتبار سنجی میتوان برای کاربران در بانک یک کد یونیک به ازای اولین هیت کاربر در سایت ایجاد کرد و این کد را بصورت رمزنگاری شده در کلاینت (کوکی) نگهداشت و مواردی از این دست را با این کد اعتبارسنجی کرد.
2. در صورتی که بخواهیم در سمت کلاینت همه چیز هندل شود: مطمئنا بهتر است که این فیلد مخفی در کنار باکس ورودی قرار داده نشود و برای نگهداری این فیلد از جاوا اسکریپت یا کوکی و از هردو کمک گرفت به این صورت که این مقدار کد شده بصورت مقداری غیر از این و با نامی بدون ربط در متغیری که از سمت سرور مقدار دهی میشود و فقط در سرور خوانده میشود تعویض کرد.

نویسنده: امیرحسین مرجانی
تاریخ: ۱۷:۵ ۱۳۹۱/۰۹/۰۷

لطفا راهنمایی کنید چطور پروژه رو دانلود کنم؟

نویسنده: مهدی پایروند
تاریخ: ۱۷:۵۶ ۱۳۹۱/۰۹/۰۷

برای دمو از این لینک میتونید استفاده کنید +
و همچنین در کدپلکس persiacaptchahandler.codeplex.com
و در همین سایت هم PersianCaptchaHandler

نویسنده: کیارش سلیمان زاده
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۲:۴۳

بهترین راه استفاده از Expiration date هستش، به این صورت که به متن رمزنگاری شده یک تاریخ انقضای 3 دقیقه ای داد، بدون استفاده از دیتابیس و کوکی.

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۲:۵۷

البته اگر از بازه عددی کوچکی استفاده بشه که ممکنه این تاریخ هم بلا استفاده بمونه، همچنین این تاریخ قرار هست کجا نگهداری بشه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۳:۰۴

در تمام قسمت‌های رمزنگاری شده سایت جاری از یک salt با طول عمر یک روز استفاده می‌شود. این salt به کلید رمزنگاری اطلاعات اضافه می‌شود. بنابراین اطلاعات رمزنگاری شده توسط آن کلید، فقط در طی روز جاری قابل رمزگشایی خواهند بود. برای مثال لینک یک پیغام خصوصی سایت را در جایی ذخیره کنید. تا پایان روز کار می‌کند. روز بعد به صورت خودکار منقضی شده (چون salt فردا چیز دیگری است) و اطلاعات روز قبل توسط کلید جدید قابل رمزگشایی نیست. به این ترتیب شخص نیاز خواهد داشت تا لینک جدیدی را در صفحه مداخل مرتبط دریافت کند.

نویسنده: کیارش سلیمان زاده
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۳:۰۵

در واقع باید متن رو مثلا شامل دو بخش، بخش اول، و بخش دوم هم تاریخ و زمان فعلی هستش که با یه splitter از هم جدا شدن و بعد رمزنگاری کرد، و در سمت سرور ابتدا بخش دوم رو با تاریخ و زمان فعلی (submit شدن فرم) مقایسه کرد و در صورتیکه اختلاف اونها بیشتر از 3 دقیقه (یا 2 یا ...) بود captcha غیر معتبر هستش.

نویسنده: کیارش سلیمان زاده
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۳:۰۷

یک روز برای یک captcha زمان زیادی نیست؟

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۳:۰۸

البته بنظرم برای نگهداری نکردن تاریخ در سرور میشه salt رو همین تاریخ امروز در نظر گرفت

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۰۸ ۱۳:۱۰

این لایه اول کلی هست (در تمام قسمت‌های سایت). تست کنید ببینید مطابق تصویر بالایی که ارسال شده می‌تونید در مورد کپچای سایت جواب بگیرید یا نه ...

نویسنده: کیارش سلیمان زاده

تاریخ: ۱۳:۱۸ ۱۳۹۱/۰۹/۰۸

تو لایه‌های بعدی چطور تست میکنید؟
طریقه ایجاد salt رو میشه بیشتر توضیح بدید؟
چیزی رو در دیتابیس هم ذخیره میکنید؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۲۱ ۱۳۹۱/۰۹/۰۸

salt مورد استفاده تاریخ ساده روز سیستم است؛ در بانک اطلاعاتی هم ذخیره نمیشود.

نویسنده: کیارش سلیمان زاده
تاریخ: ۱۳:۴۴ ۱۳۹۱/۰۹/۰۸

ممنون،
"تست کنید ببینید مطابق تصویر بالایی که ارسال شده میتونید در مورد کپچای سایت جواب بگیرید یا نه..."
بله تست کردم.
بعد از اینکه مشخص شد salt متعلق به امروز هستش، چطور تست میکنید که آیا مطابق با تصویر بالا، متن کپی شده یا اصلی هستش؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۴۷ ۱۳۹۱/۰۹/۰۸

در مرحله بعد از یک کش 5 دقیقه‌ای با کلیدی بر اساس IP شخص استفاده میکنم.

نویسنده: کیارش سلیمان زاده
تاریخ: ۱۳:۵۴ ۱۳۹۱/۰۹/۰۸

مگه IP نسبت به ISP برای بعضی‌ها یکسان نیست؟
در این صورت مشکلی پیش نمیداد؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۵۸ ۱۳۹۱/۰۹/۰۸

تا الان که کسی مشکلی گزارش نکرده.
همچنین مقدار ذخیره شده در این کش با طول عمر کوتاه 5 دقیقه‌ای به ازای هر بار ریفرش صفحه متفاوت خواهد بود.

نویسنده: رحمت اله رضایی
تاریخ: ۱۴:۱۱ ۱۳۹۱/۰۹/۰۸

هفته پیش کپچای سایت این مشکل را داشت ولی الان برطرف شده.

نویسنده: وحید نصیری
تاریخ: ۱۶:۱۲ ۱۳۹۱/۰۹/۰۸

گیرم اصلاً این کپچا نباشد. با anti forgery-token می‌خواهید چکار کنید؟ با مازول آنتی داس چطور و خیلی مسایل دیگر.
خلاصه صرف اینکه در داخل یک مرورگر به کمک یک افزونه برای مدت کوتاهی توانستید اطلاعاتی را شبیه سازی کنید به این معنا نیست که قابلیت استفاده وسیع و خارج از مرورگر را هم به سادگی دارا است (مثلاً پروت فورس پسورد کاربران). تا حد شبیه سازی کامل رفتار یک مرورگر باید پیش برید (و ... اصلاً کار ساده‌ای نیست).

نویسنده: رحمت اله رضایی

تاریخ: ۲۱:۵۴ ۱۳۹۱/۰۹/۰۸

هدف من از طرح این مساله این بود که نمایش تصویر امنیتی بدون در نظر گرفتن این مورد هیچ فایده ای برای سایت نخواهد داشت و کار به ماژول آنتی داس و مشکلات مربوط به آن کشیده می شود (یک لایه عقب تر). در واقع کپچا یک تصویر تزئینی می شود.

دستکاری اطلاعات ارسالی به سرور، فقط در اینجا مطرح نیست. عمده مورد استفاده آن در ویرایش idها (مثلا در DropDownList ها) ست. (ثبت نام مسکن برای شهری که هنوز ثبت نام آن فعال نشده، خرید بلیت جایگاه ویژه برای یک کنسرت، پرداخت بیمه با دستکاری هزینه و ...)

نویسنده: داود حنیفی
تاریخ: ۹:۵۱ ۱۳۹۱/۰۹/۲۱

با سلام
ببخشید این مورد تو vs2012 چطور کار میکنه؟
تست من تو vs2012 جواب نداد.
با تشکر از زحماتون.

نویسنده: مهدی پایروند
تاریخ: ۹:۴۶ ۱۳۹۱/۰۹/۲۱

من این [مثال](#) رو با vs2012 باز کردم مشکلی نداشت

نویسنده: داود حنیفی
تاریخ: ۱۰:۳۴ ۱۳۹۱/۰۹/۲۱

ببخشید من اشتباه کردم. منظورم با (.net 4.5) بود.

نویسنده: مهدی پایروند
تاریخ: ۱۰:۵۰ ۱۳۹۱/۰۹/۲۱

با اون دات نت 4.5 هم مشکلی نداشت
اگه کارتون با نمونه پروژه تبدیل شده راه میوفته من اونو تو این لینک گذاشتم
[PersianCaptcha_2012-12-11_10-51-20.rar](#)

نویسنده: مسعود
تاریخ: ۱۶:۱۹ ۱۳۹۱/۱۱/۲۶

سلام
لازم است در ابتدا بابت این پروژه تشکر کنم.
من مشکلی با این کپچا دارم و آن این است که روی محیط تست خودم کار می کند اما زمانی که آن را پابلیش می کنم و روی سرور اصلی قرار می دهم، تصویر کپچا نمایش داده نمی شود.
لازم به ذکر است که از پروتوکل https استفاده می کنم.
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۶:۵۴ ۱۳۹۱/۱۱/۲۶

یک حدس از راه دور:

تنظیم پیش فرض این پروژه در وب کانفیگ مربوط به IIS6 است (system.web/httpHandlers). برای IIS7 تنظیم جداگانه‌ای (system.webServer/handlers) نیاز دارد.

نویسنده: مسعود
تاریخ: ۱۷:۲۳ ۱۳۹۱/۱۱/۲۶

سلام
ممنون از پاسختون.
اما IIS من ۷ است و موردی رو هم که فرمودید پیش از این چک کردم ولی تغییری حاصل نشده بود.

نویسنده: مسعود
تاریخ: ۱۷:۳۲ ۱۳۹۱/۱۱/۲۶

با راه اندازی مجدد IIS مشکل رفع شد.

ممنون

نویسنده: مرتضی مختاری
تاریخ: ۱۶:۱۶ ۱۳۹۲/۱۰/۲۲

سلام؛ بنده آخر متوجه نشدم که از کپچا به این صورت استفاده کنیم یا نه؟
استفاده از expiration time به نظر من جواب نمیده چون توسط ربات تویه عرض یک دقیقه میشه هزار تا comment ثبت کرد. فکر کنم استفاده از captcha به همین روش و استفاده از anti forgery-token هیچ مشکل امنیتی نداشته باشه. شما چطوری مشکل این روش رو برطرف کردید؟ با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۶:۲۱ ۱۳۹۲/۱۰/۲۲

نمونه بهبود یافته مطلب جاری در اینجا « [نحوه ایجاد یک تصویر امنیتی \(Captcha\) با حروف فارسی در ASP.Net MVC](#) »

در این مطلب، سعی خواهیم کرد تا همانند تصویر امنیتی این سایت که موقع ورود نمایش داده می‌شود، یک نمونه مشابه به آنرا در ASP.Net MVC ایجاد کنیم. ذکر این نکته ضروری است که قبلاً آقای پایروند در [یک مطلب دو قسمتی](#) کاری مشابه را انجام داده بودند، اما در مطلبی که در اینجا ارائه شده سعی کرده ایم تا تفاوت‌هایی را با مطلب ایشان داشته باشد.

همان طور که ممکن است بدانید، اکشن متدها در کنترلرهای MVC می‌توانند انواع مختلفی را برگشت دهند که شرح آن در مطالب این سایت به مفصل گذشته است. یکی از این انواع، نوع ActionResult می‌باشد. این یک کلاس پایه برای انواع برگشتی توسط اکشن متدها مثل JsonResult، JsonResult می‌باشد. ([اطلاعات بیشتر را اینجا بخوانید](#)) اما ممکن است مواقعی پیش بیاید که بخواهید نوعی را توسط یک اکشن متد برگشت دهید که به صورت توکار تعریف نشده باشد. مثلاً زمانی را در نظر بگیرید که بخواهید یک تصویر امنیتی را برگشت دهید. یکی از راه حل‌های ممکن به این صورت است که کلاسی ایجاد شود که از کلاس پایه ActionResult ارث بری کرده باشد. بدین صورت:

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

namespace MVCPersianCaptcha.Models
{
    public class CaptchaImageResult : ActionResult
    {
        public override void ExecuteResult(ControllerContext context)
        {
            throw new NotImplementedException();
        }
    }
}
```

همان طور که مشاهده می‌کنید، کلاسی به اسم CaptchaImageResult تعریف شده که از کلاس ActionResult ارث بری کرده است. در این صورت باید متد ExecuteResult را override کنید. متد ExecuteResult به صورت خودکار هنگامی که از CaptchaImageResult به عنوان یک نوع برگشتی اکشن متد استفاده شود اجرا می‌شود. به همین خاطر باید تصویر امنیتی توسط این متد تولید شود و به صورت جریان (stream) برگشت داده شود

کدهای اولیه برای ایجاد یک تصویر امنیتی به صورت خیلی ساده از کلاس‌های فراهم شده توسط GDI+، که در دات نت فریمورک وجود دارند استفاده خواهند کرد. برای این کار ابتدا یک شیء از کلاس Bitmap با دستور زیر ایجاد خواهیم کرد:

```
// Create a new 32-bit bitmap image.
Bitmap bitmap = new Bitmap(width, height, PixelFormat.Format32bppArgb);
```

پارامترهای اول و دوم به ترتیب عرض و ارتفاع تصویر امنیتی را مشخص خواهند کرد و پارامتر سوم نیز فرمت تصویر را بیان کرده است. Format32bppArgb یعنی یک تصویر که هر کدام از پیکسل‌های آن 32 بیت فضا اشغال خواهند کرد، 8 بیت اول میزان آلفا، 8 بیت دوم میزان رنگ قرمز، 8 بیت سوم میزان رنگ سبز، و 8 تای آخر نیز میزان رنگ آبی را مشخص خواهند کرد

سپس شیئی از نوع Graphics برای انجام عملیات ترسیم نوشته‌های فارسی روی شیء bitmap ساخته می‌شود:

```
// Create a graphics object for drawing.
Graphics gfxCaptchaImage = Graphics.FromImage(bitmap);
```

خصوصیات مورد نیاز ما از gfxCaptchaImage را به صورت زیر مقداردهی می‌کنیم:

```
gfxCaptchaImage.PageUnit = GraphicsUnit.Pixel;  
gfxCaptchaImage.SmoothingMode = SmoothingMode.HighQuality;  
gfxCaptchaImage.Clear(Color.White);
```

واحد اندازه گیری به پیکسل، کیفیت تصویر تولید شده توسط دو دستور اول، و در دستور سوم ناحیه ترسیم با یک رنگ سفید پاک می‌شود.

سپس یک عدد اتفاقی بین 1000 و 9999 با دستور زیر تولید می‌شود:

```
// Create a Random Number from 1000 to 9999  
int salt = CaptchaHelpers.CreateSalt();
```

متد CreateSalt در کلاس CaptchaHelpers قرار گرفته است، و نحوه پیاده سازی آن بدین صورت است:

```
public int CreateSalt()  
{  
    Random random = new Random();  
    return random.Next(1000, 9999);  
}
```

سپس مقدار موجود در salt را برای مقایسه با مقداری که کاربر وارد کرده است در session قرار می‌دهیم:

```
HttpContext.Current.Session["captchastring"] = salt;
```

سپس عدد اتفاقی تولید شده باید تبدیل به حروف شود، مثلاً اگر عدد 4524 توسط متد CreateSalt تولید شده باشد، رشته "چهار هزار و پانصد و بیست و چهار" معادل آن نیز باید تولید شود. برای تبدیل عدد به حروف، آقای نصیری [کلاس خیلی خوبی نوشته اند](#) که چنین کاری را انجام می‌دهد. ما نیز از همین کلاس استفاده خواهیم کرد:

```
string randomString = (salt).NumberToText(Language.Persian);
```

در دستور بالا، متد الحاقی NumberToText با پارامتر Language.Persian وظیفه تبدیل عدد salt را به حروف فارسی معادل خواهد داشت.

به صورت پیش فرض نوشته‌های تصویر امنیتی به صورت چپ چین نوشته خواهند شد، و با توجه به این که نوشته ای که باید در تصویر امنیتی قرار بگیرد فارسی است، پس بهتر است آنرا به صورت راست به چپ در تصویر بنویسیم، بدین صورت:

```
// Set up the text format.  
var format = new StringFormat();  
int faLCID = new System.Globalization.CultureInfo("fa-IR").LCID;  
format.SetDigitSubstitution(faLCID, StringDigitSubstitute.National);  
format.Alignment = StringAlignment.Near;  
format.LineAlignment = StringAlignment.Near;  
format.FormatFlags = StringFormatFlags.DirectionRightToLeft;
```

و همچنین نوع و اندازه فونت که در این مثال tahoma می‌باشد:

```
// Font of Captcha and its size  
Font font = new Font("Tahoma", 10);
```

خوب نوشته فارسی اتفاقی تولید شده آماده ترسیم شدن است، اما اگر چنین تصویری تولید شود احتمال خوانده شدن آن توسط روبات‌های پردازش گر تصویر شاید زیاد سخت نباشد. به همین دلیل باید کاری کنیم تا خواندن این تصویر برای این روبات‌ها سخت‌تر شود، روش‌های مختلفی برای این کار وجود دارند: مثل ایجاد نویز در تصویر امنیتی یا استفاده از توابع ریاضی سینوسی و

کسینوسی برای نوشتن نوشته‌ها به صورت موج. برای این کار اول یک مسیر گرافیکی در تصویر یا موج اتفاقی ساخته شود و به شیء `gfxCaptchaImage` نسبت داده شود. برای این کار اول نمونه ای از روی کلاس `GraphicsPath` ساخته می‌شود،

```
// Create a path for text
GraphicsPath path = new GraphicsPath();
```

و با استفاده از متد `AddString`، رشته اتفاقی تولید شده را با فونت مشخص شده، و تنظیمات اندازه دربرگیرنده رشته مورد نظر، و تنظیمات فرمت بندی رشته را لحاظ خواهیم کرد.

```
path.AddString(randomString,
    font.FontFamily,
    (int)font.Style,
    (gfxCaptchaImage.DpiY * font.SizeInPoints / 72),
    new Rectangle(0, 0, width, height), format);
```

با خط کد زیر شیء `path` را با رنگ بنفش با استفاده از شیء `gfxCaptchaImage` روی تصویر `bitmap` ترسیم خواهیم کرد:

```
gfxCaptchaImage.DrawPath(Pens.Navy, path);
```

برای ایجاد یک منحنی و موج از کدهای زیر استفاده خواهیم کرد:

```
//-- using a sin wave distort the image
int distortion = random.Next(-10, 10);
using (Bitmap copy = (Bitmap)bitmap.Clone())
{
    for (int y = 0; y < height; y++)
    {
        for (int x = 0; x < width; x++)
        {
            int newX = (int)(x + (distortion * Math.Sin(Math.PI * y / 64.0)));
            int newY = (int)(y + (distortion * Math.Cos(Math.PI * x / 64.0)));
            if (newX < 0 || newX >= width) newX = 0;
            if (newY < 0 || newY >= height) newY = 0;
            bitmap.SetPixel(x, y, copy.GetPixel(newX, newY));
        }
    }
}
```

موقع ترسیم تصویر امنیتی است:

```
//-- Draw the graphic to the bitmap
gfxCaptchaImage.DrawImage(bitmap, new Point(0, 0));

gfxCaptchaImage.Flush();
```

تصویر امنیتی به صورت یک تصویر با فرمت `jpg` به صورت جریان (`stream`) به مرورگر باید فرستاده شوند:

```
HttpResponseBody response = context.HttpContext.Response;
response.ContentType = "image/jpeg";
bitmap.Save(response.OutputStream, ImageFormat.Jpeg);
```

و در نهایت حافظه‌های اشغال شده توسط اشیاء فونت و گرافیک و تصویر امنیتی آزاد خواهند شد:

```
// Clean up.
font.Dispose();
gfxCaptchaImage.Dispose();
bitmap.Dispose();
```

برای استفاده از این کدها، اکشن متدی نوشته می‌شود که نوع `CaptchaImageResult` را برگشت می‌دهد:

```
public CaptchaImageResult CaptchaImage()
{
    return new CaptchaImageResult();
}
```

اگر در یک View خصیصه src یک تصویر به آدرس این اکشن متد مقداردهی شود، آنگاه تصویر امنیتی تولید شده نمایش پیدا می‌کند:

```

```

بعد از پست کردن فرم مقدار text box تصویر امنیتی خوانده شده و با مقدار موجود در session مقایسه می‌شود، در صورتی که یکسان باشند، کاربر می‌تواند وارد سایت شود (در صورتی که نام کاربری یا کلمه عبور خود را درست وارد کرده باشد) یا اگر از این captcha در صفحات دیگری استفاده شود عمل مورد نظر می‌تواند انجام شود. در مثال زیر به طور ساده اگر کاربر در کادر متن مربوط به تصویر امنیتی مقدار درستی را وارد کرده باشد یا نه، پیغامی به او نشان داده می‌شود.

```
[HttpPost]
public ActionResult Index(LogOnModel model)
{
    if (!ModelState.IsValid) return View(model);

    if (model.CaptchaInputText == Session["captchastring"].ToString())
        TempData["message"] = "تصویر امنیتی را صحیح وارد کرده اید";
    else
        TempData["message"] = "تصویر امنیتی را اشتباه وارد کرده اید";

    return View();
}
```

کدهای کامل مربوط به این مطلب را به همراه یک مثال از لینک زیر دریافت نمایید:

[MVC-Persian-Captcha.zip](#)

نظرات خوانندگان

نویسنده:

وحید نصیری

تاریخ:

۱۹:۳۶ ۱۳۹۲/۰۳/۰۳

ممنون. میشه قسمت بررسی نهایی در اکشن متد رو هم کپسوله کرد (چیزی شبیه به امکانات AOP سرخود در MVC). مثلاً یک ویژگی جدید به نام ValidateCaptcha درست کرد که به اکشن متد اعمال شود و کار بررسی صحت اطلاعات ورودی مخصوص Captcha رو انجام و نهایتاً اطلاعات ModelState رو بر اساس اطلاعات ورودی و Session ایی که در اینجا تعریف شده، به روز کنه:

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = false)]
public sealed class ValidateCaptchaAttribute : ActionFilterAttribute
{
    public override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        var controllerBase = filterContext.Controller;

        var captchaInputTextProvider = controllerBase.ValueProvider.GetValue("CaptchaInputText");
        if (captchaInputTextProvider == null)
        {
            controllerBase.ViewData.ModelState.AddModelError("CaptchaInputText", "لطفا تصویر امنیتی را وارد کنید");
            base.OnActionExecuting(filterContext);
            return;
        }
        var inputText = captchaInputTextProvider.AttemptedValue;

        if (inputText != Session["captchastring"].ToString())
            controllerBase.ViewData.ModelState.AddModelError("CaptchaInputText", "تصویر امنیتی را اشتباه وارد کرده اید");
    }
}
```

به این صورت (با استفاده از ویژگی فوق) همان بررسی متداول ModelState.IsValid در یک اکشن متد کافی خواهد بود.

نویسنده:

پژمان پارسائی

تاریخ:

۰:۲۷ ۱۳۹۲/۰۳/۰۴

بسیار عالی و سازنده، خیلی نکته مفیدی بود

نویسنده:

میثم خوشقدم

تاریخ:

۱۳:۵۷ ۱۳۹۲/۰۳/۰۴

ممنون - مطلب کاربردی و مفیدی بود

شخصاً این Captcha را به انواع دیگر ترجیح میدم چرا که:

1- شخصی سازی شده

2- ارجاع به سایت واسطی نداره

3- هیچ گونه سرباری نداره

4- اعمال هر گونه تغییر و اصلاح مناسب با فرهنگ و کشور و یا حتی نوع سایت قابل انجام است

بازهم تشکر می‌کنم

نویسنده: میثم خوشقدم
تاریخ: ۱۵:۱۷ ۱۳۹۲/۰۳/۰۴

سلام

اگر کد شما را گسترش دهیم و به جای پیام‌های خطای ارسال شده، مقادیر Attribute ها یا متادیتاهایی مثل Required را خواند و مستقیم پیغامی به کنترل و ModelState پاس نداد، فکر می‌کنم کامل‌تر باشه .

نویسنده: وحید نصیری
تاریخ: ۱۶:۳۸ ۱۳۹۲/۰۳/۰۴

حالت کاملتر زمانی است که در سازنده کلاس ویژگی، این خطاها رو دریافت کنید و به کاربر اجازه بدید، پیغام‌های دلخواهی رو تنظیم کنه. چون Required فقط یک حالت است. حالت بعدی عدم تساوی مقدار ورودی با مقدار اصلی Captcha است که نیاز به پیغام دومی دارد.

نویسنده: پژمان پارسائی
تاریخ: ۸:۴۱ ۱۳۹۲/۰۳/۰۵

خواهش می‌کنم دوست عزیز. لطف دارید

موارد بیشتری که می‌توان به آن افزود:
استفاده از فونت‌ها به صورت اتفاقی برای نوشتن متن تصویر امنیتی،
دادن رنگ‌ها به متن و پس زمینه به صورت اتفاقی به تصویر امنیتی،
معتبر نبودن session نگهدارنده عدد تصویر امنیتی بعد از یک میزان مشخص،
و غیره

نویسنده: حسینی
تاریخ: ۱۹:۴ ۱۳۹۲/۰۳/۰۶

سلام و مرسی از آموزش خوب و کاربردیتون
اما یکی از گزینه‌های چک لیست برنامه‌های ASP.NET MVC که آقای نصیری فرمودند عدم استفاده از سشن در برنامه‌ها بود و استفاده از کوکی‌های رمزنگاری شده پیشنهاد شده بود.

نویسنده: پژمان پارسائی
تاریخ: ۲۱:۲ ۱۳۹۲/۰۳/۰۶

سلام. ممنون از نظر شما
چک لیست رو مطالعه کردم و عملاً هم استفاده کردم.
استفاده از سشن راحت‌تر هست و به همین دلیل اون رو انتخاب کردم. نسخه بعدی این تصویر امنیتی رو با استفاده از نظرات همین مطلب ارائه می‌کنم.

نویسنده: پژمان پارسائی
تاریخ: ۶:۲۸ ۱۳۹۲/۰۳/۱۵

سلام

نسخه بعدی این تصویر امنیتی رو از لینک زیر دریافت نمائید:

[MVCPersianCaptcha-2.zip](#)

امکاناتی که اضافه کردم:

- استفاده از کوکی رمزنگاری شده جهت ذخیره کردن مقدار عدد معادل تصویر امنیتی
- اضافه کردن ویژگی ValidateCaptcha جهت تعیین اعتبار کوکی و مقداری که کاربر وارد کرده
- اضافه کردن نویزهای اتفاقی
- تعیین یک میزان 30 ثانیه ای (قابل تغییر است) جهت معتبر بودن مقدار ارسالی توسط کاربر
- ایجاد قابلیت تازه سازی (refresh) تصویر امنیتی
- تغییر کلید رمزنگاری و رمزگشایی اطلاعات به ازاء هر روز
- و غیره

نویسنده: پیمان پارسائی
تاریخ: ۲۲:۵۲ ۱۳۹۲/۰۳/۲۱

پوشه‌های bin, packages, و obj از نسخه‌های اول و دوم این تصویر امنیتی حذف شده اند تا حجم فایل‌های مربوطه کمتر شوند. به همین دلیل، بعد از اجرای هر کدام از این برنامه‌ها خطایی صادر می‌شود مبنی بر این که بسته ای به اسم Newtonsoft.Json در پروژه وجود ندارد. لطفاً برای حل این مشکل به فایل Global.asax.cs مراجعه کنید و خط کد زیر را از آن حذف کنید: (این خط کد در این پروژه غیر ضروری است و نیازی به آن نیست)

```
WebApiConfig.Register(GlobalConfiguration.Configuration);
```

روش دیگر برای حل این نوع مشکلات، در مطلب [بازسازی کامل پوشه packages بسته‌های NuGet به صورت خودکار](#) بیان شده است.

نویسنده: بهمن خلفی
تاریخ: ۱۰:۳۴ ۱۳۹۲/۰۴/۱۱

با سلام و تشکر از همه دوستان
بنده در اولین استفاده یک تغییر داشتم که خواستم بقیه هم از آن استفاده کنند:

```
public ValidateCaptchaAttribute()
{
    ErrorWasHappened = "خطایی اتفاق افتاده است";
    CaptchaCodeIsRequired = "لطفاً کد امنیتی را وارد کنید";
    CaptchaCodeIsIncorrect = "کد امنیتی را اشتباه وارد کرده اید";
    CookieMustEnabled = "باید ابتدا قابلیت کوکی‌ها را در مرورگر خود فعال کنید";
    ExpireTimeCaptchaCodeBySeconds = 60;
    TimeIsExpired = string.Format("{0} ثانیه", ExpireTimeCaptchaCodeBySeconds.ToString());
    "است", ExpireTimeCaptchaCodeBySeconds.ToString());
}
```

باز ممنون!

نویسنده: ایلیا اکبری فرد
تاریخ: ۲۱:۳۸ ۱۳۹۲/۰۴/۱۴

با سلام.
سپاس از مقاله خوبتون.
در کلاینت همه چی درست کار میکند ولی وقتی آنرا درون هاست واقعی publish میکنم، تصویر نمایش داده نمی‌شود و خطایی که توسط elmah لاگ می‌شود به صورت زیر است:

```
The system cannot find the file specified " source="mscorlib"
detail="System.Security.Cryptography.CryptographicException: The system cannot find the file specified.
```

با تشکر.

نویسنده: محسن خان
تاریخ: ۲۳:۲ ۱۳۹۲/۰۴/۱۴

```
<system.web>  
<trust level="Medium" originUrl=".*" />
```

شاید هاست شما مدیوم تراست هست. برای تست روی لوکال این تنظیم بالا رو به وب کانفیگ اضافه کنید تا خطا رو بتونید لوکال دیباگ کنید.

نویسنده: ایلیا اکبری فرد
تاریخ: ۲۳:۱۶ ۱۳۹۲/۰۴/۱۴

سپاس از پاسخ شما.
ولی با اضافه کردن آن ، همواره خطای زیر رخ می‌دهد:

```
The application attempted to perform an operation not allowed by the security policy. To grant this application the required permission please contact your system administrator or change the application's trust level in the configuration file
```

نویسنده: پژمان پارسائی
تاریخ: ۶:۷ ۱۳۹۲/۰۴/۱۵

سلام، از نظر شما متشکرم

همون طور که آقای محسن خان گفت، احتمالا هاست شما medium trust هست. اما رو کامپیوتر خودتون full trust برنامه نویسی می‌کنید.

چون هویت کاربر هنوز مشخص نشده پیغامی مبنی بر این لاگ میشه که اسمبلی mscorlib وجود نداره. در واقع وجود داره ولی نه برای کاربر anonymous ! همچنین این به دلیل medium trust بودن هم میتونه باشه. برای حل این مشکل کارهای زیر رو انجام بدین:

1) به فایل web.config برین و کد زیر رو اضافه کنید:

```
<trust level="Full" originUrl=".*" />
```

2) باید تغییری رو در متدهای الحاقی Encrypt و Decrypt بدین، که از این متدها برای رمزنگاری و رمزگشایی محتوای کوکی تصویر امنیتی استفاده میشه. قبل از هر کدوم از این متدها flag زیر رو اضافه کنید:

```
[System.Security.Permissions.PermissionSet(System.Security.Permissions.SecurityAction.Assert,  
Unrestricted = true)]
```

همچنین، یک خط داخل بدنه هر کدوم از متدهای الحاقی Encrypt و Decrypt هست، منظورم این خط کد هست:

```
var cspp = new CspParameters { KeyContainerName = key };
```

که باید به خط کد زیر تبدیل بشه:

```
var cspp = new CspParameters { KeyContainerName = key, Flags = CspProviderFlags.UseMachineKeyStore };
```

نویسنده: ایلیا اکبری فرد
تاریخ: ۹۰:۷ ۱۳۹۲/۰۴/۱۵

بله کار کرد. متشکرم.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۸:۲۹ ۱۳۹۲/۰۴/۲۹

با سلام.

برای اولین بار که فرمی لود می‌شود همواره حتی با وارد کردن مقدار صحیح متن کپچا باز هم خطای نادرست بودن می‌دهد. ولی وقتی یکبار بر روی رفرش مربوط به کپچا کلیک میکنم تا کپچای جدیدی تولید کند ، درست کار می‌کند؟

نویسنده: پژمان پارسائی
تاریخ: ۲۱:۵۱ ۱۳۹۲/۰۵/۰۱

سلام دوست عزیز

بابت تاخیر بوجود آمده در پاسخ دهی عذرخواهی می‌کنم.

من همین کپچا رو دارم استفاده می‌کنم و این مشکلو هم باهاش ندارم. یک حدسی که می‌زنم اینه که شاید کوکی بار اول درست مقداردهی نشده. لطفا کل عملیات از تشکیل تصویر امنیتی، تا مقداری که درون کوکی قرار می‌گیره، و بعد رمز گزاری میشه ، تا مقداری که درون کوکی مرورگر ذخیره میشه (برای مشاهده کوکی‌های مرورگر از [این روش](#) می‌تونید استفاده کنید) . سپس ببینید بعد از پست شدن فرم حاوی کپچا به سرور مقداری که از درون کوکی خوانده میشه و می‌خواد decrypt بشه با مقداری که از اول ذخیره شده یکسان باشه.

امیدوارم مشکلتون حل شه دوست عزیز.

نویسنده: کوروش ت
تاریخ: ۱۰:۳ ۱۳۹۲/۰۹/۲۸

باسلام، این مطلب خیلی کاربردی و خوب بود، فقط نکته ای که در فرم Register پیش فرض MVC وجود داره، اینکه این View به کنترلر Account وصله که هنگامی که CaptchaImageResult را درون آن قرار می‌دهیم، در فرم Register اجرا نمی‌شود. می‌خواستم راهنمایی کنید که در این فرم چگونه باید عمل کنم. مرسی!

نویسنده: محسن خان
تاریخ: ۱۰:۹ ۱۳۹۲/۰۹/۲۸

ویژگی AllowAnonymous را بالای آن قرار دهید. همچنین Url.Action نام کنترلر و نام اکشن متد را دریافت می‌کند (برای استفاده از آن در سایر Viewهایی که در کنترلر جاری نیستند) .

نویسنده: کوروش ت
تاریخ: ۹:۱۳ ۱۳۹۲/۱۰/۱۵

با تشکر از راهنمایی شما، این مورد به خوبی در حالت localhost کار می‌کند و تغییر تصویر نیز به خوبی عمل می‌کند. هنگامی که سایت را Publish میکنم، تصاویر نمایش داده نمی‌شود. ممنون میشم راهنمایی کنید.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۵ ۹:۵۷

برای دیباگ کردن این مساله در سمت سرور بهتر است از [ELMAH](#) استفاده کنید. لاگ آن را بررسی کنید شاید جایی در پشت صحنه خطایی صادر می‌شود.

نویسنده: کوروش ت
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۰:۲۷

با سلام، Elmah Log رو چک کردم، مشکل دسترسی داشت که در قسمت Web.config تگ Trust رو اضافه کردم و فعلاً ظاهر شد. اضافه کردن trust مشکل امنیتی ایجاد نمی‌کند؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۰:۳۶

دوم قسمتش مشکل دسترسی داشت؟ چه خطایی گرفتید دقیقاً؟ چه trust ایی رو اضافه کردید؟ (در هاستی که بتونید trust level رو تغییر بدید، یعنی عملاً از لحاظ امنیتی درست تنظیم نشده و مداخل رو قفل نکرده)

نویسنده: کوروش ت
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۰:۴۸

پیغام:

The system cannot find the file specified " source=" mscorlib "

کد تراست

```
< trust level = "Medium" originUrl = ".*" />
```

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۱:۳۷

عجیبه که کار کرده. [چون با حالت medium مشکل داره](#) .

نویسنده: کوروش ت
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۳:۰۰

هنوز کامل باهاش سر و کله نزدم، ولی یک ثبت نام کردم دیدم داره درست کار می‌کنه. پس خطری نداره این کد.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۴:۵۶

trust level یعنی کد شما مثلاً نتونه دایرکتوری فونت ویندوز رو روی سرور لیست کنه یا نتونه به کلیدهای رمزنگاری سطح ماشین دسترسی پیدا کنه. از این دید باید به آن نگاه کرد نه از دید کد خطرناک. این پروژه که سورش باز هست و می‌تونید کدهای آن را بررسی کنید.

نویسنده: کوروش ت
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۵:۲۶

منظور من این نبود، منظورم خطری برای سایر Third Party API ها بود که دادن تراست لول خطری برای ای پی آی های دیگه نداشته باشه. در مورد این کد هیچ چیز مبهمی وجود نداره و هم توضیحات خوب داده شده بود و هم از متدهای رادست و قابل فهم استفاده شده بود. از راهنمایی های شما هم بسیار ممنونم.

نویسنده: ایلیا

تاریخ: ۱۶:۲۰ ۱۳۹۲/۱۰/۱۵

با سلام.

من از این کپچا در یک سرور استفاده کردم، بر روی پورت 80 هیچ خطایی نمیگیرم و تصویر امنیتی بدرستی نمایش داده می شود ولی روی پورت 8080 تصویر کپچا لود نمیشود و خطای زیر را در کنسول کروم لاگ میکند:

```
System.Security.Cryptography.CryptographicException: Object already exists
```

```
Resource interpreted as Image but transferred with MIME type text/html:  
"http://site:8080/Error/Index?aspxerrorpath=/Shared/CaptchaImage"
```

با تشکر.

نویسنده: محسن خان

تاریخ: ۱۷:۴۲ ۱۳۹۲/۱۰/۱۵

اگر به سرور دسترسی دارید [این دستور](#) را اجرا کنید:

```
aspnet_regiis -pa "SampleKeys" "NT AUTHORITY\NETWORK SERVICE"
```

نویسنده: محسن

تاریخ: ۵:۳۵ ۱۳۹۳/۰۱/۰۷

من در یک پروژه می خواستم از این Captcha استفاده کنم. زمانی که صفحه اصلی لود میشه Captcha درست نمایش داده میشه ولی زمانی که کاربر از سایت SignOut می کنه دیگه محتوای Captcha نمایش داده نمی شود و اصلاً کنترلر CaptchaImage فراخوانی نمی شود ممنون میشم راهنمایی کنید

نویسنده: وحید نصیری

تاریخ: ۱۱:۴۶ ۱۳۹۳/۰۱/۰۷

اگر فراخوانی نمی شود، یعنی کش شده. برای رفع این مشکل، ویژگی زیر را بر روی اکشن متد CaptchaImage قرار دهید:

```
[OutputCache(Location = OutputCacheLocation.None, NoStore = true)]
```

برای بررسی دقیق تر از [افزونه ی فایرباگ](#) استفاده کنید.

نویسنده: فواد عبداللهی

تاریخ: ۱۱:۴۸ ۱۳۹۳/۰۵/۱۰

سلام

میدونم پست نسبتا قدیمی!

ولی من یه مورد پیدا کردم و اونم اینه که در صورتی که captcha رو درست بزنی و برید به صفحه جدید (مثلا بعد لاگین برید به صفحه مدیر) دوباره back بزنی همون مقدار رو می بینید و متاسفانه دوباره کار هم میکنه.

کش رو هم غیر فعال کردم ولی تاثیری نداشت!

اگر راه حلی دارد ممنون میشم در میان بگذارید.

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۵/۱۰ ۱۲:۷

نکته‌ی [غیرفعال کردن کش مرورگر در MVC](#) رو تست کردید؟

نویسنده: فواد عبداللهی
تاریخ: ۱۳۹۳/۰۵/۱۰ ۱۲:۱۲

بله جواب نداد!

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۱۰ ۱۳:۲۴

1- فایل CaptchaImageResult.cs [نگارش دوم](#) را باز کنید.

- در انتهای آن تغییر زیر را اعمال کنید:

```
// فایل در CaptchaImageResult.cs
HttpResponseBase response = context.HttpContext.Response;
response.ContentType = "image/jpeg";
context.HttpContext.DisableBrowserCache(); // این سطر جدید است
bitmap.Save(response.OutputStream, ImageFormat.Jpeg);
```

متد الحاقی DisableBrowserCache در مطلب [غیرفعال کردن کش مرورگر در MVC](#) ارائه شده.

2- امضای متد Index کنترل Home نیاز به NoBrowserCache دارد (کل صفحه‌ی لاگین کش نشود):

```
public class HomeController : Controller
{
    [NoBrowserCache]
    public ActionResult Index()
```

3- امضای متد CaptchaImage را در کنترلر Home به نحو زیر تغییر دهید (آدرس خاص تصویر نمایش داده شده، کش نشود):

```
[NoBrowserCache]
[OutputCache(Location = OutputCacheLocation.None, NoStore = true, Duration = 0, VaryByParam = "None")]
public CaptchaImageResult CaptchaImage(string rndDate)
```

سپس در محل استفاده در View به صورت زیر باید استفاده شود:

```

```

نویسنده: افشار محبی
تاریخ: ۱۳۹۳/۰۶/۲۵ ۱۰:۸

من هم همین مشکل را داشتم. قبل از این که توضیحات اینجا را ببینم، msdn را دیده بودم:

<http://social.msdn.microsoft.com/Forums/vstudio/en-US/7ea48fd0-8d6b-43ed-b272-1a0249ae490f/systemsecuritycryptographiccryptographicexception-the-system-cannot-find-the-file-specified?forum=clr>

روشی که آنجا گفته بود باید application pool را از integrated به classic می‌پردی و از network service برای identity

استفاده می‌کردی. هر چند که این روش هم جواب داد ولی روش توضیح داده اینجا، یعنی اصلاح کد خیلی بهتر است. البته در مورد من نیاز به اضافه کردن trust level به web.config نبود.

نویسنده: افشار محبی

تاریخ: ۱۳۹۳/۰۶/۲۵ ۱۰:۱۰

کد خیلی مفید و خوبی است. کاش آن را در github یا codeplex هم قرار می‌دادید تا همه به روز رسانی‌ها و pull request های احتمالی از همانجا انجام می‌شد. در هر صورت بابت این کد مفید تشکر می‌کنم.

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 4+3). این تصویر خروجی این اکشن متد است. به سادگی می‌توانید این اکشن را بر اساس خواسته خود اصلاح کنید؛ مثلاً 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 = "{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)
    
</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) {
                $("#imgcaptcha").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، بررسی برابری 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). بدیهی است اگر این دو مقدار نابرابر باشند، یک خطا به ModelState اضافه می شود 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();
    $('#imgcaptcha').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: ۹:۳۳ ۱۳۹۳/۰۳/۲۶

ممنون از شما. فقط یک نکته‌ی کوچک در مورد memory stream هست که بهتره در نظر گرفته بشه. در این شیء متدهای ToArray و GetBuffer یکی نیستند. متد [GetBuffer](#) حجمی نزدیک به 2 برابر آرایه اصلی رو عموماً داره و انتهایش 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.