عنوان: **CAPTCHAfa**
نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۵/۱۵ ۱۲:۴۶
آدرس: www.dotnettips.info
گروه‌ها: 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، پروژه ای 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 گونه ای تغییر کنه که برای دیجیتالی کردن متن‌های پارسی استفاده بشه. در حال حاضر، این پروژه از کلماتی از پیش تعریف شده استفاده می‌کنه.

متاسفانه 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 اسکریپ 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 ViewResult 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: ۱۳۹۱/۰۵/۱۵ ۱۵:۹

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 فاضله اهمیت نداره! ... چرا اینجوریه؟!

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

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

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

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 ری‌کپتچا خواسته کار کنه اما خب چیز خوبی از کار درنیامده

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

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: ۱۳۹۱/۰۵/۱۵ ۱۵:۵۳

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: ۱۳۹۱/۰۵/۱۵ ۱۹:۵۳

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

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

با سلام. جهت حل مشکل کلمات از پیش تعریف شده، می‌توان از Google Translate استفاده کرد.
متاسفانه 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 بیضی هایی 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: ۱۳۹۱/۰۵/۱۸ ۱۱:۱۱

"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 دیجتالی کردن نیست

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۵/۱۸ ۱۲:۴

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 CAPTCHAیی 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: ۱۳۹۱/۰۵/۲۸ ۱۲:۵۰

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.

نویسنده: بابک
تاریخ: ۱۳۹۳/۰۸/۲۸ ۱۵:۲۳

آدرس ظاهرا به آدرس زیر تغییر کرده است:

http://www.captchafa.net

نویسنده: محمد رضا صفری

تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۳:۱۲

فکر کنم که کلا پروژه تعطیل شده ...