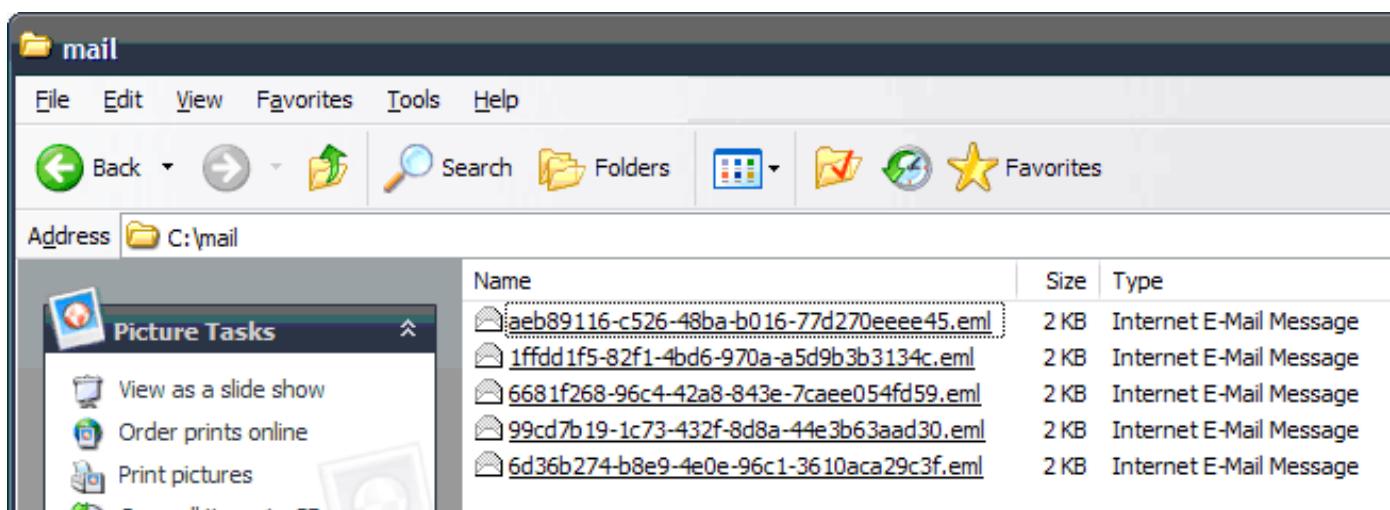


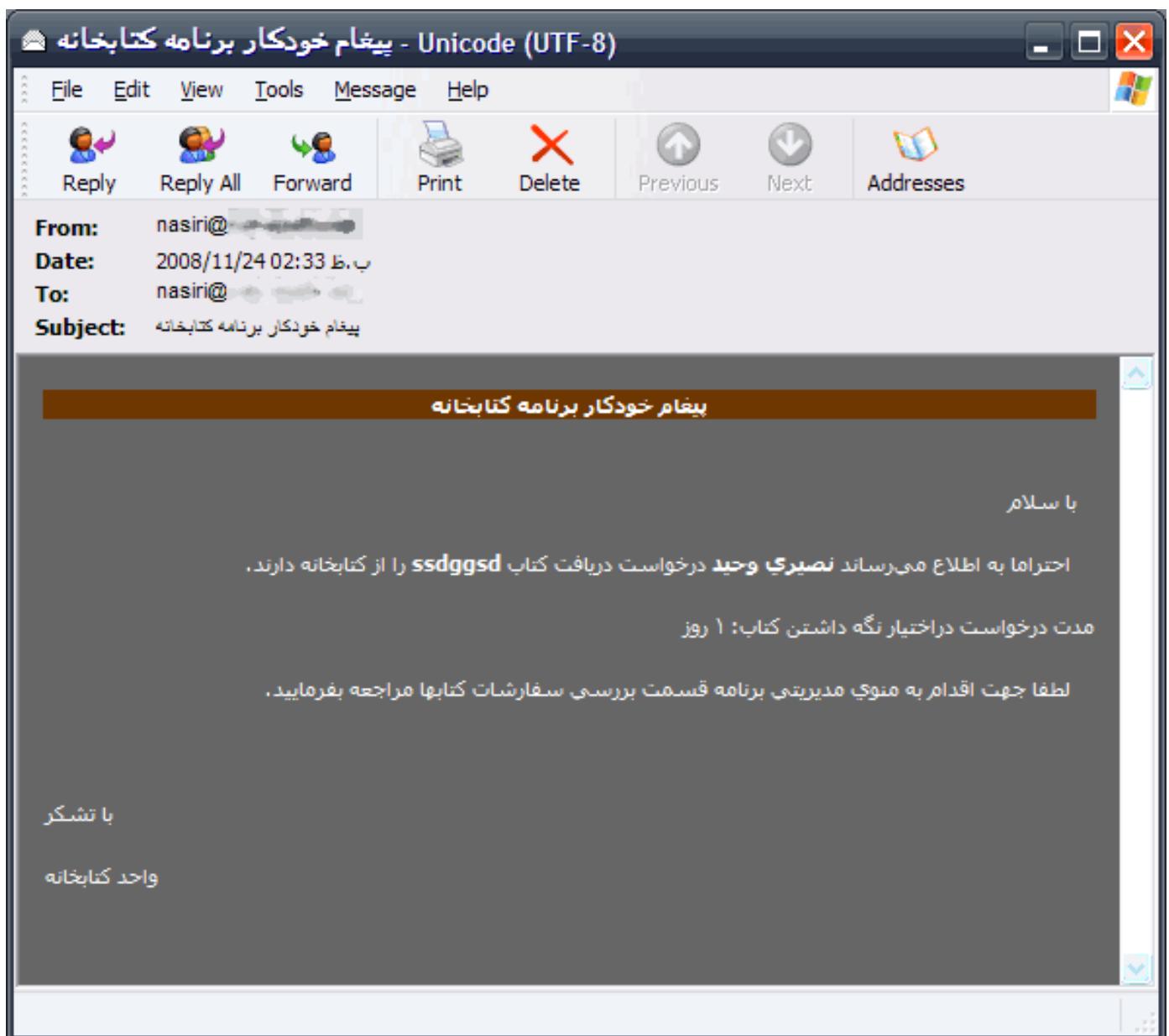
فرض کنید مشغول به کار بر روی کامپیوتری هستید که دسترسی به هیچ شبکه‌ای ندارد و همچنین نیاز است تا قسمت اطلاع رسانی برنامه ASP.NET خود را که از طریق ایمیل کار می‌کند، تست کنید. برای مثال حداقل یکبار شکل و شمايل و محتوای ایمیل واقعی ارسالی آنرا در آوت لوک مشاهده کنید. برای حل این مساله چه باید کرد؟
 برای تحقق این منظور باید کمی فایل web.config سایت را ویرایش کرد و سطرهای زیر را به آن افزود (پس از بسته شدن تگ :system.web

```

<system.net>
  <mailSettings>
    <smtp deliveryMethod="SpecifiedPickupDirectory">
      <specifiedPickupDirectory pickupDirectoryLocation="c:\mail"/>
    </smtp>
  </mailSettings>
</system.net>
    
```

و همچنین در اینجا باید دقت داشت که هنگام کد نویسی دیگر نیازی به ذکر smtp server نخواهد بود و تنظیمات خودش را از فایل کانفیگ خواهد خواند.
 اکنون با هر بار ارسال ایمیل، نتیجه حاصل (مطابق تصاویر زیر) در مسیر c:\mail ذخیره خواهد شد و فرمت حاصل با استفاده از outlook قابل مشاهده است.





شایان ذکر است که این روش با برنامه‌های غیر ASP.Net نیز کار می‌کند و تنها کافی است یک فایل app.config به برنامه اضافه کرده و تنظیمات فوق را به آن اعمال نمایید.

نظرات خوانندگان

نوبتند: **salarblog** | تاریخ: **۱۳۸۷/۰۹/۰۴ ۱۹:۱۶:۰۰**

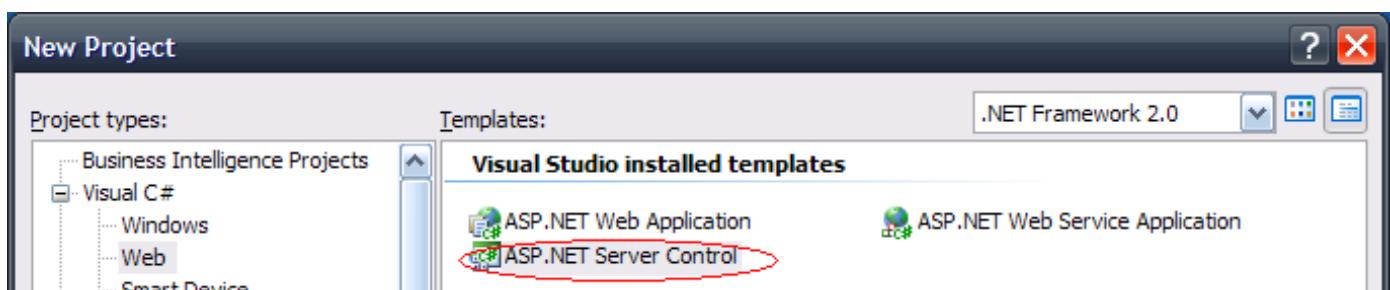
نکته جالب بود.
بسیار ممنون

امروز داشتم یک سری از پلاگین‌های jQuery را مرور می‌کردم، مورد زیر به نظرم واقعاً حرفاً ای اوMD و کمبود آن هم در بین کنترل‌های استاندارد ASP.NET محسوس است:

Masked Input Plugin

استفاده از آن به صورت معمولی بسیار ساده است. فقط کافی است اسکریپت‌های jQuery و سپس این افزونه به هدر صفحه اضافه شوند و بعد هم مطابق صفحه [usage](#) آن عمل کرد.
خیلی هم عالی! ولی این شیوه‌ی متداوول کار در ASP.NET نیست. آیا بهتر نیست این مجموعه را تبدیل به یک کنترل کنیم و از این پس به سادگی با استفاده از Toolbox ویژوال استودیو آنرا به صفحات اضافه کرده و بدون درگیر شدن با دستکاری سورس HTML صفحه، از آن استفاده کنیم؟ به عبارتی دیگر یکبار باید با جزئیات درگیر شد، آنرا بسته بندی کرد و سپس بارها از آن استفاده نمود.
(مفاهیم شیء‌گرایی)

برای این کار، یک پروژه جدید ایجاد `ASP.NET server control` (به نام `MaskedEditCtrl`) را آغاز نمائید (به نام



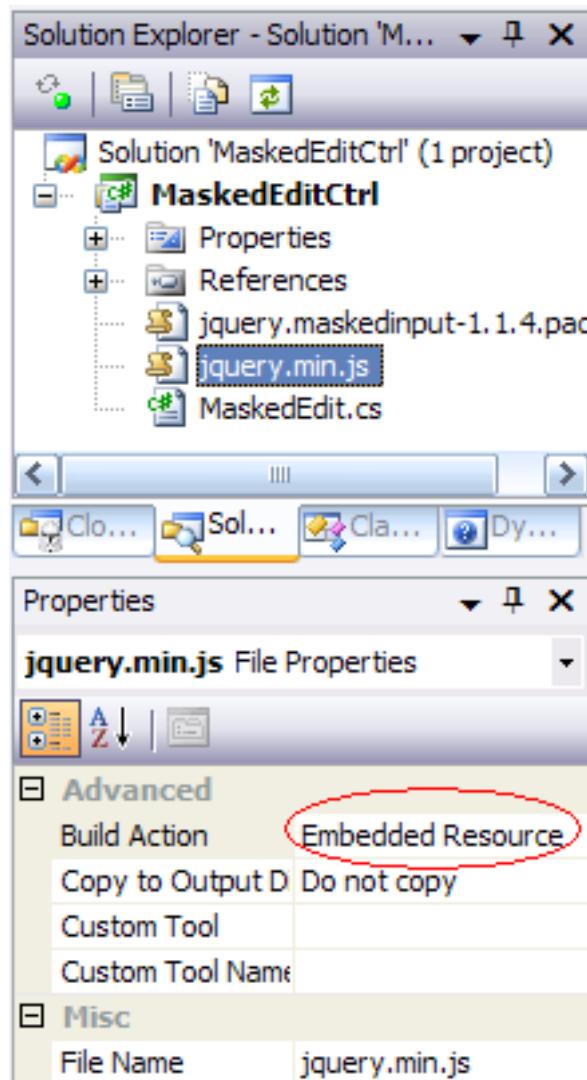
به صورت پیش فرض یک قالب استاندارد ایجاد خواهد شد که کمی نیاز به اصلاح دارد. نام کلاس را به `MaskedEdit` تغییر خواهیم داد و همچنین در قسمت `ToolboxData` نیز نام کنترل را به `MaskedEdit` ویرایش می‌کنیم.

برای اینکه مجبور نشویم یک کنترل کاملاً جدید را از صفر ایجاد کنیم، خواص و توانایی‌های اصلی این کنترل را از `TextBox` استاندارد به ارت خواهیم برد. بنابراین تا اینجای کار داریم:

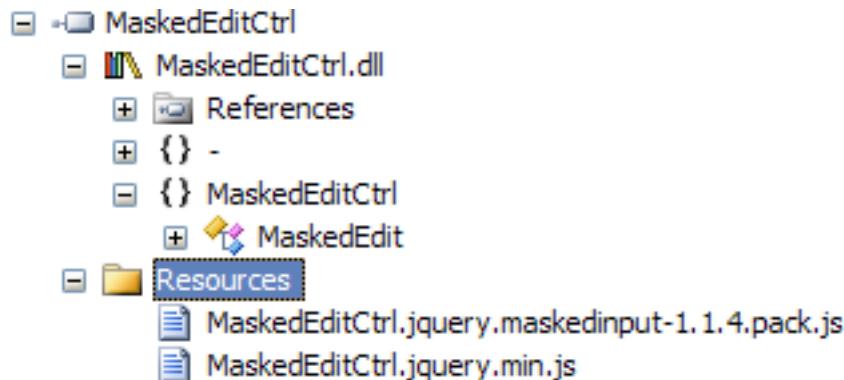
```
namespace MaskedEditCtrl
{
  [DefaultProperty("MaskFormula")]
  [ToolboxData("<{0}:MaskedEdit runat=server></{0}:MaskedEdit>")]
  [Description("MaskedEdit Control")]
  public class MaskedEdit : TextBox
  {
```

سپس باید رویداد `OnPreRender` را تحریف (override) کرده و در آن همان اعمالی را که هنگام افزودن اسکریپت‌ها به صورت دستی انجام می‌دادیم با برنامه نویسی پیاده سازی کنیم. چند نکته ریز در اینجا وجود دارد که در ادامه به آنها اشاره خواهد شد.
از ASP.NET 2.0 به بعد، امکان قرار دادن فایل‌های اسکریپت و یا تصاویر همراه یک کنترل، درون فایل `.aspx` آن بدون نیاز به توزیع مجزای آنها به صورت `WebResource` مهیا شده است. برای این منظور اسکریپت‌های jQuery و افزونه `mask edit` را به پروژه اضافه

نمائید. سپس به قسمت خواص آنها (هر دو اسکریپت) مراجعه کرده و آنها را به build action Embedded Resource تغییر دهید (شکل زیر):



از این پس با کامپایل پروژه، این فایل‌ها به صورت resource به dll ما اضافه خواهند شد. برای تست این مورد می‌توان به برنامه reflector مراجعه کرد (تصویر زیر):



پس از افزودن مقدماتی اسکریپت‌ها و تعریف آنها به صورت resource، باید آنها را در فایل AssemblyInfo.cs پروژه نیز تعریف کرد (به صورت زیر).

```
[assembly: WebResource("MaskedEditCtrl.jquery.min.js", "text/javascript")]
[assembly: WebResource("MaskedEditCtrl.jquery.maskedinput-1.1.4.pack.js", "text/javascript")]
```

نکته مهم: همانطور که ملاحظه می‌کنید نام فضای نام پروژه (namespace) باید به ابتدای اسکریپت‌های معرفی شده اضافه شود.

پس از آن نوبت به افزودن این اسکریپت‌ها به صورت خودکار در هنگام نمایش کنترل است. برای این منظور داریم:

```
protected override void OnPreRender(EventArgs e)
{
    base.OnPreRender(e);

    //adding .js files
    if (!Page.ClientScript.IsClientScriptIncludeRegistered("jquery_base"))
    {
        string scriptUrl = Page.ClientScript.GetWebResourceUrl(this.GetType(),
            "MaskedEditCtrl.jquery.min.js");
        Page.ClientScript.RegisterClientScriptInclude("jquery_base", scriptUrl);
    }

    if (!Page.ClientScript.IsClientScriptIncludeRegistered("edit_ctrl"))
    {
        string scriptUrl = Page.ClientScript.GetWebResourceUrl(this.GetType(),
            "MaskedEditCtrl.jquery.maskedinput-1.1.4.pack.js");
        Page.ClientScript.RegisterClientScriptInclude("edit_ctrl", scriptUrl);
    }

    if (!Page.ClientScript.IsStartupScriptRegistered("MaskStartup" + this.ID))
    {
        // Form the script to be registered at client side.
        StringBuilder sbStartupScript = new StringBuilder();
        sbStartupScript.AppendLine("jQuery(function($){");
        sbStartupScript.AppendLine("$.("#" + this.ClientID + ").mask(\"" + MaskFormula +
"\"');");
        sbStartupScript.AppendLine("});");
        Page.ClientScript.RegisterStartupScript(typeof(Page),
            "MaskStartup" + this.ID, sbStartupScript.ToString(), true);
    }
}
```

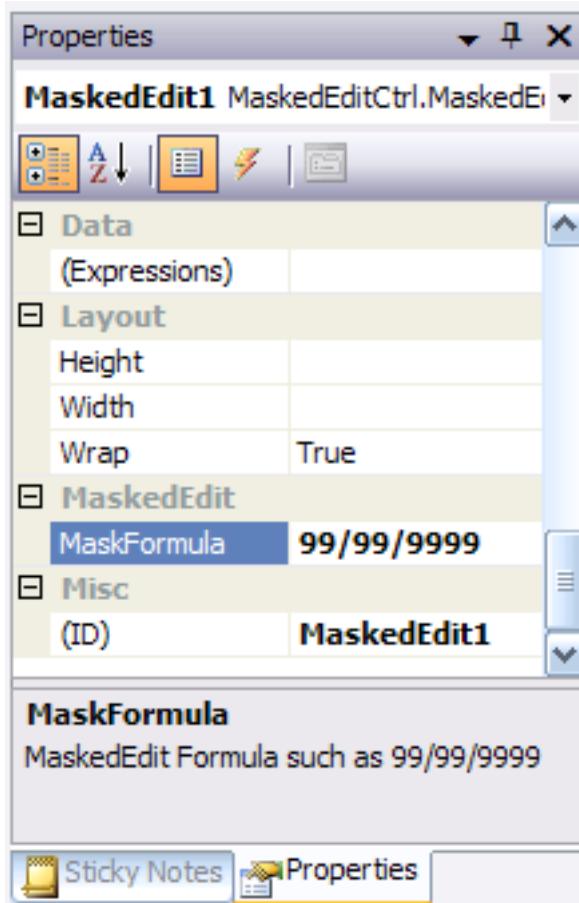
همانطور که ملاحظه می‌کنید، ابتدا WebResource دریافت شده و سپس به صفحه اضافه می‌شود. در آخر مطابق راهنمای افزونه mask edit عمل شد. یعنی اسکریپت مورد نظر را ساخته و به صفحه اضافه کردیم.

نکته جواوا اسکریپتی: علت استفاده از this.ClientID جهت معرفی نام کنترل جاری این است که هنگامیکه کنترل توسعه یک master page رندر شود، آن توسعه موتور ASP.Net کمی تغییر خواهد کرد. برای مثال myTextBox به this.ID ذکر شده شد و اگر صرفاً ct100_ContentPlaceHolder1_myTextBox کدهای جواوا اسکریپت مقدور نخواهد بود. بنابراین از ClientID نهایی رندر شده توسعه ASP.Net کمک می‌گیریم.

در اینجا MaskFormula مقداری است که هنگام افزودن کنترل به صفحه می‌توان تعریف کرد.

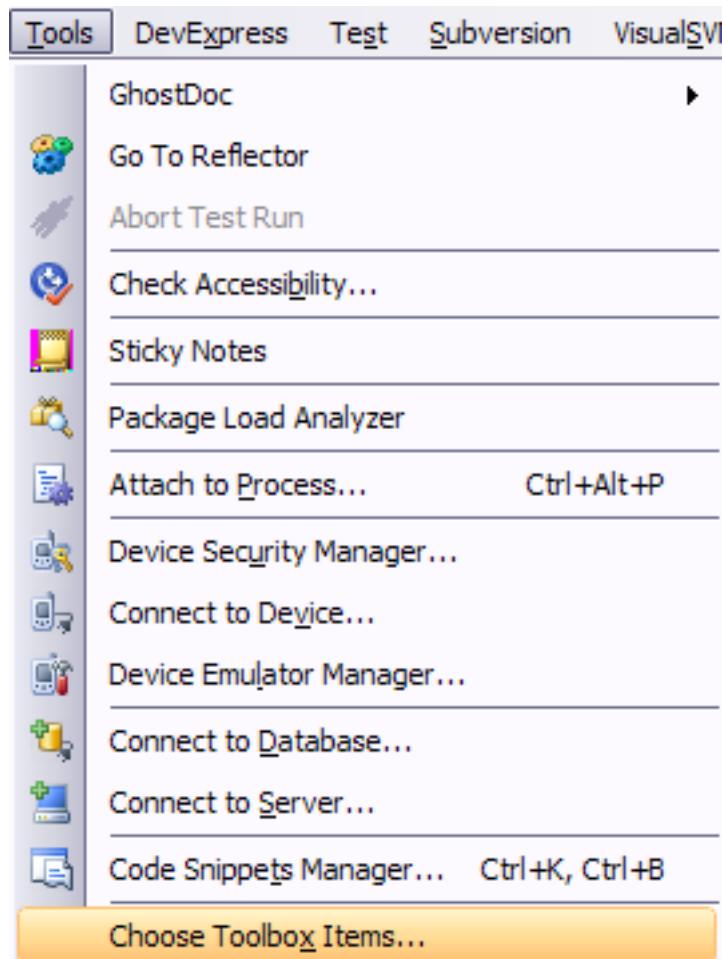
```
[Description("MaskedEdit Formula such as 99/99/9999")]
[Bindable(true), Category("MaskedEdit"), DefaultValue(0)]
public string MaskFormula
{
    get
    {
        if (ViewState["MaskFormula"] == null) ViewState["MaskFormula"] = "99/99/9999";
        return (string)ViewState["MaskFormula"];
    }
    set { ViewState["MaskFormula"] = value; }
}
```

این خاصیت public هنگام نمایش در Visual studio به شکل زیر درخواهد آمد:



نکته مهم: در اینجا حتماً باید از view state جهت نگهداری مقدار این خاصیت استفاده کرد تا در حین post back ها مقادیر

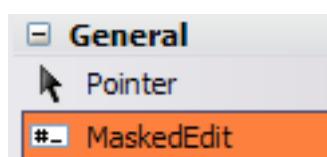
اکنون پروژه را کامپایل کنید. برای افزودن کنترل ایجاد شده به toolbox می‌توان مطابق تصویر عمل کرد:



نکته: برای افزودن آیکون به کنترل (جهت نمایش در نوار ابزار) باید: (الف) تصویر مورد نظر از نوع bmp باشد با اندازه 16x16 pixel . (ب) باید آنرا به پروژه افزود و آن را به build action Embedded Resource تغییر داد. سپس آنرا در فایل AssemblyInfo.cs پروژه نیز تعریف کرد (به صورت زیر).

```
[assembly: System.Web.UI.WebControls.WebResource("MaskedEditCtrl.MaskedEdit.bmp", "img/bmp")]
```

کنترل ما پس از افزوده شدن، شکل زیر را خواهد داشت:



جهت دریافت سورس کامل و فایل بایناری این کنترل، [اینجا](#) کلیک کنید.

نظرات خوانندگان

نویسنده: شاهین کایست
تاریخ: ۱۳۸۹/۱۰/۲۳ ۲۲:۳۹:۳۰

سلام.

از کنترلی که طراحی کردید درون یک UI دیالوگ استفاده کردم (درون محتويات Dialog در یک Update Panel هست).
اما پس از قرار دادن کنترل Dialog از کار افتاد.

نکته: Dialog را از سمت Server پس از Postback اجرا کردم.

به نظرتون مشکل از کجا هست؟

ممnon

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۰/۲۳ ۲۲:۵۷:۴۱

سلام،

چند مورد هست:

- یکی اینکه بهتر است نسخه‌ی جدید jQuery را به این سورس اضافه و کامپایل کنید.
- مورد دیگر آشنایی با jQuery Live باشد و موردنی: [\(+\)](#) است: که پس از postback، نیاز به تزریق یا بایند مجدد یک سری اطلاعات می‌باشد و

همچنین: [\(+\)](#)

اگر مطلب تبدیل پلاگین‌های جی‌کوئری به کنترل‌های ASP.NET را مطالعه کرده باشید، در مورد ClientID بحث شد. با مراجعه به اصول HTML درخواهیم یافت که هر کنترل یا شیء مربوطه می‌تواند شامل ID و name باشد. عموماً در کدهای جاوا اسکریپتی برای دسترسی به یک شیء در صفحه از ID آن شیء به صورت document.getElementById استفاده شده و از name برای پردازش‌های سمت سرور استفاده می‌شود. برای مثال اگر به دوران ASP کلاسیک برگردیم، از شیء Request برای دریافت مقادیر ارسال شده به سرور با استفاده از name شیء مورد نظر استفاده می‌شود.

در حالت فرم‌های معمولی، ID و name عموماً یکسان هستند. اما اگر از master page ها استفاده شود یا از یک user control برای ترکیب چندین کنترل کمک گرفته شود، دیگر ID و name در فرم رندر شده نهایی ASP.NET یکسان نخواهند بود. برای مثال:

```
<input name="ctl00$ContentPlaceHolder1$txtID" type="text" id="ctl00_ContentPlaceHolder1_txtID" />
```

اگر به سورس دات نت فریم ورک مراجعه کنیم علت وجود _ بجای \$ را در مقدار id می‌توان مشاهده کرد:

```
public virtual string ClientID
{
    get
    {
        this.EnsureID();
        string uniqueID = this.UniqueID;
        if ((uniqueID != null) && (uniqueID.IndexOf(this.IdSeparator) >= 0))
        {
            return uniqueID.Replace(this.IdSeparator, '_');
        }
        return uniqueID;
    }
}
```

وجهت تولید name به صورت زیر عمل می‌شود (یک الگوریتم بازگشتی است):

```
<Container>$<Container>$<Container>$<Container>$...$<ID>
```

زمانیکه یک کنترل توسط ASP.NET رندر می‌شود، خاصیت UniqueID معادل name آن و ClientID آن کنترل در سمت کلاینت خواهد بود. اگر از IE استفاده کنید به هر نحوی سعی در پردازش اسکریپت شما خواهد کرد و document.getElementById بالاخره جواب خواهد داد. اما اگر از سایر مرورگرها استفاده کنید، در صورتیکه بجای ID از name استفاده شده باشد، اسکریپت شما با پیغامی مبنی بر یافت نشدن شیء مورد نظر متوقف خواهد شد (چون مقدار این دو همانطور که ملاحظه کردید الزاماً یکسان خواهد بود).

این نکته در طراحی کنترل‌های ASP.NET که از کدهای جاوا اسکریپتی استفاده می‌کنند بسیار مهم است. در تست اول و با یک صفحه ساده کنترل شما خوب کار خواهد کرد. اما اگر همین کنترل را بر روی یک صفحه مشتق شده از یک master page قرار دهید، دیگر ID آن یافت نشده و کدهای جاوا اسکریپتی شما کار نخواهند کرد. به همین جهت اگر قرار است قسمت جاوا اسکریپتی کنترل شما توسط کنترل به صورت خودکار ایجاد شود و در این کد ارجاعی به شیء جاری وجود دارد، این شیء جاری با استفاده از آن در سمت کلاینت قابل دسترسی خواهد بود و نه با استفاده از ID سمت سرور و یا UniqueID آن.

نظرات خوانندگان

نویسنده: hajloo
تاریخ: ۱۳۸۷/۰۹/۰۹ ۱۴:۲۴:۰۰

از query که برای جمع ساعت برای من نوشته بود و ارسال کردید واقعاً ممنونم . منظور من دقیقاً همان کاری بود که شما انجام دادید .
صمیمانه از وقتی که برای نوشتمن کوئری صرف کردید سپاسگزارم . پاینده و سربلند باشید ناصر حاجلو.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۰۹/۰۹ ۱۴:۲۷:۰۰

خواهش می‌کنم . خوشحالم که مفید واقع شد .
موفق و مؤید باشید .

سایت [نستعلیق آنلاین](#) با استفاده از ASP.NET و فونت [ایران نستعلیق](#) ایجاد شده است. شاید این سؤال پیش بیاید که چگونه اینکار را انجام داده‌اند؟ چگونه متن را به تصویر تبدیل کرده‌اند یا از همه مهم‌تر چگونه فونت را به صورت پویا بارگذاری می‌کنند (چون عموماً هاستها فونتی را برای شما نصب نخواهند کرد)؟

برای انجام اینکار از کلاس [System.Drawing.Text.PrivateFontCollection](#) فضای نام `PrivateFontCollection` می‌توان استفاده کرد. نحوه انجام این کار را در یکی از پروژه‌های سایت [codeproject](#) می‌توان ملاحظه نمود.

کمی این پروژه را اصلاح کردم به همراه افزودن و تنظیم خواص تولید تصویر با کیفیت بالا. پروژه نهایی را به همراه قلم‌های مربوطه، از [اینجا](#) می‌توانید دریافت کنید.



بارگذاری قلم از فایل بدون نصب آن

احتمالاً در سایت [نستعلیق آنلاین](#) از روش موجود در مقاله بالا استفاده شده است که مجبور شده‌اند تصویر نهایی را در یک صفحه دیگر نمایش دهند (تخلیه بافر در مرورگر). در پروژه‌ای که ضمیمه شد، تصویر در یک پوشه ذخیره شده و سپس نمایش داده می‌شود. به این صورت تصویر را می‌توان در هر جایی از صفحه بدون ارجاع کاربر به صفحه دوم نمایش داد. البته باید دقت داشت که یوزر `asp.net` باید دسترسی `write` را بر روی این فolder که در اینجا `files` نام دارد، داشته باشد. راه دیگر انجام اینکار استفاده از `http handlers` است. نحوه پیاده سازی این تکنیک را در [این مقاله](#) می‌توانید مشاهده نمائید.

نظرات خوانندگان

نویسنده: hajloo
تاریخ: ۱۳۸۷/۰۹/۱۰ ۱۳:۵۵:۰۰

خیلی استفاده کردم . واقعا مفید بود. اما نم فکر نمی کنم که این سایت فونتی رو روی سیستم کلاینتهای نصب کنه. تنها اتفاقی که می افته اینه که یک متن رو به عنوان ورودی می گیره و بعد (با استفاده از فونتی که روی سرور نصب کرده) یک عکس تولید میکنه .
اما در هر صورت مطلب شما خیلی خوب بود .

نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۰۹/۱۰ ۱۴:۰۸:۰۰

بله. روی کلاینت چیزی نصب نمی کنند. همین بحث بارگذاری پویای فونت را (در سمت سرور) احتمالا پیاده سازی کرده اند که روش آن بحث شد.
یا اگر ساده تر عمل کرده باشند می شود همان روش نصب فونت روی سرور که در این روش بالا شما انعطاف بیشتری خواهید داشت و محدود به فونت های نصب شده نخواهید بود. هر فونتی را می شود بارگذاری کرد. یا حتی تولید captcha با یک سری فونت مخصوص هم به این روش ساده تر خواهد شد (بدون نیاز به هماهنگی با هاست برای نصب فونت).

نویسنده: Alex's Blog
تاریخ: ۱۳۸۷/۰۹/۱۱ ۰۷:۲۰:۰۰

ممnon

نویسنده: Anonymous
تاریخ: ۱۳۸۸/۰۴/۲۰ ۱۱:۵۵:۲۶

با سلام و تشکر از مقاله مفیدتان .
لطفا این پروژه را طوری اصلاح کنید که عکسها runtime ایجاد و نمایش داده شوند و هیچ عکسی موقع تولید ، ذخیره نشود ! بلکه فقط در حین ایجاد ، به کلاینت فرستاده شود .
ممnon .

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۴/۲۰ ۱۳:۰۱:۰۸

سلام،
نکته آنرا اینجا گفته ام:
http://www.dotnettips.info/2009/05/blog-post_27.html

امروز هنگام نمایش تصاویر در یک گرید با مشکل زیر مواجه شده بودم. (گرفتار شده بودم!!)
فرض کنید یک فرم درست کرده‌اید که یک تصویر به همراه توضیحاتی توسط آن ثبت می‌شود. جهت اطمینان خاطر ثبت کننده، تصاویر ثبت شده در یک گرید پائین فرم نمایش داده خواهد شد. نام این تصاویر همواره ثابت است (برای مثال نام تصویر تشکیل شده است از یک شماره پروژه بعلاوه پسوند فایل).
مشکلی در ثبت تصاویر یا توضیحات آنها وجود نداشت. مشکل در این بود که پس از ویرایش رکورد و انتخاب تصویری جدید و ثبت آن، همان تصویر قبلی در مرورگر نمایش داده می‌شد. این مورد را می‌شد با فشردن دکمه‌های [ctrl+f5](#) حل کرد (به روز کردن کش مرورگر)، اما این راه حل اصولی حل این مشکل نیست. (تصورش را بکنید که به کاربر گفته شود پس از هر بار فشردن دکمه ثبت، یکبار هم دکمه‌های [ctrl+f5](#) را فشار دهید!)
[راه حل‌های استاندارد](#) غیرفعال کش در ASP.Net هم هیچکدام افاقه نکردن. ترکیبی از موارد زیر در [page_load](#) صفحه تست شدند:

```
//do not cache...
Response.CacheControl = "no-cache";
Response.AddHeader("Pragma", "no-cache");
Response.Expires = -1;
Response.Cache.SetNoStore();
Response.Cache.SetCacheability(HttpCacheability.NoCache);
```

چون نام تصاویر تغییری نمی‌کرد، برای مثال 1234 همیشه همان 1234 باقی می‌ماند (صرفنظر از محتوای جدید آن)، مرورگر این تصویر غیرپویا را کش می‌کرد و فقط با [ctrl+f5](#) این کش به روز می‌شد. (روش فوق در مورد غیرفعال کردن کش کردن یک صفحه پویای ASP.Net مؤثر است (برای مثال توصیه می‌شود که کش کردن صفحات لایگین را حتماً به این صورت غیرفعال کنید)، اما در مورد اشیاء غیرپویای صفحه مطابق آزمایش من اثری نداشت)

این مشکل به صورت زیر حل شد: (یک ستون [GridView](#) است)

```
<asp:TemplateField HeaderText="لوگو">
<ItemTemplate>
    <asp:Image ID="Image2" runat="server"
        ImageUrl='<%# "pics/" +Eval("filename") + "?uid=" + Guid.NewGuid().ToString("N") %>' />
</ItemTemplate>
<ItemStyle HorizontalAlign="Center" />
</asp:TemplateField>
```

یک عبارت منحصر بفرد به صورت کوئری استرینگ به انتهای نام تصویر اضافه شد. مشکلی هم در نمایش تصویر ایجاد نمی‌کند، مرورگر آن را یک تصویر جدید به حساب آورد و دوباره همان تصویر قبلی موجود در کش را نمایش نخواهد داد. (با IE و فایرفاکس تست شد و اینجا دیگر مهم نیست که وضعیت تنظیمات به روز رسانی کش مرورگرهای تک تک کاربران به چه صورتی است و آیا باید نگران همه‌ی آنها بود یا خیر یا این‌که اصلاً ارزش آنرا دارد که برای یک صفحه کلا کش مرورگر را غیرفعال کرد؟)

زمانیکه [GridView](#) رender شود، تصویر ما به صورت زیر خواهد بود:

```

```

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۰۹/۱۳ ۱۸:۲۶:۰۰

آقای راد هم به نظر قبلًا با این مشکل مواجه شده بودند، البته از نوعی دیگر.

<http://brad.barnamenevis.org/?p=120>

فرض کنید ایمیل اطلاع رسانی برنامه ASP.NET شما قرار است ایمیل زیر را پس از تکمیل یک فرم ارسال کند.

با سلام
احتراماً آقای/خانم [lblName] درخواست چاب [lblDesc] دارای
[lblNumber] صفحه را داده‌اند. لطفاً جهت تائید درخواست ایشان به
برنامه مراجعه بفرمایید.

با تشکر

برای ارسال این قالب که مطابق تصویر هر بار باید سه برچسب آن تغییر کند چه راهی را پیشنهاد می‌دهید؟

راه اول: (راه متدائل)

این فرم را در یک `html editor` درست کرده و جای سه برچسب را خالی می‌گذاریم. سپس `html` مورد نظر را در تابع ارسال ایمیل خود به صورت یک رشته تعریف نموده و جاهای خالی را پر خواهیم کرد. مثلا:

```
string Name = "علی";
string Desc = "منابع مورد نیاز";
int Number = 10;
string content =
"<div dir=\"rtl\" style=\"text-align: right; font-family:Tahoma; font-size:9pt\">" +
"<br />" +
"با سلام<br />" +
"احتراماً آقای/خانم" +
Name +
"درخواست چاپ" +
Desc +
"دارای" +
Number +
"صفحه را داده‌اند. لطفاً جهت تائید درخواست ایشان به برنامه مراجعه بفرمایید" +
"<br />" +
"</div>" +
"با تشکر";
```

ایرادات:

- (الف) امکان مشاهده شکل نهایی تا زمانیکه ایمیل مورد نظر را دریافت نکرده باشیم، وجود ندارد.
- (ب) اعمال تغییرات جدید به این فرمت رشته‌ای مشکل است. همیشه استفاده از ابزارهای بصری برای بهبود کار کمک بزرگی هستند که در این حالت از آن‌ها محروم خواهیم شد.
- (ج) اگر تغییر رسیده جدید، درخواست اضافه کردن لیست پرینت‌های قبلی این شخص بود چه باید کرد؟ آیا جدول مورد نظر را باید به صورت دستی ایجاد و باز هم به صورت یک رشته به این مجموعه اضافه کرد؟ در این حالت از کنترل‌های استانداردی مانند `GridView` و امثال آن محروم خواهیم شد.
- (د) هر بار تغییر، نیاز به `recompile` برنامه دارد.
بنابراین همانطور که مشاهده می‌کنید، نگهداری این روش مشکل است.

راه دوم: استفاده از قالب‌ها

خوشنختانه در ASP.Net امکان رندر کردن کنترل‌ها به صورت یک `string` نیز موجود است. در مثال ما نیاز است تا چندین کنترل در کنار هم قرار گیرند تا شکل نهایی را ایجاد کنند. بنابراین می‌توان تمام آنها را در یک یوزر کنترل قرار داد. سپس باید کل یوزر کنترل را به صورت یک رشته، رندر کرد که در ادامه به آن خواهیم پرداخت.

اگر قالب فوق را بخواهیم در یک یوزر کنترل طراحی کنیم، سورس صفحه `html` یوزر کنترل به صورت زیر خواهد بود (فایل : `(WebUserControl1.ascx)`

```
<%@ Control Language="C#" AutoEventWireup="true" CodeBehind="WebUserControl1.ascx.cs"
Inherits="testWebForms87.WebUserControl1" %>
<div dir="rtl" style="text-align: right; font-family:Tahoma; font-size:9pt">
    سلام با  

    احتراماً آقای خانم  

    <asp:Label ID="lblName" runat="server"></asp:Label>  

    درخواست چاپ  

    <asp:Label ID="lblDesc" runat="server"></asp:Label>  

    دارای  

    <asp:Label ID="lblNumber" runat="server"></asp:Label>  

    <br />.صفحه را داده‌اند. لطفاً جهت تأیید درخواست ایشان به برنامه مراجعه بفرمائید.  

    <br />با تشکر
</div>
```

همچنین می‌توان مقادیر برچسب‌ها را از طریق خواصی که برای یوزر کنترل تعریف خواهیم کرد، در تابع ارسال ایمیل خود مقدار دهی نمائیم. در این حالت در سورس صفحه یوزر کنترل داریم (فایل : `(WebUserControl1.ascx.cs)`)

```
public string Name { get; set; }
public int Number { get; set; }
public string Desc { get; set; }

protected void Page_Load(object sender, EventArgs e)
{
    lblNumber.Text = Number.ToString();
    lblName.Text = Name;
    lblDesc.Text = Desc;
}
```

تا اینجا طراحی اولیه محتوای ایمیل به پایان می‌رسد. دیگر از آن رشته کذا بخوبی نیست و همچنین می‌توان از `designer` ویژوال استودیو برای طراحی بصیری قالب مورد نظر استفاده کرد که این مزیت بزرگی است و در آینده اگر نیاز به تغییر متن ایمیل ارسالی وجود داشت، تنها کافی است فایل `ascx` ما ویرایش شود (بدون نیاز به کامپایل مجدد پروژه). یا در اینجا به سادگی برای مثال می‌توان یک `GridView` را تعریف، طراحی و `bind` کرد.

مرحله بعد، رندر کردن خودکار این یوزر کنترل و سپس تبدیل محتوای حاصل به یک رشته است. برای این منظور از تابع زیر می‌توان کمک گرفت (برای مثال تعریف شده در کلاس `DLXUserControl` :

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Reflection;
using System.Web;
using System.Web.UI;

/// <summary>
/// تبدیل یک یوزر کنترل به معادل اج تی ام ال آن
/// </summary>
/// <param name="path">مسیر یوزر کنترل</param>
/// <param name="properties">هرمراه مقادیر مورد نظر</param>
/// <returns></returns>
/// <exception cref="NotImplementedException"><c>NotImplementedException</c>.</exception>
public static string RenderUserControl(string path,
    List<KeyValuePair<string, object>> properties)
{
```

```

Page pageHolder = new Page();
UserControl viewControl =
    (UserControl)pageHolder.LoadControl(path);
Type viewType = viewControl.GetType();
foreach (var pair in properties)
{
    PropertyInfo property =
        viewType.GetProperty(pair.Key);
    if (property != null)
    {
        property.SetValue(viewControl, pair.Value, null);
    }
    else
    {
        throw new NotImplementedException(string.Format(
            "UserControl: {0} does not have a public {1} property.",
            path, pair.Key));
    }
}
pageHolder.Controls.Add(viewControl);
StringWriter output = new StringWriter();
HttpContext.Current.Server.Execute(pageHolder, output, false);
return output.ToString();
}

```

ماخذ اصلی تابع فوق [این آدرس](#) است.

که البته تابع نهایی آنرا کمی اصلاح کردم تا بتوان لیستی از خواص پابلیک یک یوزر کنترل را به آن پاس کرد و محدود به یک خاصیت نبود.

اکنون استفاده از یوزر کنترلی که تاکنون طراحی کرده‌ایم به سادگی زیر است:

```

List<KeyValuePair<string, object>> lst =
    new List<KeyValuePair<string, object>>
    {
        new KeyValuePair<string, object>("Name", "علی"),
        new KeyValuePair<string, object>("Number", 10),
        new KeyValuePair<string, object>("Desc", "منابع مورد نیاز");
    };
string content = CLoadUC.RenderUserControl("WebUserControl1.ascx", lst);

```

ابتدا لیستی از خواص پابلیک یوزر کنترل تهیه شده و مقدار دهی می‌شوند . سپس مسیر فایل یوزر کنترل به همراه این لیست به تابع رندر کردن نهایی پاس خواهد شد. حاصل، رشته html محتوای ایمیل ما را تشکیل خواهد داد.

نظرات خوانندگان

نویسنده: مهدی پایروند
تاریخ: ۱۳۸۸/۰۵/۱۴ ۱۸:۵۹

توی برنامه های تحت ویندوز چطور؟

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۵/۱۴ ۰۳:۳۸

- یک فایل html درست کنید و جاهای مورد نظر را replace کنید. (قالب درست کنید)
- حالت پیشرفته‌تر استفاده از xs1 transformation در محصولاتش زیاد استفاده می‌کنه.

راههای زیادی برای لاغ کردن خطاهای حاصل در یک برنامه ASP.NET وجود دارند. از روش‌های exception handling معمول تا افزودن یک فایل global.asax به برنامه و دریافت و لاغ کردن خطاهای مدیریت نشده توسط روال رخ داد گردان

بررسی این خطاهای فوق العاده مهم است، حداقل به دو دلیل : (الف) قبل از این که کاربران به شما بگویند برنامه مشکل پیدا کرده، از طریق ایمیل دریافتی مطلع خواهد شد. (فرض کنید علاوه بر ثبت وقایع، آنها را ایمیل هم می‌زنید) این مورد در جهت بالا بردن کیفیت کار تمام شده واقعاً مؤثر است. (ب) رفتارهای مخرب را هم بهتر می‌توانید [تحت نظر](#) داشته باشید.

تمام این موارد مستلزم کد نویسی است. دریافت خطا در روال Application_Error و سپس کد نویسی برای ارسال ایمیل، از ASP.NET 2.0 به بعد این کار را بدون کد نویسی و با استفاده از امکانات [ASP.NET health monitoring](#) نیز می‌توان به سادگی و دقت هرچه تمامتر انجام داد.

کار زیادی را قرار نیست انجام دهید! فایل وب کانفیگ سایت را باز کنید و چند سطر زیر را به آن اضافه کنید (قسمت `: (mailSettings و healthMonitoring`)

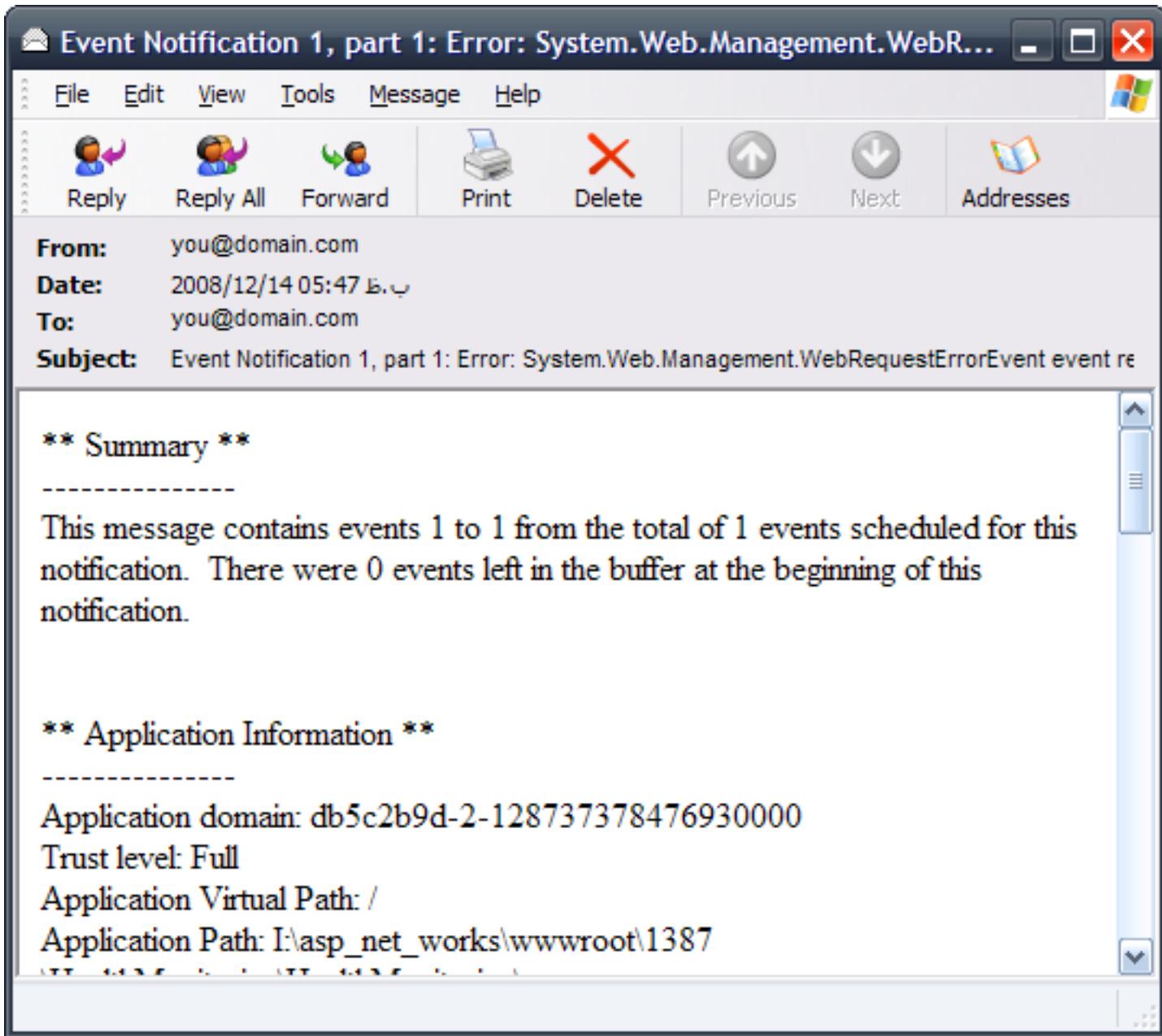
```
<?xml version="1.0"?>
<configuration>
<appSettings/>
<connectionStrings/>
<system.web>
  <compilation debug="true">
  </compilation>
  <authentication mode="Windows"/>

  <healthMonitoring enabled="true">
    <providers>
      <add name="EmailProvider"
           type="System.Web.Management.SimpleMailWebEventProvider"
           from="you@domain.com"
           to="you@domain.com"
           subjectPrefix="Error: "
           buffer="true"
           bufferMode="Notification"/>
    </providers>
    <rules>
      <add
          provider="EmailProvider"
          name="All App Events"
          eventName="All Errors"/>
    </rules>
  </healthMonitoring>
</system.web>

<system.net>
  <mailSettings>
    <smtp deliveryMethod="SpecifiedPickupDirectory">
      <specifiedPickupDirectory pickupDirectoryLocation="C:\emails"/>
    </smtp>
  </mailSettings>
</system.net>

</configuration>
```

در این مثال قسمت mailSettings طوری تنظیم شده که ایمیل ارسالی در مسیر C:\emails جهت مرور نحوه عملکرد این سیستم، ذخیره شود.

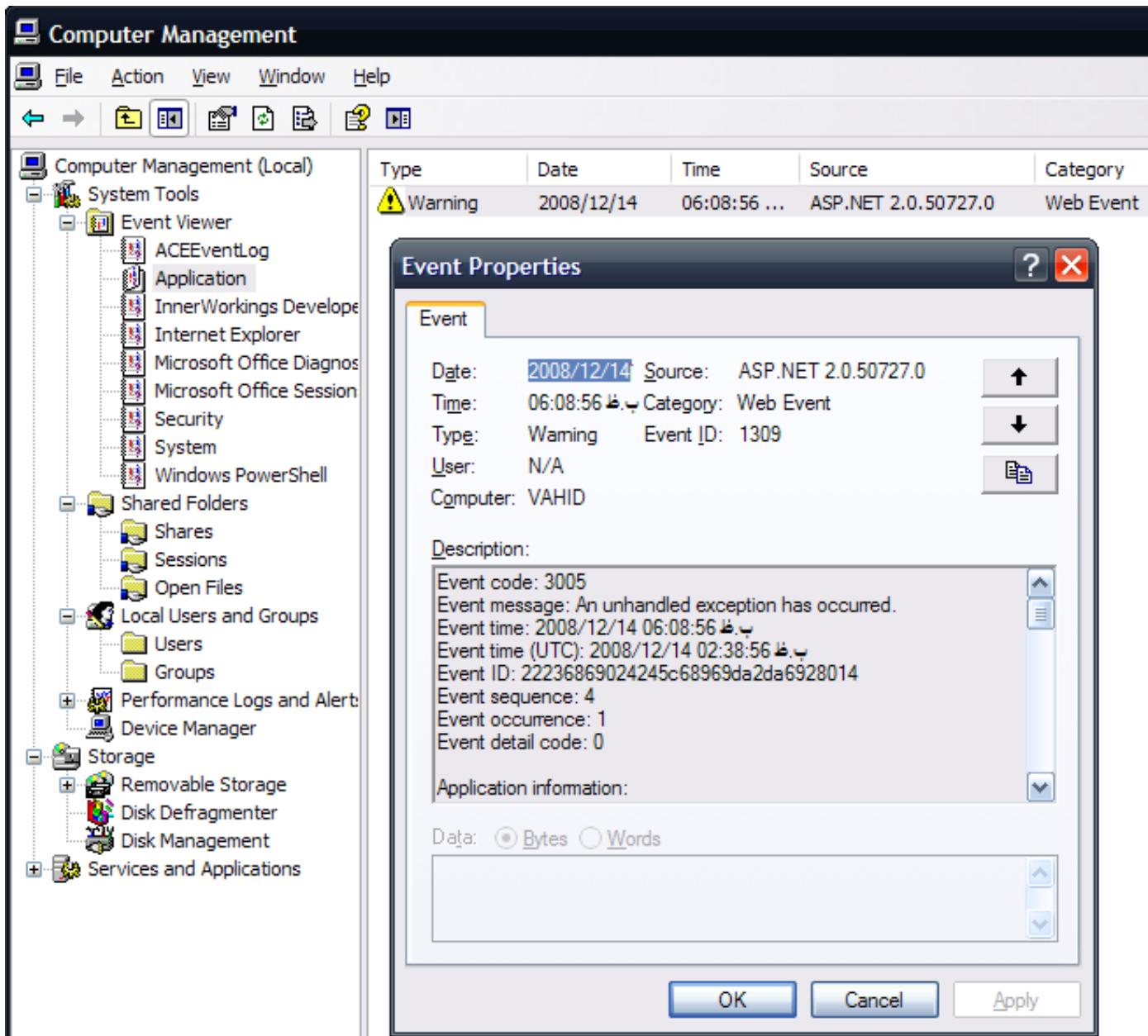


در حالت اجرا بر روی یک سرور، [این قسمت را](#) می‌توان به صورت زیر تنظیم نمود و آدرس smtp server را توسط آن مشخص کرد تا به صورت خودکار مورد استفاده قرار گیرد:

```
<mailSettings>
  <smtp from="you@domain.com">
    <network host="smtp.domain.com" />
  </smtp>
</mailSettings>
```

سیستم ارسال ایمیل آن نیز هوشمند است و میل باکس شما را به یکباره پر نخواهد کرد. اگر در طی یک دقیقه (مقدار پیش فرض) خطاهای حاصل شده مانند هم باشند، تنها یک ایمیل حاوی تمامی آنها را ارسال می‌کند.

شایان ذکر است از ASP.Net 2.0 به بعد امکان ثبت وقایع در event log ویندوز محدود شده است و اگر نیاز به انجام این کار باشد باید دسترسی بیشتری را به یوزر asp.net اعطای کرد. اما با استفاده از روش فوق، جزئیات خطای حاصل به صورت خودکار به event log ویندوز نیز اضافه می‌شود.



اگر علاقمند باشید که خطاهای حاصل را در یک دیتابیس نیز لاغ کنید، به [این مقاله](#) می‌توان رجوع کرد.

نظرات خوانندگان

نویسنده: نیما

تاریخ: ۱۳۸۷/۰۹/۲۴

۲۳:۵۱:۰۰

سلام استاد نصیری
نیما هستم همون مزاحم همیشگی. مقاله جالب بود. من هم زمانی راجع به این موضوع تحقیق کردم. این روشی که فرمودین یه مشکلی داره و اون هم اینه که اگر میل سرور ما رو مجبور به استفاده از SSL کنه این روش جواب نمیده و باید یه پروایدر برای ارسال میل نوشت.

من نتیجه تحقیقاتم رو اینجا گذاشتم.
<http://www.p2p.dotnetsource.com/Default.aspx?g=posts&t=7080>
افتخار میدین اگر نگاهی بندازین بهش و نظرتون رو بگین. موفق باشید

نویسنده: وحید نصیری

تاریخ: ۱۳۸۷/۰۹/۲۵

۰۱:۱۹:۰۰

سلام
ممnon. خیلی عالی و جامع بود.

درسته، چیزی شبیه به این مورد برای

<http://forums.asp.net/t/1124257.aspx>

نویسنده: بهروز

تاریخ: ۱۳۸۷/۰۹/۲۵

۱۲:۲۲:۰۰

تا اودم همون چیزی رو که نیما گفته بگم، دیدم که نیما گفته!
ضمن اینکه برخی هاست ها ارسال ایمیل با کلاس های دات نت رو فراهم نمی کنن و در نتیجه باید یک پرووایدر سفارشی ساخت.

یکی از روش‌های معمولی که برای بدست آوردن اندازه ViewState در ASP.Net بکار می‌رود، نمایش صفحه، مراجعه به سورس آن و سپس ذخیره مقدار فیلد مخفی ViewState به صورت یک فایل متند مراجعه به اندازه آن فایل است!

راه بهتری هم وجود دارد که به شرح زیر است:

زمانیکه قرار است ViewState به صفحه اضافه شود، روال رخداد گردن SavePageStateToPersistenceMedium اجرا خواهد شد.
نحوه تحریف آن برای بدست آوردن اندازه ViewState به صورت زیر است (چند سطر زیر را به وب فرم خود اضافه کنید):

```
using System;
using System.Diagnostics;
using System.IO;
using System.Web.UI;

protected override void SavePageStateToPersistenceMedium(object state)
{
    if (Debugger.IsAttached)
    {
        using (MemoryStream stream = new MemoryStream())
        {
            LosFormatter formatter = new LosFormatter();
            formatter.Serialize(stream, state);
            int size = stream.Capacity / 1024;
            if (size > 32)
            {
                throw new Exception("لطفا به وضعیت ویوو استیت این صفحه رسیدگی نمائید");
            }
        }
        base.SavePageStateToPersistenceMedium(state);
    }
}
```

در اینجا اگر در حال دیباج برنامه باشید و همچنین حجم ViewState از 32 کیلوبایت بیشتر شد، استثنایی جهت گوشتزد گردن این مطلب به شما، تولید می‌شود.

اگر علاقمند بودید که این روال را به تمامی صفحات اضافه کنید می‌شود یک کلاس جدید ایجاد کرد و این کلاس را از کلاس استاندارد Page به ارث برد (برای نمونه public class MyPage : Page). سپس همان چند سطر فوق را به آن اضافه نمایید.
اگر تغییری که در کدها باید صورت گیرد به صورت زیر است، برای مثال تبدیل:

```
public partial class WebForm4 : System.Web.UI.Page
```

```
public partial class WebForm4 : MyPage
```

: به

نظرات خوانندگان

نوبتند: مهرداد
تاریخ: ۱۳۸۷/۱۰/۱۱ ۲۳:۱۱:۰۰

میشه یک توضیح درباره رویداد SavePageStateToPersistenceMedium اینجا بنویسید .

نوبتند: وحید نصیری
تاریخ: ۱۳۸۷/۱۰/۱۱ ۲۳:۱۹:۰۰

هدف اصلی از معرفی این روال رویدادگردن این است که شما بتوانید یک ViewState manager سفارشی برای خودتون تهیه کنید.
یعنی چی؟

در روال SavePageStateToPersistenceMedium کل ViewState ای که قرار است به صفحه به صورت یک فیلد مخفی اضافه شود در اختیار شما خواهد بود. برای مثال آنرا در دیتابیس ذخیره کنید. دیگر هیچ مقداری به فیلد مخفی ViewState اضافه نمی شود.
حالا در روال رویدادگردن LoadPageStateFromPersistenceMedium این ViewState ذخیره شده در دیتابیس را بازیابی کنید و مقدار آنرا بازگشت دهید.

به این صورت صفحه سبک می شود. هم در سمت سرور قابل ذخیره شدن خواهد بود.
برای مثال مقاله زیر از سشن برای حذف ViewState استفاده کرده :

<http://www.dotnetjunkies.com/WebLog/whoiskb/archive/2005/06/23/128346.aspx>

نوبتند: Anonymous
تاریخ: ۱۳۸۷/۱۰/۱۲ ۰۰:۲۹:۰۰

اینو می دونیم که هنگام پستیک یک صفحه باید ویو استیت به سرور آپلود شود که این مستلزم زمان و هزینه است.
آیا امکان فشرده سازی استیت، برای کاهش حجم ارسالی آن وجود دارد؟

نوبتند: وحید نصیری
تاریخ: ۱۳۸۷/۱۰/۱۲ ۰۱:۱۶:۰۰

یک نمونه فشرده سازی ViewState سازگار با ASP.Net Ajax <http://blog.sb2.fr/post/2008/11/18/Ajax-Enabled-ViewState-Compression-for-ASPNET.aspx>

نوبتند: مهدی
تاریخ: ۱۳۸۷/۱۰/۱۴ ۰۲:۳۵:۰۰

مرسى از جوابتون البته مقاله french بود که زحمت ترجمه رو دادم به سرویس translate گوگل.
راستی آقای نصیری به نظرم جای Ajax تو برجسب ها خالیه. فکر کنم بقیه دوستان هم خوشحال می شن اگه راجب ajax از شما بشونند!
مگه نه بچه ها؟

گاهی از اوقات قبل از درگیر شدن با کاربران (!)، بهتر است حد اکثر اندازه مجاز فایل قابل ارسال به سرور را به آن‌ها نمایش داد. در غیراینصورت باید پاسخگوی این باشد که چرا فایل 100 مگابایتی که من ارسال کردم، ذخیره نشده و برنامه کار نمی‌کنه! خطای دریافتی این خواهد بود: `Maximum request length exceeded` در ASP.NET اگر هیچ تنظیم خاصی صورت نگرفته باشد، حد اکثر اندازه فایل قابل ارسال به سرور، [4 مگابایت](#) است. این مورد را در `web.config` و یا در `machine.config` می‌توان [تغییر داد](#). برای مثال، جهت بالا بردن اندازه فایل قابل ارسال به سرور در وب کانفیگ برنامه به 39 مگابایت، می‌توان سطر زیر را به قسمت `system.web` اضافه کرد.

```
<httpRuntime executionTimeout="1200" maxRequestLength="39936" />
```

البته در این حالت بهتر است `executionTimeout` را نیز تنظیم نمود (بر اساس ثانیه) تا یک فایل حجیم را بتوانند آپلود کنند و در این حین مشکل `timeout` رخ ندهد (در اینجا به 20 دقیقه تنظیم شده است).

اما یک نکته را هم باید درنظر داشت. اگر هاست مورد استفاده شما فایل `machine.config` را [قفل کرده باشد](#) (که از لحاظ امنیتی توصیه می‌شود)، سطر فوق در `web.config` هیچ تاثیری نخواهد داشت.

به همین منظور کلاس زیر را تهیه کرده‌ام که تمامی این موارد را لحاظ می‌کند.
ابتدا مقدار پیش فرض 4 مگابایت درنظر گرفته خواهد شد.
سپس سعی می‌شود که مقدار مجاز `MaxRequestLength` از فایل `machine.config` خوانده شود. همچنین وضعیت قفل بودن آن نیز دریافت می‌شود.
اگر این قسمت قابل خواندن بود و همچنین قفل نشده بود، مقدار تنظیم شده `maxRequestLength` در وب کانفیگ، دریافت و استفاده خواهد شد.
و در آخر، اندازه دریافتی، که بر اساس KB است به شکلی قابل خواندن بازگشت داده می‌شود.

```
using System;
using System.Configuration;
using System.Web.Configuration;

/// <summary>
/// جهت نمایش اندازه مجاز فایل قابل ارسال به سرور
/// </summary>
public class CMaxLimit
{
  /// <summary>
  /// اندازه مجاز فایل قابل ارسال به سرور
  /// </summary>
  /// <returns></returns>
  public static string MaxFileSizeLimit()
  {
    // مقدار پیش فرض//
    int resultKB = 4096;

    //machine.config
    Configuration mConfig =
      WebConfigurationManager.OpenMachineConfiguration();
    bool mConfigIsLocked = false;
    HttpRuntimeSection section =
      mConfig.GetSection("system.web/httpRuntime") as HttpRuntimeSection;
    if (section != null)
    {
```

```
resultKB = section.MaxRequestLength;
mConfigIsLocked = section.ElementInformation.IsLocked;
}

//web.config
if (!mConfigIsLocked)
{
    HttpRuntimeSection httpRuntimeSection =
        WebConfigurationManager.GetSection("system.web/httpRuntime") as HttpRuntimeSection;
    if (httpRuntimeSection != null)
    {
        resultKB = httpRuntimeSection.MaxRequestLength;
    }
}

return
    SizeToString(resultKB * 1024);
}

/// <summary>
/// نمایش اندازه یک فایل به صورتی قابل درک
/// </summary>
/// <param name="len">اندازه فایل</param>
/// <returns></returns>
public static string SizeToString(long len)
{
    int order = 0;
    string[] sizes = new[] { "B", "KB", "MB", "GB" };
    while (len >= 1024 && order + 1 < sizes.Length)
    {
        order++;
        len = len / 1024;
    }
    return String.Format("{0:0.##} {1}", len, sizes[order]);
}
```

نظرات خوانندگان

نویسنده: نیما

تاریخ: ۱۳۸۷/۱۰/۲۳ ۱۴:۰۱:۰۰

سلام استاد نصیری

در این موردی که فرمودین اگر config machine قفل باشه اون خط تاثیر گذار نخواهد بود آیا وقتی که قبلش به تگ clear استفاده کنیم هم باز تاثیر گذار نیست؟ چون تگ clear تنظیمات والد رو بی تاثیر میکنه.
منون از مطلب مفیدتون موفق باشید

نویسنده: وحید نصیری

تاریخ: ۱۳۸۷/۱۰/۲۳ ۱۵:۲۰:۰۰

سلام،

بله، بی اثر خواهد کرد اگر allowOverride که برای قفل کردن ذکر شد ، به false تنظیم نشده باشد.
<http://msdn.microsoft.com/en-us/library/ms178693.aspx>
و همانطور که در راهنمای فوق صریحا ذکر شده، هر تلاشی برای تغییر یک section قفل شده، منجر به بروز یک خطا خواهد شد.

کلا قفل کردن ماشین کانفیگ برای هاستها طراحی شده تا از یک سری موارد بتوانند جلوگیری کنند. مثلا شما نتوانید با impersonation ، بجای یک یوزر ویندوز NT لوکال عمل خاصی را روی هاست اجرا کنید.
<http://www.devx.com/vb2themax/Tip/18801>

نویسنده: مسعود

تاریخ: ۱۳۸۷/۱۰/۲۶ ۲۰:۴۵:۰۰

جالب بود استاد.

فقط یه چیزی به نظرت با این کد میشه کل بخش آپلود رو کنترل کرد.

به نظرم مقطعی کار کنه.

نویسنده: وحید نصیری

تاریخ: ۱۳۸۷/۱۰/۲۶ ۲۱:۱۷:۰۰

سلام،

نمی دونم دقیقا منظور شما چیست؟
این کد فقط عبارات درج شده در فایل های کانفیگ رو استخراج و نمایش میده. کار خاص دیگری نمیکنه.

شاید بعضی از سایت‌ها را دیده باشید که در حین ثبت نام، پس از وارد کردن یک نام کاربری و سپس مشغول شدن به پر کردن فیلد کلمه‌ی عبور، در قسمت نام کاربری شروع به جستجو در مورد آزاد بودن نام کاربری درخواستی می‌کنند یا نمونه‌ای دیگر، فرم پرداخت الکترونیکی بانک سامان. پس از اینکه شماره قبض را وارد کردید، بلاfaciale بدون ریفرش صفحه به شما پیغام می‌دهد که این شماره معتر است یا خیر. امروز قصد داریم این قابلیت را با استفاده از کتابخانه‌ی Ajax مجموعه jQuery در ASP.Net پیاده سازی کنیم (بدون استفاده از ASP.Net Ajax مایکروسافت).

ابتدا سورس کامل را ملاحظه نمائید:

```

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="AjaxTest.aspx.cs"
Inherits="testWebForms87.AjaxTest" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>jQuery Ajax Text</title>

<script src="jquery.min.js" type="text/javascript"></script>
<script type="text/javascript">
    $(document).ready(function() {
        $("#<%= TextBox1.ClientID %>").blur(function(event) {
            $.ajax({
                type: "POST",
                url: "AjaxTest.aspx/IsUserAvailable",
                data: {"username": '' + $('#<%= TextBox1.ClientID %>').val() + ''},
                contentType: "application/json; charset=utf-8",
                dataType: "json",
                success: function(msg) {
                    $('#valid').html("<img src='ajaxImages/waiting.gif' alt='> لطفا کمی تامل کنید");
                    var delay = function() {
                        AjaxSucceeded(msg);
                    };
                }
            });
            setTimeout(delay, 2000); //remove this
        }, error: AjaxFailed
    });
});
function AjaxSucceeded(result) {
    if (result.d == true)
        $('#msg').html("<img src='ajaxImages/available.gif' alt='نام کاربری درخواستی موجود'> است");
    else
        $('#msg').html("<img src='ajaxImages/taken.gif' alt='متاسفانه نام کاربری مورد نظر پیشتر دریافت شده است'>");
}
function AjaxFailed(result) {
    alert(result.status + ', ' + result.statusText);
}
</script>

</head>
<body>
<form id="form1" runat="server">
<div>
    user name:
    <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    <span id="msg"></span>
    <br />
    pass:
    <asp:TextBox ID="TextBox2" TextMode="Password" runat="server"></asp:TextBox>
</div>
<!-- preload -->
<div style="display: none">
    
    
</div>

```

```

</div>
</form>
</body>
</html>
```

```
using System;
using System.Web.Services;

namespace testWebForms87
{
    public partial class AjaxTest : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        [WebMethod]
        public static bool IsUserAvailable(string username)
        {
            این مورد را با خواندن اطلاعات از دیتابیس می‌شود تعویض کرد // return username != "test";
        }
    }
}
```

توضیحات:

همانطور که ملاحظه می‌کنید صفحه‌ی ASP ما بسیار ساده است و از دو تکست باکس استاندارد تشکیل می‌شود، به همراه تصاویر مربوط به Ajax که یک سری تصاویر ساده چرخان معروف منتظر بمانید، یافت شد یا موجود نیست می‌باشد. این تصاویر در یک div مخفی (display: none) در صفحه قرار گرفته‌اند و در هنگام بارگذاری صفحه، این‌ها نیز بارگذاری شده و حاضر و آماده خواهند بود. بنابراین هنگام استفاده از آن‌ها، کاربر تا خیری را مشاهده نخواهد کرد. همچنین یک span با id msg را هم پس از تکست باکس اضافه کرده‌ایم تا تصاویر مربوط به رخدادهای Ajax را با استفاده از توابعی‌های jQuery به آن اضافه کنیم.

اسکریپت Ajax ما با دراختیار گرفتن روال رخداد گردان blur شیء textBox1 شروع می‌شود. همانطور که در مقالات پیشین سایت نیز ذکر شد، روش صحیح دریافت ID یک کنترل ASP.NET در کدهای سمت کلاینت جاوا اسکریپتی، بر اساس خاصیت ClientID آن است که در اولین سطر کدهای ما مشخص است (زیرا در ASP.NET نام و ID یک کنترل در هنگام رندر شدن به همراه ID کنترل‌های دربرگیرنده آن نیز خواهد بود، بنابراین بهتر است این مورد را داینامیک کرد).

کار بررسی موجود بودن نام کاربری (یا مثلاً یک شماره قبض و امثال آن) توسط `IsUserAvailable` ای به نام `WebMethod` در `code` صفحه انجام می‌شود که پیاده سازی آن را ملاحظه می‌کنید. بدیهی است در این مثال ساده، تنها نام کاربری از پیش رزرو شده، کلمه‌ی `test` است و در یک کد واقعی این مورد با مقایسه‌ی نام کاربری با اطلاعات موجود در دیتابیس باید صورت گیرد (و حملات تزریق اس کیوال را هم فراموش نکنید. برای رهایی از آن‌ها " حتماً" باید از پارامترهای ADO.NET استفاده کرد و گرنه کد شما مستعد به این نوع حملات خواهد بود).

سؤال: چرا از `web method` استفاده شد و همچنین چرا این متدهای `static` است؟
زمانیکه یک متدهای کلیدی `static` مشخص می‌شود حالت state less پیدا می‌کند یعنی مستقل از وله‌ی کلاس عمل می‌کند. در این حالت نیازی به ارسال `ViewState` نبوده (بنابراین در کوئری مورد نظر ما بسیار بهینه و سبک عمل می‌کنند) و همچنین نیازی به ایجاد یک وله‌ای از کلاس صفحه‌ی ما نیز نخواهد بود. برای توضیحات بیشتر به این مقاله مراجعه نمائید. به صورت خلاصه، دلیل اصلی، کارآیی بالا و بهینه بودن این روش در این مساله ویژه است و در ASP.NET Ajax مایکروسافت به صورت گسترده‌ای در پشت صحنه مورد استفاده قرار می‌گیرد)

استفاده از ویژگی `WebMethod` عملکرد صفحه‌ی ما را شبیه به یک وب سرویس خواهد کرد و امکان دسترسی به آن در متدهای استاندارد POST به صورت ارسال دیتا به آدرس `WebService.asmx/WebMethodName` خواهد بود. یک مثال ساده و عملی

بررسی تابع Ajax بکار رفته:

این تابع هنگام فراخوانی رخداد blur تکستباکس ما (مطابق کد فوق) فراخوانی می‌شود. ساختار ساده‌ای دارد که به شرح زیر است:

نحوه‌ی ارسال داده را به متدهای سرویس ما مشخص می‌کند.

ارسال "AjaxTest.aspx/IsUserAvailable" اطلاعات پست شده، به صفحه‌ی AjaxTest.aspx و وب متدهای IsUserAvailable خواهد شد

ما ارسال "IsUserAvailable" اطلاعات پست شده، به آرگومان username که به آرگومان آن به صورت داینامیک گرفته تا مشکل زا نشود، همان مقدار تایپ شده در TextBox1 دریافت ID آن به اشیاء مورد نظر خواهد شد. برای

مثال اگر خروجی یک وب سرویس در حالت XML به صورت زیر باشد:

```
, "contentType: "application/json; charset=utf-8
"dataType: "json", در این دو سطر از نوع داده‌ی JSON استفاده شده است که فرمت سیار سبک و بهینه‌ای برای تبادل اطلاعات در وب به شمار می‌آید و توسط کتابخانه‌های جاوا اسکریپتی به سادگی پردازش شده و تبدیل به اشیاء مورد نظر خواهد شد. برای
```

```
<xx yy="nn"></xx>
```

معادل JSON آن به شرح زیر است:

```
{ "xx": { "yy": "nn" } }
```

پس از موفقیت آمیز بودن عملیات Ajax در jQuery فراخوانی می‌شود

و اگر در این بین خطای رخداده باشد، قسمت error فراخوانی می‌شود.

در این مثال برای نمایش بهتر عملیات، یک وقفه‌ی 2 ثانیه‌ای توسط setTimeout ایجاد شده و بدیهی است در یک مثال واقعی باید آنرا حذف نمود.

نکته: با استفاده از افزونه‌ی فایرباگ فایرفاکس، می‌توان جزئیات این عملیات را بهتر مشاهده نمود:

user name: 

pass:

Inspect Clear Profile

Console ▾ HTML CSS Script DOM Net Cookies Reference Pixel Perfect YSlow

POST http://localhost:9884/AjaxTest.aspx/IsUserAvailable 200 OK 42ms

Headers Post Response JSON

```
{"username": "test1"}
```

user name: 

pass:

Inspect Clear Profile

Console ▾ HTML CSS Script DOM Net Cookies Reference Pixel Perfect YSlow

POST http://localhost:9884/AjaxTest.aspx/IsUserAvailable 200 OK 42ms

Headers Post Response JSON

```
{"d":true}
```

نظرات خوانندگان

نویسنده: مسعود

تاریخ: ۱۳۸۷/۱۱/۲۹ ۲۲:۳۸:۰۰

سلام استاد.

من برام یه سوال پیش اومند ،

من چند وقته ذهنم درگیر این شده که آیا واقعا Static متدهای vb.net رو داره یا نه!

من سر این با خیلی ها بحث کردم.

بعضی ها میگن Modual جایگزینش شده ، بعضی ها متدهای Shared رو جایگزین می دونن.

نظر شما چیه؟

راستی شما روی jQuery Ajax چ کار میکنید؟

من خیلی دوست دارم کار کردن باهاش رو.

دارم یه فریم ورک کوچولو برای Jquery Ajax برای دات نت می سازم. البته پر از ارور :دی

نویسنده: وحید نصیری

تاریخ: ۱۳۸۷/۱۱/۳۰ ۰۱:۰۴:۰۰

سلام

بله. برنامه reflector معروف که مد نظر هست. اسambilی برنامه فوق را با آن باز کردم و زبان انتخابی هم VB . حاصل به صورت زیر نمایش داده شد:

Public Shared Function IsUserAvailable

یک سری کار سورس باز دیدم در CodePlex که تلاش هایی برای ساده تر کردنش کردند. بد نیست برای تکمیل کار یک سری هم به آن ها هم بزنید. برای ایده گرفتن خوب است.

نویسنده: میلاد

تاریخ: ۱۳۸۸/۰۵/۰۱ ۱۰:۲۱:۱۳

سلام

اگه بخوایم دو پارامتر ارسال کنیم چی ؟ !!

نویسنده: وحید نصیری

تاریخ: ۱۳۸۸/۰۵/۰۱ ۱۰:۳۰:۴۶

محدو دیتی ندارد. وب متدهای تا دو پارامتر مورد نظر را قبول کند. همچنین قسمت data در اسکریپت را که با فرمت json باید ارسال شود را نیز اصلاح کنید. مثلا

```
'{var json = '{"param1":"val1", "param2":"val2", "param3":number3'}
```

نویسنده: mat21

تاریخ: ۱۳۸۸/۱۰/۰۹ ۱۱:۵۹:۵۷

سلام و تشکر

من مثال فوق را استفاده کردم اما همیشه قسمت error به اجرا در میاد و 200 error status را بر می گرداند اما زمانی که json "datatype": "json" را لازم کرد حذف می کنم قسمت success اجرا می شد اما نتیجه درست نیست.
من از framework 2.0 استفاده می کنم، ممنون می شم راهنمایم کنید.

نویسنده: وحید نصیری
تاریخ: ۱۴:۲۰:۴۴ ۱۳۸۸/۱۰/۰۹

سلام

در مورد دات نت 2 به آخرین کامنت بنده در این مقاله مراجعه بفرمایید (انتهای صفحه):
http://www.dotnettips.info/2009/07/blog-post_06.html

نویسنده: maryam
تاریخ: ۱۲:۳۲:۵۳ ۱۳۸۹/۰۷/۲۰

من این برنامه را تست کردم همیشه قسمت error را اجرا می کند و این پیغام می دهد
Internal Server Error 500

نویسنده: وحید نصیری
تاریخ: ۱۳:۱۷:۳۸ ۱۳۸۹/۰۷/۲۰

کمکی نمی تونم بهتون بکنم و من عموما پیغام هایی مانند پیغام شما را حذف می کنم به این دلایل:
- نه کدی ارائه شده (کل Solution شما نیاز است برای بررسی)
- نه توضیحات کافی که چه مراحلی را طی کردید
- نه توضیحاتی که از چه ابزاری و چه نگارشی از VS.NET دارید استفاده می کنید.

و توقع دارید که من از راه دور کد شما را در VS.NET آیی که معلوم نیست چه نگارشی است دیباگ کنم، پس از عیب یابی مشخص کنم مشکل چی بوده ...

اینها رو هم نوشتیم تا کسانی که پیغام هاشون حذف شده بدونند علت این امر چی بوده.

نویسنده: وحید نصیری
تاریخ: ۱۳:۲۰:۴۱ ۱۳۸۹/۰۷/۲۰

یک مورد کلی را اضافه کنم برای دیباگ پیغام Internal Server Error 500 که عرض کردم از راه دور نمی شود آن را دیباگ کرد: افزونه فایربراگ مربوط به فایرفاکس را نصب کنید. سپس آن را روی صفحه جاری خودتون فعال کنید. صفحه رو ریفرش کنید و سپس یکبار مراحل رسیدن به خطأ رو طی کنید. در قسمت network مربوط به firebug دقیقا اطلاعات رد و بدل شده و همچنین خطای نهایی که سیستم برگردانده لاغ می شود. این را بررسی کنید تا مشکل را بشود حل کرد.

نویسنده: شاهین کیاست
تاریخ: ۲۱:۴۱:۴۳ ۱۳۸۹/۱۱/۱۱

سلام آقای نصیری.

من عین کد شما را در Page تست کردم و صحیح کار کرد اما در User Control pیام Forbidden را بر می گرداند. آیا علت خاصی دارد؟

نویسنده: وحید نصیری
تاریخ: ۲۳:۲۸:۳۶ ۱۳۸۹/۱۱/۱۱

بله. در یوزر کنترل ها این مورد ساپورت نمی شود. در تمام حالات روش صحیح و قابل نگهداری و قابل استفاده در قسمت های

مختلف برنامه، استفاده از یک WCF Service است (یا یک درجه پایین‌تر یک وب سرویس معمولی asmx با سرعت کمتر از WCF سرویس).

نویسنده: ایمان اسلامی
تاریخ: ۱۴:۲۶ ۱۳۹۱/۰۴/۲۱

با سلام و خسته نباشید خدمت استاد نصیری من برنامه ام رو با code first و الگوی UnitOfWork که شما مطرح کردید نوشتم که واقعاً عالی و لذت بخش هست. مشکلی که هست اینه که اگه بخواه از این امکان jqueryAjax ز در برنامه خودم استفاده کنم می‌بایست userServieice تعريف شده رو در یک متد استاتیک(با توجه به مطلب بالا) استفاده کنم که اینکار انجام نمی‌شه. ممنون می‌شم کمک کنید.

نویسنده: وحید نصیری
تاریخ: ۱۶:۴ ۱۳۹۱/۰۴/۲۱

- در MVC این مشکلات وجود ندارد و سازگاری کاملی با jQuery در آن درنظر گرفته شده. در آنجا لازم نیست چیزی را استاتیک تعريف کنید. همچنین چون صفحات دارای ViewState هم نیستند، به سادگی می‌شود قسمتی از آنرا با jQuery Ajax کلا تعویض کرد بدون اینکه پیغام خراب بودن ViewState را در post back به سرور، دریافت کنید. در آنجا مفهومی به نام PartialView تعريف شده که به صورت خودکار یک View را به شکل رشته برای شما رندر می‌کند و در اختیار jQuery Ajax قرار می‌دهد.
- در اینجا اگر نمی‌توانید از وله سازی خودکار توسط DI Container استفاده کنید، می‌شود آنرا دستی هم انجام داد. اصطلاحاً به آن Service locator گفته می‌شود. نمونه آنرا در استفاده از ObjectFactory.GetInstance در [قسمت ۱۲](#) دیده‌اید (استفاده از الگوی واحد کار و کلاس‌های سرویس تهیه شده در یک برنامه کنسول ویندوزی).

نویسنده: ایمان اسلامی
تاریخ: ۱۶:۳۰ ۱۳۹۱/۰۴/۲۱

من فراموش کردم که اشاره کنم من از asp.net web form استفاده می‌کنم.
راه حل برای asp.net web form چیست؟

نویسنده: وحید نصیری
تاریخ: ۱۶:۳۳ ۱۳۹۱/۰۴/۲۱

همان توضیحات بند دوم نظر قبلی بنده (الگوی service locator) مخصوص به فناوری خاصی نیست و در صورت لزوم در همه جا می‌توان از آن استفاده کرد. یک ضد الگو است. تا حد امکان نباید از آن استفاده کرد؛ اما اگر امکان استفاده صحیح از تزریق وابستگی‌ها نبود، این روش هم کار می‌کنه).

نویسنده: ایمان اسلامی
تاریخ: ۱۶:۵۶ ۱۳۹۱/۰۴/۲۱

مهندس فقط می‌تونم بگم " یه دونه ای !"
واقعاً ممنون

کار کرد. البته در تمامی page ها از base Page ارث بری می‌کنیم. این استفاده از Object Factory با توجه به استفاده از Factory در سازنده کلاس base page ، مشکلی ایجاد نمی‌کند؟

نویسنده: وحید نصیری
تاریخ: ۱۷:۱۱ ۱۳۹۱/۰۴/۲۱

خیر. در پشت صحنه این کتابخانه، Object Factory به صورت خودکار عمل می‌کند. اگر امکان استفاده خودکار از آن نبود، روشی دستی آن (ObjectFactory.GetInstance) همان عملیات را انجام می‌دهد.

نویسنده: مهدی غفاری
تاریخ: ۱۳۹۱/۰۷/۰۸ ۱۵:۴۶

من یک پوشه ایجاد کردم و صفحه AjaxTest.aspx را تو اون قرار دادم بعد برای اینکه کسی نتوانه با وارد کردن آدرس اون به صفحه دسترسی داشته باشه، توی اون پوشه یه web.config ایجاد کردم و تگ زیر رو به اون اضاف کردم

```
<deny users="*"/>
```

چون میخواستم هیچ کس نتوانه اون صفحه رو ببینه. ولی خوب کدهای نوشته شده توسط من باید بتونن به اون صفحه دسترسی داشته باشن (فقط برنامه) لطفا بفرمائید این رو چطور ادیت کنم تا این کار ممکن بشه؟
ممنونم از شما

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۰۸ ۱۶:۳۱

با کلا jQuery یا کلا JavaScript یعنی کدهای سمت کاربر. بنابراین این کدها باید دسترسی رسیدن به مدخل سمت سرور را داشته باشند و اگر عموم از آن منع شده باشند، یک اسکریپت معمولی هم بدیهی است که دسترسی خاصی نخواهد داشت چون سمت سرور اجرا نمی‌شود. فقط نتیجه عملیات آن به سمت سرور Post می‌شود.
یکی از کارهای متداول سمت سرور جهت منع عموم و فقط دسترسی دادن به jQuery مورد زیر است:
« [بررسی امنیتی، حین استفاده از jQuery](#) »

نویسنده: مهدی غفاری
تاریخ: ۱۳۹۱/۰۷/۰۸ ۲۲:۴۶

ممنونم از توجه شما:
آیا نمیشه از هندرها استفاده کرد تو همچین موقعی؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۰۸ ۲۲:۵۲

بله امکان پذیر است. مثل این مثال [اسلایدشو](#) می‌تونید JSON بازگشت دهید.

عنوان: نگارش نهایی ASP.NET MVC ارائه شد
نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۱/۰۱ ۱۶:۰۴:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: ASP.NET

دریافت نگارش نهایی

مشاهده مستندات و راهنمای مربوطه

کتابی از مدیر پروژه مربوطه

سایت رسمی

برای استفاده از [jQuery Ajax](#) یکی از روش‌های ارسال دیتا به برنامه، تبدیل داده‌ها به فرمت JSON می‌باشد. برای داده‌های ساده، تشخیص این فرمت ساده است. مثلاً اگر امضای تابع وب سرویس اجکس ما به صورت زیر باشد:

```
public static bool IsUserAvailable(string username)
```

اطلاعات جی‌سوئی را که قرار است ارسال کنیم، فرمت زیر را باید داشته باشد:

```
{'username':'value'}
```

حال اگر آرگومان‌های ما پیچیده‌تر بودند چطور؟ مثلاً بجای یک رشته ساده، یک لیست جنریک داشتیم، فرمت ورودی را چگونه باید تشخیص داد؟ برای این منظور در دات نت ۳ و نیم، کلاسی جهت انجام اینگونه تبدیلات پیش‌بینی شده است که شرح مختصر آن به صورت زیر است:

ابتدا باید ارجاعی را به اسمبلی `system.web.extensions` به برنامه افزود و سپس جهت سهولت کار می‌توان یک `System.Web.Script.Serialization.JavaScriptSerializer` مهیا در فضای نام `JavaScriptSerializer` `method`:

```
public static string ToJson(this object data)
{
    return new JavaScriptSerializer().Serialize(data);
}
```

اکنون چند مثال زیر را در نظر بگیرید:

```
public static string GetJsonTest0()
{
    var data = "a1";
    return data.ToJson();
}

public static string GetJsonTest1()
{
    var data = new List<string> { "a1", "a2", "a3" };
    return data.ToJson();
}

public static string GetJsonTest2()
{
    var lst =
        new List<KeyValuePair<string, object>>
    {
        new KeyValuePair<string, object>("Name", "علی"),
        new KeyValuePair<string, object>("Number", 10),
        new KeyValuePair<string, object>("Desc", "منابع مورد نیاز");
    };
    return lst.ToJson();
}
```

خروجی‌های آن‌ها به ترتیب به صورت زیر خواهد بود:

```
"a1"
```

```
[ "a1", "a2", "a3" ]  
[ { "Key": "Name", "Value": "علي" }, { "Key": "Number", "Value": 10 }, { "Key": "Desc", "Value": "منابع مورد نیاز" } ]
```

این کلاس همچنین قابلیت Deserialize و تبدیل داده‌هایی به فرمت JSON به اشیاء مورد نظر ما را نیز دارد.

کنترل FileUpload زمانیکه داخل UpdatePanel Ajax قرار گیرد کار نمی‌کند و خاصیت Hasfile آن در این حالت همیشه بر می‌گردداند، زیرا این کنترل برای کار کردن نیاز به post کامل دارد. برای معرفی این post back داخل panel ، باید یک تریگر به نام postbacktrigger را افزود و سپس ControlID آنرا مساوی کنترلی قرار داد که جهت ارسال فایل به سرور مشخص کردۀایم.
یک مثال:

```
<asp:UpdatePanel runat="server" id="UpdatePanel1">
    <ContentTemplate>
        <asp:FileUpload runat="server" id="FileUpload1" />
        <asp:Button runat="server" id="btnSubmit" Text="Upload" />
    </ContentTemplate>
    <Triggers>
        <asp:postbacktrigger ControlID="btnSubmit" />
    </Triggers>
</asp:UpdatePanel>
```

این تریگر عملیات post back را سبب خواهد شد و اکنون عملیات ارسال فایل به سرور صحیح کار خواهد کرد.

نظرات خوانندگان

نویسنده: مهرداد
تاریخ: ۱۳۸۸/۰۲/۰۳ ۲۲:۲۷:۰۰

مثل همیشه بسیار عالی ...
ممnon آقای نصیری .

با تشکر ، مهرداد قاسمی

نویسنده: Anonymous
تاریخ: ۱۳۸۸/۰۲/۰۷ ۱۹:۰۱:۰۰

آیا امکان اینکه فایل هم به صورت Ajax آپل.د بشه وجود نداره ؟

مثل کاری که Gmail برای Attachment هاش انجام میده ؟

Silverboy

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۲/۰۷ ۲۰:۰۳:۰۰

باید در نحوه ارسال فایل به سرور دست برد و اون رو سفارشی کرد. مثلا با استفاده از یک generic handler . نحوه پیاده سازی آن را در jQuery می تونید اینجا مشاهده کنید:
<http://www.ajaxprojects.com/ajax/tutorialdetails.php?itemid=512>

توسط ASP.Net ajax هم همین مفهوم به شکل دیگری پیاده سازی می شود:
<http://msmvps.com/blogs/luisabreu/archive/2006/12/14/uploading-files-without-a-full-postback.aspx>

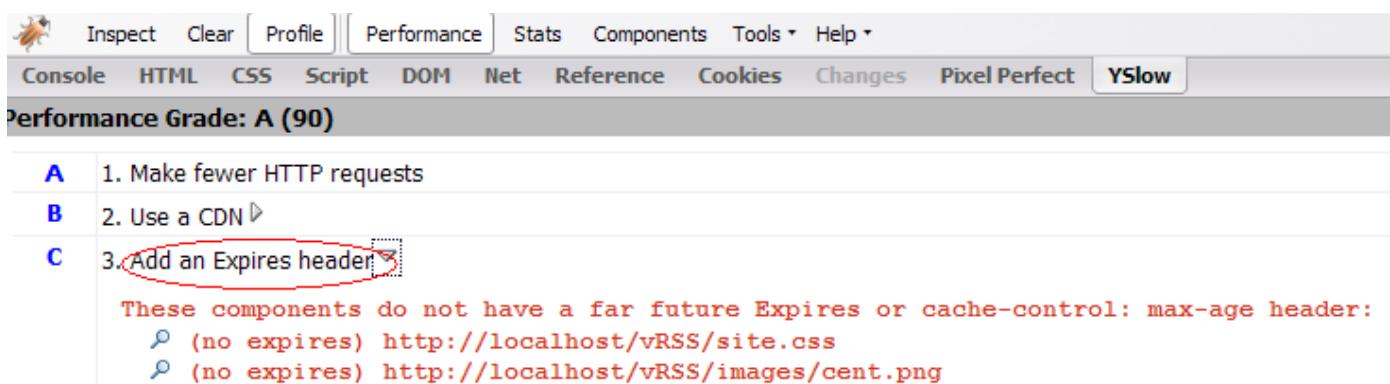
نویسنده: Anonymous
تاریخ: ۱۳۸۸/۰۲/۲۵ ۱۰:۵۴:۰۰

ممnon خیلی مفید بود

نویسنده: Anonymous
تاریخ: ۱۳۸۸/۰۴/۱۴ ۱۰:۴۷:۲۱

ممnon خیلی خوب بود. یه reference ajax معرفی می کنید. مثلا همین پنجره captcha رو چجوری با ajax درست می کنن؟؟؟

در مورد افزونه YSlow افزونه Firebug فایرفاکس پیشتر صحبت شد. این افزونه پس از آنالیز یک سایت، پیشنهاداتی را نیز جهت بهبود سرعت، ارائه می‌دهد.

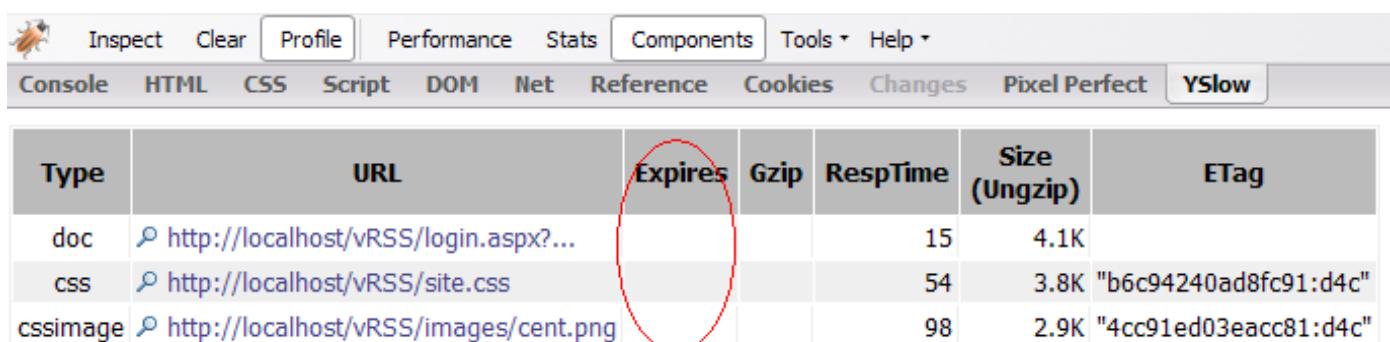


Performance Grade: A (90)

- A 1. Make fewer HTTP requests
- B 2. Use a CDN ▶
- C 3. Add an Expires header X

These components do not have a far future Expires or cache-control: max-age header:

- 🔗 (no expires) <http://localhost/vRSS/site.css>
- 🔗 (no expires) <http://localhost/vRSS/images/cent.png>



Type	URL	Expires	Gzip	RespTime	Size (Ungzip)	ETag
doc	🔗 http://localhost/vRSS/login.aspx?...			15	4.1K	
css	🔗 http://localhost/vRSS/site.css			54	3.8K	"b6c94240ad8fc91:d4c"
cssimage	🔗 http://localhost/vRSS/images/cent.png			98	2.9K	"4cc91ed03eacc81:d4c"

همانطور که در شکل بالا مشخص است، عناصری مانند css و js ، قسمت expires اشاره (تاریخ منقضی شدن کش آنها در سمت کلاینت) خالی است و پیشنهاد داده که به هر کدام از این عناصر، هدر مخصوص مشخص سازی مدت زمان کش شدن در سمت کلاینت اضافه شود.

در مورد کش کردن اطلاعات صفحات پویا به اندازه‌ی کافی امکانات در اختیار برنامه نویس قرار می‌دهد اما در مورد اضافه کردن این هدر جهت یک فایل css غیر پویا شاید نتوان مطلب خاصی را یافت.

در IIS 7 امکانات ویژه‌ای برای این منظور در نظر گرفته شده که نحوه استفاده از آن در ASP.Net به صورت زیر است:
فایل وب کانفیگ سایت را باز کرده و به قسمت system.webServer چند سطر زیر را اضافه کنید:

```
<staticContent>
<clientCache httpExpires="Sun, 29 Mar 2020 00:00:00 GMT" cacheControlMode="UseExpires" />
</staticContent>
```

این مورد فقط مختص به IIS7 است و بر روی نگارش‌های پایین‌تر کار نمی‌کند.
با این کار، تاریخ منقضی شدن هر آنچه که توسط موتور ASP.net سرو نمی‌شود به سال 2020 تنظیم خواهد شد. (کلیه محتواهی غیرپویای سایت، اعم از تصاویر، فایلهای js، css و غیره)
پس از این تنظیم مجدد YSlow را اجرا کرده و Performance Grade ای را که نمایش می‌دهد بررسی نمائید.

بدیهی است اگر یکی از فایل‌های css یا js شما تغییر کند، کلاینت، اطلاعات جدیدی را تا سال 2020 دریافت نمی‌کند. برای حل این مشکل یک کوئری استرینگ ساده به انتهای لینک مربوط به css یا js خود اضافه کنید تا URL جدید با URL قبلی آن یکسان نباشد (این کوئری استرینگ تاثیری روی محتواهای ایستای ما ندارد). به این صورت این آدرس جدید، مجدد دریافت شده و تا سال 2020 کش خواهد شد.

ذکره:

اعمال تنظیم فوق، در IIS7 ویندوز سرور 2008 مجاز است؛ اما در IIS7 ویندوز ویستا قفل شده است و قابل override نیست.
برای تغییر آن، فایل زیر را پیدا کنید:

```
open %systemroot%\System32\inetsrv\config\applicationHost.config
```

و در آن سطر

```
<section name="staticContent" overrideModeDefault="Deny" />
```

را به صورت زیر تغییر دهید تا مجاز به اعمال تغییرات شوید:

```
<section name="staticContent" overrideModeDefault="Allow" />
```

در قسمت بعد در مورد نگارش‌های پایین‌تر IIS توضیح داده خواهد شد.

ما آخذ:

[YSlow: Add expires header to images in IIS 7](#)

[?IIS7: How to set cache control for static content](#)

قسمت قبل

به IIS7 اختصاص داشت که شاید برای خیلی‌ها کاربرد نداشته باشد خصوصاً اینکه برنامه نویس‌ها ترجیح می‌دهند به روش‌هایی روی بیاورند که کمتر نیاز به دخالت مدیر سرور داشته باشد؛ یا زمانی‌که سایت شما بر روی یک هاست اینترنتی قرار گرفته است عملایشاید دسترسی خاصی به تنظیمات IIS نداشته باشد (مگر اینکه یک هاست اختصاصی را تهیه کنید). برای IIS6 و ماقبل از آن و حتی بعد از آن!، حداقل دو روش برای کش کردن اطلاعات استاتیک وجود دارد:

الف) استفاده از web resources معرفی شده در ASP.Net 2.0 به بعد در مورد نحوه تعریف و بکارگیری web resources می‌توان به مقاله "[تبديل پلاگین‌های jQuery به کنترل‌های ASP.Net](#)" رجوع کرد.

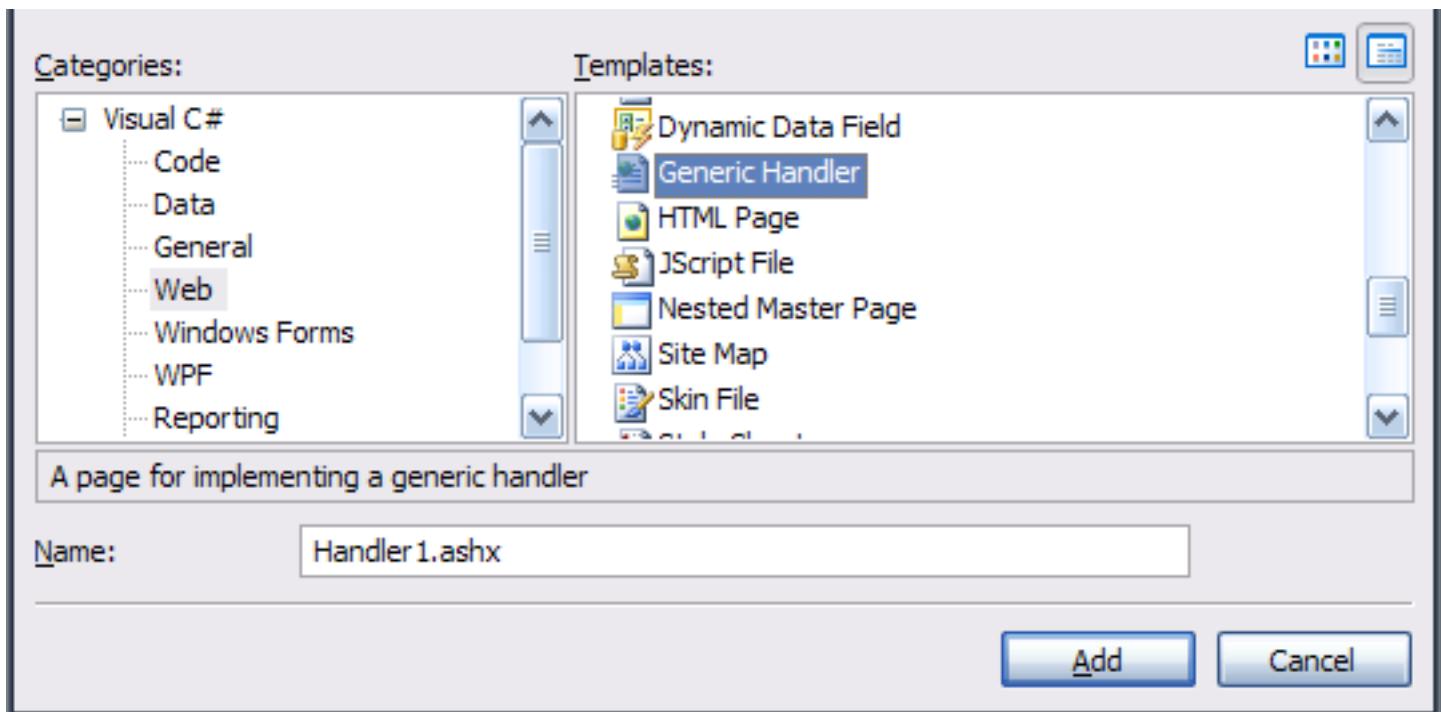


Type	URL	Expires	Gzip	RespTime	Size (Ungzip)	ETag
doc	http://localhost/library/login.aspx			104	13.7K	
js	http://localhost/library/WebResource.axd?...	4/26/2010		18	20.7K	
js	http://localhost/library/WebResource.axd?...	4/26/2010		9	21.5K	
js	http://localhost/library/WebResource.axd?...	4/26/2010		10	3.0K	

همانطور که در شکل فوق نیز ملاحظه می‌کنید، هدر مربوط به مدت زمان منقضی شدن کش سمت کلاینت یک web resource به صورت خودکار به سال 2010 تنظیم شده است و این مقدار خالی نیست.

ب) افزودن این هدر به صورت دستی

برای این منظور باید در نحوه ارائه فایل‌های استاتیک دخالت کنیم و این کار را با استفاده از یک generic handler می‌توان انجام داد.



کد این generic handler می‌تواند به صورت زیر باشد:

```

using System;
using System.IO;
using System.Web;
using System.Web.Services;
using System.Reflection;

namespace test1
{
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    public class cache : IHttpHandler
    {

        private static void cacheIt(TimeSpan duration)
        {
            HttpCachePolicy cache = HttpContext.Current.Response.Cache;

            FieldInfo maxAgeField = cache.GetType().GetField("_maxAge", BindingFlags.Instance | BindingFlags.NonPublic);
            maxAgeField.SetValue(cache, duration);

            cache.SetCacheability(HttpCacheability.Public);
            cache.SetExpires(DateTime.Now.Add(duration));
            cache.SetMaxAge(duration);
            cache.AppendCacheExtension("must-revalidate, proxy-revalidate");
        }

        public void ProcessRequest(HttpContext context)
        {
            string file = context.Request.QueryString["file"];
            if (string.IsNullOrEmpty(file))
            {
                return;
            }

            string contetType = context.Request.QueryString["contetType"];
            if (string.IsNullOrEmpty(contetType))
            {
                return;
            }

            context.Response.Write(File.ReadAllText(context.Server.MapPath(file)));
            //Set the content type
        }
    }
}

```

```

        context.Response.ContentType = contetType;
        // Cache the resource for 30 Days
        cacheIt(TimeSpan.FromDays(30));
    }

    public bool IsReusable
    {
        get
        {
            return false;
        }
    }
}

```

توضیحات:

این generic handler دو کوئری استرینگ را دریافت می‌کند: file چهت دریافت نام فایل و contetType چهت مشخص سازی نوع محتوایی که باید سرو شود؛ مثلًا جاوا اسکریپت یا استایل شیت و امثال آن. سپس زمانیکه محتوا را Response.Write می‌کند، هدر مربوط به کش شدن آن را نیز به 30 روز تنظیم می‌نماید.

تابع مربوط به کش کردن اطلاعات از مقاله [ASP.NET Ajax Under-the-hood Secrets](#) استخراج شد.

روش استفاده در مورد فایل‌های CSS

بجای تعریف یک فایل CSS در صفحه، به صورت استاندارد، اکنون تعریف متدائل را به صورت زیر اصلاح کنید:

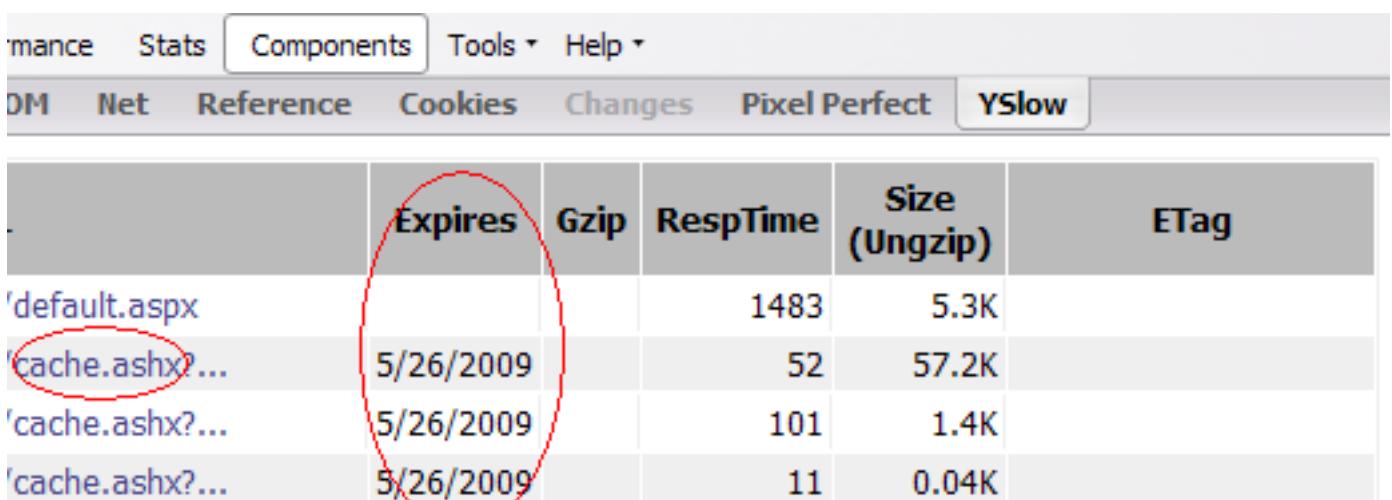
```
<link type="text/css" href="cache.ashx?v=1&file=site.css&contetType=text/css" rel="stylesheet" />
```

هر زمانیکه که فایل site.css درخواست می‌شود، باید از فیلتر ما عبور کند و سپس ارائه گردد. در این حین، هدر مربوط به مدت زمان کش شدن سمت کلاینت به آن اضافه می‌شود. از کوئری استرینگ مربوط v هم جهت به روز رسانی‌های بعدی استفاده می‌شود تا اگر تغییری را اعمال کردیم، کلاینت حتماً با توجه به آدرس جدید، محتویات جدید را یکبار دیگر دریافت کند. (مرورگر آدرس‌های مشابه را در صورتیکه هدر مربوط به کش شدن آن‌ها تنظیم شده باشد، از کش خواهد خواند و کاری به آخرین تغییرات شما در سرور ندارد)

روش استفاده در مورد فایل‌های JS

```
<script type="text/javascript" src="cache.ashx?v=1&file=js/jquery-1.3.2.min.js&contetType=application/x-javascript"></script>
```

اکنون اگر سایت را مجدداً با افزونه YSlow بررسی کنیم، می‌توان این هدر جدید را مشاهده کرد:



	Expires	Gzip	RespTime	Size (Ungzip)	ETag
'default.aspx'			1483	5.3K	
'cache.ashx?...'	5/26/2009		52	57.2K	
'cache.ashx?...'	5/26/2009		101	1.4K	
'cache.ashx?...'	5/26/2009		11	0.04K	

نظرات خوانندگان

نویسنده: Anonymous
تاریخ: ۲۱:۲۴:۵۴ ۱۳۸۸/۰۴/۲۵

خیلی عالی بود. ممنون

نویسنده: mahdi
تاریخ: ۱۸:۵۸:۵۸ ۱۳۸۸/۱۰/۰۱

با سلام و تشکر
روش کش کردن background-image
در css به چه صورت است

نویسنده: وحید نصیری
تاریخ: ۱۹:۳۹:۲۶ ۱۳۸۸/۱۰/۰۱

تصاویر اگر آدرسشان عوض نشود عموما کش می‌شوند اما الزاما هم ندارد.
برای اطمینان حاصل کردن از این موضوع برای تصاویر هم مطابق کدهای فوق یک generic handler با نکات مربوط به کش کردن اطلاعات که ذکر شد ایجاد کنید. به این صورت هر نوع تصویر مورد نظر شما به همراه هدرهای لازم کش شدن ارائه می‌شوند.
تصویر مثل کدی است که در مقاله زیر آمده:
http://www.dotnettips.info/2009/05/blog-post_27.html
فقط این مقاله چند سطر فوق مربوط به اضافه کردن هدرهای کش شدن اطلاعات را ندارد.

نویسنده: مهدی پایرونده
تاریخ: ۱۴:۰۲:۱۷ ۱۳۸۹/۰۲/۱۲

ورژن های پایین در تب HTTP Headers گزینه Enable content expiration وجود دارد آیا این گزینه در صورت استفاده از یک هاست اختصاصی همان امکانات را برای سمت کلاینت انجام میدهد.

نویسنده: وحید نصیری
تاریخ: ۱۴:۲۸:۲۵ ۱۳۸۹/۰۲/۱۲

http://www.dotnettips.info/2009/05/aspnet_17.html

این مثال شبیه به مثال [بررسی وجود نام کاربر با استفاده از jQuery Ajax](#) است که از ذکر توضیحات مشابه آن، در اینجا خودداری خواهد شد.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="TestBrokenImages.aspx.cs"
Inherits="testWebForms87.TestBrokenImages" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>detecting broken images</title>

    <script src="jquery.min.js" type="text/javascript"></script>

    <script type="text/javascript">
        function errorReplace(arg) {
            ارسال پیغام خطای
            $.ajax({
                type: "POST",
                url: "TestBrokenImages.aspx/GetErros",
                data: "{'image': '" + arg.src + "','"page':'" + location.href + "'}",
                contentType: "application/json; charset=utf-8",
                dataType: "json"
            });
            نمایش تصویری دلخواه بجای نمونه مفقود
            $(arg).attr('src', 'missing.png');
        }
    </script>
    بررسی وضعیت تک تک تصاویر پس از بارگذاری کامل صفحه
    $(document).ready(function() {
        $(window).bind('load', function() {
            $('img').each(function() {
                if (!this.complete || (!$.browser.msie && (typeof this.naturalWidth == "undefined"
|| this.naturalWidth == 0))) {
                    errorReplace(this);
                }
            });
        });
    });
</script>

</head>
<body>
    <form id="form1" runat="server">
        <div>
            
            
        </div>
    </form>
</body>
</html>
```

```
using System;
using System.IO;
using System.Web.Services;

namespace testWebForms87
{
    public partial class TestBrokenImages : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
    }
}
```

```
public static void GetErros(string image,string page)
{
    {
        ارسال ایمیل به مسؤول سایت و یا ذخیره خطاهای در دیتابیس// 
    }
}
```

در این مثال زمانیکه صفحه کاملا بارگذاری شد، وضعیت تک تک تصاویر بررسی می شود، اگر تصویر مفقودی وجود داشت (با اکثر مرورگرها سازگار است)، اطلاعات آن به تابع errorReplace ارسال خواهد شد.

در این تابع با استفاده از `jQuery Ajax` ، اطلاعات تصویر مفقود و صفحه مربوطه به وب متده `GetErros` ما ارسال می شود. سپس در این متده می توان یا آرگومان های دریافتی را به صورت یک ایمیل به مسؤول سایت ارسال نمود و یا آنها را جهت بررسی آتی در یک دیتابیس ذخیره کرد.

بدیهی است بجای قرار دادن وب متده فوق در صفحه جاری، می توان یک وب سرویس را نیز ایجاد و متده را در آن قرار داد تا نیازی نباشد به ازای هر صفحه سایت یکبار این متده تکرار شود.

اگر موفق به اجرای این مثال نشدید، برای مثال یک `break point` داخل متده `GetErrors` قرار دهید و برنامه را در حالت دیبیگ در ویژوال استودیو شروع کنید، اگر اتفاق خاصی رخ نداد و به این `break point` ترسیدید، احتمالا تنظیمات وب کانفیگ شما مناسب نیست. قسمت مربوط به `webServices` ، `system.web.extensions` و `jsonSerialization` باید در وب کانفیگ موجود باشد که VS 2008 این موارد را به صورت خودکار اضافه می کند.

مطلوبی را دیروز در وبلاگ آقای صحرایی دیدم در مورد [یهود کارآیی برنامه‌ها و سایت‌های ASP.NET](#) ، که یکی از موارد آن "فاسله بین تگ ها را تا حد ممکن از بین ببرید" بود.
برای پیاده سازی آن به صورت زیر می‌توان عمل کرد:

```
using System.Text.RegularExpressions;  
// حذف فاسله‌های خالی  
private static readonly Regex REGEX_BETWEEN_TAGS = new Regex(@">\s+<", RegexOptions.Compiled);  
private static readonly Regex REGEX_LINE_BREAKS = new Regex(@"\n\s+", RegexOptions.Compiled);  
  
protected override void Render(HtmlTextWriter writer)  
{  
    using (HtmlTextWriter htmlwriter = new HtmlTextWriter(new System.IO.StringWriter()))  
    {  
        base.Render(htmlwriter);  
        string html = htmlwriter.InnerWriter.ToString();  
  
        html = REGEX_BETWEEN_TAGS.Replace(html, "> <");  
        html = REGEX_LINE_BREAKS.Replace(html, string.Empty);  
  
        writer.Write(html.Trim());  
    }  
}
```

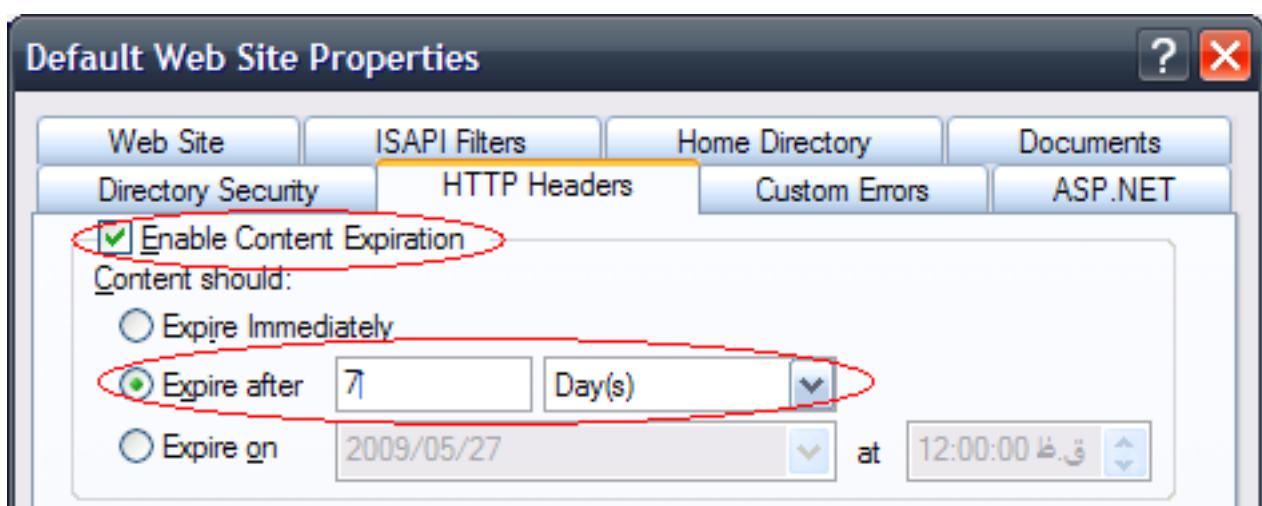
این قطعه کد را در مسترپیج سایت خود قرار دهید تا به صورت یکسان به کل سایت اعمال شود.

یا روش دیگر اعمال آن سفارشی ساختن `ASP.NET pipeline` با کمک `Response.Filter` آن است. برای مشاهده پیاده سازی آن لطفاً به مقاله زیر مراجعه بفرمائید:

[Free Bandwidth For All ASP.NET Users](#)

برای بهینه سازی قسمت اعمال `regular expressions` آن می‌توان به مقاله "[چگونه Regex سریعتری داشته باشیم؟](#)" مراجعه کرد.

در دو قسمت قبل در مورد IIS7 و IIS6 صحبت شد (+ و +). در تکمیل قسمت دوم، یک مورد هم جزو قابلیت‌های ذاتی IIS6 و همچنین IIS5 است که می‌توان آن را فعال نمود (اگر دسترسی به سرور دارید):



Inspect							
	Console	HTML	CSS	Script	DOM	Net	Reference
▲ TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/M/D)
[+] doc (1)	13.7K						
[+] js (3)	45.3K						
[+] css (1)	0.6K						
css	0.6K			314	🔍	http://localhost/library/App_Themes/Main.css	2009/5/24
[+] cssimage (4)	3.0K						
cssimage	0.4K			314	🔍	http://localhost/library/App_Themes/images/bkg.jpg	2009/5/24

تنظیم مدت زمان [content expiration](#)، بدون نیاز به برنامه نویسی خاصی، کار اضافه کردن هدر مربوط به مدت زمان کش شدن سمت کلاینت را به محتویات غیرپویایی سایت شما مانند تصاویر، فایل‌های CSS و غیره انجام می‌دهد. آمارها نشان می‌دهند که این تنظیم، زمان بارگذاری بعدی را بین 50 تا 70 درصد کاهش می‌دهد. تنظیم این قابلیت را می‌توانید به چک لیست نصب IIS خود اضافه نمایید.

نظرات خوانندگان

نویسنده: نیما
تاریخ: ۱۳۸۸/۰۲/۲۸ ۱۶:۵۳:۰۰

سلام جناب نصیری
این تنظیمات بیشتر برای سایتهاست که محتوای غیر پویا دارند مناسب نیست؟
چون اگر که اطلاعات ما زود به زود آپدیت بشه ممکنه اطلاعات قدیمی به کاربر نشون داده بشه درسته؟
باتشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۲/۲۸ ۱۷:۲۳:۰۰

روی محتوای پویای سایت شما اثری ندارد. فقط تصاویر، `css` و فایل‌های جاوا اسکریپت و امثال آن را کش می‌کند (که عموماً پویا نیستند). بنابراین برای خیلی از سایتها مفید است.
و همچنین در خیلی از سایتها هم تغییرات `css` یا تصاویر اصلی سایت، شاید ماهی یکبار باشد. زمانیکه طراحی قالب و یک سری از موارد ثابت سایت تمام شد، این‌ها دیگر هر روز قرار نیست تغییری کنند.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۱/۱۱ ۱۳:۳۰:۳۴

؟؟ بله. این قسمت سوم مربوط است به IIS است. در قسمت‌های دیگر هم شرایط مرتبط دقیقاً ذکر شده است.

نویسنده: ehsan
تاریخ: ۱۳۸۹/۰۱/۱۱ ۱۵:۵۷:۲۸

استاد قسمت چهارم چی؟
من کلاس‌ها رو نوشتم و در `web config` هم اضافه کردم ولی `expire time` برای `css`‌ها تنظیم نمی‌شے!
میشه کمک کنید

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۱/۱۱ ۱۶:۳۱:۵۴

برنامه رو دیباگ کنید ببینید اصلاً متدهای `context_PresendRequestHeaders` صدا زده می‌شود؟
کنید ببینید شاید جایی مشکلی هست.
ضموناً من این رو با IIS7 تست کردم و نتایج قسمت 4 با استفاده از IIS7 منتشر شده.
ولی در کل دیباگ کردن این کدها بهترین روش برای پیدا کردن مشکل است.

عموماً کاربران نمی‌توانند گزارش خطای خوبی را ارائه بدهند و البته انتظاری هم از آنان نیست. تنها گزارشی که از یک کاربر دریافت می‌کنید این است: "برنامه کار نمی‌کنه!" و همین! روش‌های متعددی برای لاغ کردن خطاهای یک برنامه ASP.NET موجود است؛ چه خودتان آن‌ها را توسعه دهید و یا از [health monitoring](#) استفاده کنید.

روش دیگری که این روزها در وبلاگ‌های متعددی در مورد آن مطلب منتشر می‌شود، استفاده از ELMAH است. (البته ELMAH به تازگی منتشر نشده ولی تا کیفیت محصولی به عموم ثابت شود مدتی زمان می‌برد)

یک ماژول رایگان و سورس باز لاغ کردن خطاهای مدیریت نشده برنامه‌های ASP.NET است. برای استفاده از این ماژول نیازی نیست تا تغییری در برنامه خود ایجاد کنید یا حتی آن را کامپایل مجدد نمایید. یک فایل `d11` دارد به همراه کمی تغییر در `web.config` برنامه جهت معرفی آن و این تمام کاری است که برای برپایی آن لازم است صورت گیرد. این ماژول تمامی خطاهای مدیریت نشده‌ی برنامه شما را لاغ کرده (در حافظه سرور، در یک فایل `xml`، در یک دیتابیس اس کیوال سرور یا اوراکل، در یک دیتابیس اکسس و یا در یک دیتابیس اس کیوال لایت) و برای مرور آن‌ها یک صفحه‌ی وب سفارشی یا فیدی مخصوص را نیز در اختیار شما قرار می‌دهد. همچنین این قابلیت را هم دارد که به محض بروز خطای یک ایمیل را نیز به شما ارسال نماید.

با توجه به این که این ماژول در Google code قرار گرفته احتمالاً دسترسی به آن مشکل خواهد بود. سورس و فایل‌های کامپایل شده آن را از آدرس‌های زیر نیز می‌توان دریافت نمود:

([+](#) [+](#) [+](#))

نحوه استفاده از ELMAH :

برای استفاده از ELMAH دو کار را باید انجام دهید:
 الف) کپی کردن فایل `Elmah.d11` در پوشه `bin` برنامه
 ب) تنظیم وب کانفیگ برنامه

بهترین مرجع برای آشنایی با نحوه بکار گیری این ماژول، مراجعه به فایل `web.config` موجود در پوشه `samples` آن است. بر اساس این فایل نمونه:

- ابتدا باید `configSections` آن را به وب کانفیگ خود اضافه کنید.

- سپس تگ `elmaht` باید اضافه شود. در این تگ موارد زیر مشخص می‌شوند:

الف) آیا خطاهای توسط آدرس `elmaht.axd` توسط کاربران راه دور قبل مشاهده شود یا خیر.
 ب) خطاهای کجا ذخیره شوند؟ موارد زیر پشتیبانی می‌شوند:

دیتابیس‌های اس کیوال سرور، اوراکل، حافظه سرور، فایل‌های `xml`، دیتابیس اکسس و یا دیتابیسی از نوع `VistaDB`

ج) آیا خطاهای ایمیل هم بشوند؟ اگر بله، تگ مربوطه را تنظیم کنید.

د) آیا خطاهای به اکانت `twitter` شما نیز ارسال شوند؟

- در ادامه تگ مربوط به معرفی این `httpModules` باید تنظیم شود.

- سپس `elmah.axd` که جهت مروء خطاها می‌توان از آن استفاده نمود معرفی می‌گردد.

- از IIS7 استفاده می‌کنید؟ قسمت `system.webServer` را نیز باید اضافه نمایید.

- و در آخر نحوه دسترسی به `elmah.axd` مشخص می‌شود. اگر اجازه دسترسی از راه دور را داده باشد، به این طریق می‌شود دسترسی را فقط به کاربران مجاز و تعیین اعتبار شده، اعطاء کرد و یا به نقشی مشخص مانند ادمین و غیره.

برای نمونه، اگر بخواهید از دیتابیس SQLite جهت ذخیره سازی خطاهای حاصل شده استفاده نمایید و نیز از ارسال ایمیل صرفنظر کنید، وب کانفیگ برنامه شما باید به شکل زیر تغییر یابد:

```
<?xml version="1.0"?>
<configuration>
<configSections>
  <sectionGroup name="elmah">
    <section name="security" requirePermission="false" type="Elmah.SecuritySectionHandler, Elmah"/>
    <section name="errorLog" requirePermission="false" type="Elmah.ErrorLogSectionHandler, Elmah" />
    <section name="errorMail" requirePermission="false" type="Elmah.ErrorMailSectionHandler, Elmah" />
    <section name="errorFilter" requirePermission="false" type="Elmah.ErrorFilterSectionHandler, Elmah"/>
    <section name="errorTweet" requirePermission="false" type="Elmah.ErrorTweetSectionHandler, Elmah"/>
  </sectionGroup>
</configSections>

<elmah>
  <security allowRemoteAccess="1" />
  <errorLog type="Elmah.SQLiteErrorLog, Elmah" connectionStringName="cn1" />
</elmah>

<appSettings/>

<connectionStrings>
  <add name="cn1" connectionString="data source=~/ErrorsLog/Errors.db" />
</connectionStrings>

<system.web>
  <httpModules>
    <add name="ErrorLog" type="Elmah.ErrorLogModule, Elmah"/>
  </httpModules>

  <httpHandlers>
    <add verb="POST,GET,HEAD" path="elmah.axd" type="Elmah.ErrorLogPageFactory, Elmah" />
  </httpHandlers>

  <compilation debug="true"/>

  <authentication mode="Windows" />
</system.web>

</configuration>
```

در اینجا یک پوشه جدید به نام `ErrorsLog` را باید به ریشه سایت خود اضافه کنید (یا هر نام دلخواه دیگری که در قسمت `connectionStrings` باید تنظیم شود). فایل `Errors.db` به صورت خودکار ایجاد خواهد شد. بدیهی است کاربر ASP.net باید دسترسی `write` بر روی این پوشه داشته باشد تا عملیات ثبت خطاهای با موفقیت صورت گیرد. همچنین فایل `System.Data.SQLite.DLL` نیز باید در پوشه `bin` برنامه شما کپی شود.

ساده‌ترین تنظیم این مژول استفاده از حالت `xml` است که به ازای هر خطایک فایل `xml` را تولید کرده و نیاز به اسمبلی دیگری بجز مژول مربوطه نخواهد داشت.

تذکر:

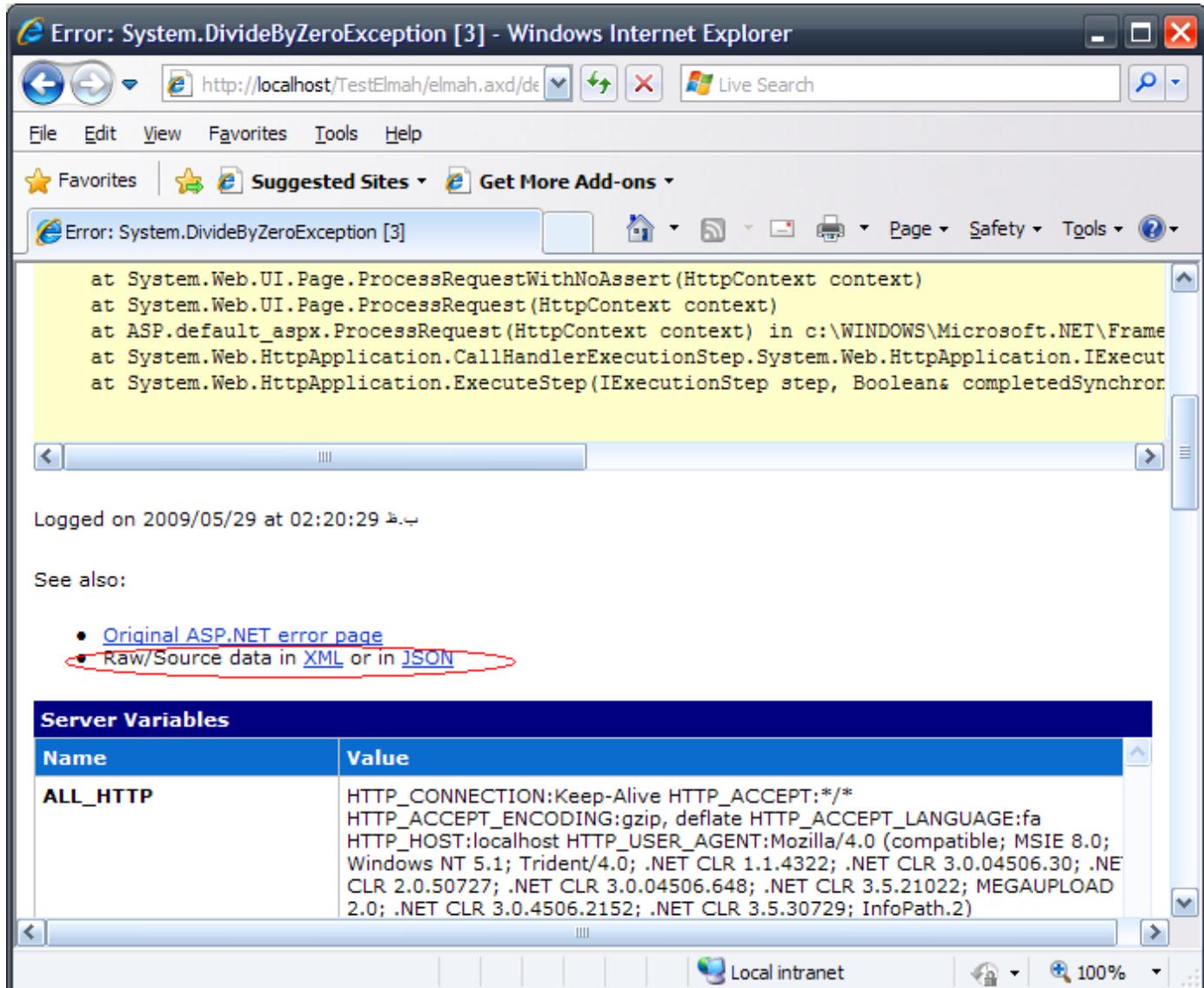
از لحاظ امنیتی مثال فوق توصیه نمی‌شود زیرا آن 1 است و قسمت `authorization` ذکر نشده است. این مثال فقط جهت راه اندازی و آزمایش اولیه ارائه گردیده است. (همچنین بهتر است این نام پیش فرض را به نامی دیگر مثلاً `myloggermdl.axd` تغییر داده و در قسمت `httpHandlers` تنظیم نمایید. سپس این نام را به تگ `location` نیز اضافه کنید)

اکنون برای مشاهده خروجی این مژول به انتهای آدرس سایت خود، elmah.axd را اضافه کرده و سپس enter کنید:

Host	Code	Type	Error	User	Date	Time
VAHID	500	DivideByZero	Attempted to divide by zero. Details...		2009/05/29	02:20
VAHID	500	DivideByZero	Attempted to divide by zero. Details...	VAHID\vhahid	2009/05/23	10:15
VAHID	500	DivideByZero	Attempted to divide by zero. Details...	VAHID\vhahid	2009/05/23	10:14

Powered by [ELMAH](#), version 1.0.11211.0. Copyright (c) 2004-9, Atif Aziz. All rights reserved. Licensed under [Apache License, Version 2.0](#). Server date is Friday, 29 May 2009. Server time is 14:21:41. All dates and times displayed are in the Iran Daylight Time zone. This log is provided by the SQLite Error Log.

همانطور که در تصویر مشخص است، تمامی خطاهای لاغ شده گزارش داده شده‌اند. همچنین دو نوع فید به همراه امکان دریافت خطاهای به صورت CSV نیز موجود است. با کلیک بر روی لینک details، صفحه‌ی بسیار ارزش‌های ارائه می‌شود که تقریباً نحوه‌ی وقوع ماجرا را بازسازی می‌کند.
نکته‌ی مهمی که در صفحه‌ی جزئیات ارائه می‌شود (علاوه بر stack trace و مشخصات کاربر)، مقادیر تمامی فیلد‌های یک صفحه هنگام بروز خطأ است (قسمت Raw/Source data in XML or in JSON در این صفحه):



به این صورت دیگر نیازی نیست از کاربر بپرسید چه چیزی را وارد کرده بودید که خطا حاصل شد. دقیقاً مقادیر فیلدهای همان صفحه‌ی زمان بروز خطا نیز برای شما لاغ می‌گردد.

نکته:

ماژول SQLite ای که به همراه مجموعه ELMAH ارائه می‌شود 32 بیتی کامپایل شده (64 بیتی آن نیز موجود است که باید از آن در صورت لزوم استفاده شود). بنابراین برای اینکه در یک سرور 64 بیتی به مشکل برنخورید و خطای BadImageFormat را دریافت نکنید نیاز است تا به [این نکته](#) دقت داشت.

برای مطالعه بیشتر:

[Error Logging Modules And Handlers](#)

[Sending ELMAH Errors Via GMail](#)

[Exception-Driven Development](#)

[Using HTTP Modules and Handlers to Create Pluggable ASP.NET Components](#)

نظرات خوانندگان

نویسنده: افشار محبی
تاریخ: ۱۳۸۸/۰۳/۰۹ ۱۰:۱۳:۵۱

جناب نصیری

شما واقعا ابزارهای جالب و مفیدی را می‌شناسید. خیلی از آنها کار بندۀ را راه انداخته است!

نویسنده: Mohsen
تاریخ: ۱۳۸۹/۰۵/۲۲ ۱۲:۵۰:۰۵

ممnon
جالب بود

نویسنده: mmshahshahani
تاریخ: ۱۳۸۹/۰۵/۲۴ ۱۱:۲۶:۱۸

خیلی حال داد ، ممنون

نویسنده: maysam.sa
تاریخ: ۱۳۸۹/۰۵/۲۵ ۲۳:۳۰:۲۵

سلام

با تشکر از مطالب جالب و آموزنده شما
میشه دوباره یه لینک برای دانلود بذارید؟
لینک قبلی خراب شده

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۵/۲۵ ۲۳:۴۵:۱۹

<http://rapidshare.com/files/412297366/ELMAH-1.1-bin.zip.html>

نویسنده: Majid
تاریخ: ۱۳۸۹/۰۶/۰۸ ۱۴:۲۰:۰۴

سلام و درود خدمت آقای نصیری:
معرفی شده یکی از پرکاربردترین و دوست داشتنی ترین مواردی که توسط استاد بندۀ (آقای مهدی کرامتی) معرفی (ارجاع به وبلاگ شما) و از اینجا مورد استفاده قرار گرفت.
به نظر در پیاده سازی و عملکرد elmah در پروژه هیچ مشکلی وجود ندارد ...
اما مشکلی در مورد ارسال ایمیل وجود دارد که خدمتتان عرض می کنم.
در حالت local ایمیل به زیبایی هر چه تمام تر ارسال می شود ...
اما پس از آپلود سایت در ارسال ایمیل مشکل وجود دارد !
به نظر شما چگونه باید این مشکل را حل نمایم ؟
این مشکل به سرور مربوط می شود یا به web.config

من قسمت نظرات رو هر روز بررسی می کنم تا بلکه پاسخی از طرف شما در یافت نمایم و یا اینکه پاسخ را بدون زحمت به آدرس زیر ایمیل بزنید:
majid.darab@gmail.com

با تشکر فراوان از وب لاغ تخصصی حضر تعالی

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۶/۰۸ ۱۵:۴۱:۰۸

سلام

تمام هاستها برای ارسال ایمیل نیاز به smtp authentication دارند. برای این منظور باید یوزر نیم و پسورد داده شده به شما را در web.config برنامه اضافه کنید که در این مقاله توضیح داده شده است:

<http://www.4guysfromrolla.com/articles/072606-1.aspx>

نویسنده: Majid
تاریخ: ۱۳۸۹/۰۶/۰۸ ۱۶:۱۷:۵۹

با تشکر از پاسخگویی شما...
یعنی authentication زیر که مربوط به تنظیمات خود elmah بود کفایت نمی کند و باید در web.config آنوتنتیکیشن دیگری هم تعریف نمایم:

من کمی گیج شدم ...

همان طور که گفتم در local این موضوع به خوبی کار می کنداما پس از آپلود مشکل بوجود آمد.

من از داخل کدبیهایند (بدون دست کاری در web.config) هم سیستم ارسال ایمیل رو دارم و این قسمت پس از آپلود به خوبی کار می کند.
تقریباً مشابه همون عملکرد و در مورد elmah (طبق فرمول بالا) که در local درست و پس از آپلود مشکل دار شد.

اگر لازم است در web.config کدی اضافه نمایم پس چرا ایمیل داخل کدبیهایند داره به خوبی کار می کنه

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۶/۰۸ ۱۸:۱۷:۴۵

در ASP.NET برای تعريف authentication smtp می توانید که این حالت نیازی به تعريف همان موارد در web.config ندارد.

یا این تنظیمات رو مطابق مقاله ای که ذکر کردم می توانید به web.config انتقال بدید (قسمت -- system.net -- mailSettings) و نیازی به تعريف مجدد آنها در code behind نیست و به صورت سراسری به برنامه اعمال می شود. یکی از این دو حالت رو انتخاب کنید.

در مطابق مقاله elmah در پوشه samples web.config اضافه errorMail باید قسمت comment شود که در آن مقاله البته comment شده که باید آن برداشته شود.

سپس elmah هم چون از همان کدهای متداول دات نتی استفاده می کند تنظیمات smtp authentication را که بحث مجزایی است می تواند از web.config هم دریافت کند. راه دیگر هم این است که این یوزر نیم و پسورد را در همان تگ errorMail هم مطابق مثل web.config موجود در پوشه samples آن هم می شود تعريف کرد. تمام اینها یک اثر را دارند و فقط یکی باید اعمال شود.

ضمنا یک مطلب رو هم به صراحة خدمت شما عرض کنم. تابحال هاستی رو ندیدم که با همان تنظیمات local شما بدون مشکل کار کند. علت هم این است که کسی نتواند از امکانات ارسال ایمیل آنها سوء استفاده کند. بنابراین بر روی هاست حتما نیاز به smtp authentication هست. یا این اطلاعات را باید از هاست بگیرید. یا اگر کنترل پنل ارسال ایمیل هم سایت دارد یک یوزر جدید در آن تعريف کنید و مشخصات آن را در برنامه استفاده کنید. در غیر اینصورت آن هاست از لحاظ امنیتی به شدت مشکل دارد و دیر یا زود زیر بار ارسال میلیون ها اسپم از پا در خواهد آمد.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۶/۰۸ ۱۸:۳۴:۳۵

برای تکمیل مبحث هم حتماً یکبار فایل web.config آن را مفصل بررسی کنید و یک سری کامنت دارد که باید بخونید.

برای مثال اگر هاست شما از IIS7 به بعد استفاده می‌کند قسمت system.webServer فایل web.config شما باید شامل Elmah.ErrorMailModule شود در غیر اینصورت ایمیلی ارسال نخواهد شد.

این موارد دقیقاً در این فایل راهنمای همراه مجموعه موجود است.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۶/۰۸ ۱۸:۴۶:۴۸

خلاصه بحث ارسال ایمیل:

- امکان ندارد هاست شما برای ارسال ایمیل نیاز به smtp authentication نداشته باشد (من ندیدم). بنابراین یا این اطلاعات را باید زنگ بزنید و از هاست بگیرید یا در کنترل پنل ایمیل‌های سایت یک یوزر مخصوص را درست کنید تا بشود در برنامه از آن استفاده کرد.
- سپس تگ errorMail باید دقیقاً مقدار دهی شود. آدرس و نام کاربری وغیره. (این موارد مرتبط به ارسال ایمیل است و با نام کاربری مثلاً ftp سایت متفاوت می‌باشد)
- در IIS6 : باید در قسمت system -- httpModules ، مطابق مثال sample web.config پوشش کتابخانه، قسمت Elmah.ErrorMailModule را از کامنت خارج کنید.
- در IIS7 : باید در قسمت system.webServer شما Elmah.ErrorMailModule فایل web.config نیز مطابق مثال یاد شده ، ذکر شده باشد.

نویسنده: Majid
تاریخ: ۱۳۸۹/۰۶/۰۸ ۲۱:۲۰:۰۵

از اینکه با صبر و حوصله پاسخگوی مشکل مطرح شده بودید از شما به شدت قدر دانی می‌کنم و فکر کنم فهمیدم مشکل از کجا بود!

شما مورد ایمیل را کاملاً مو شکافی کردید و فکر می‌کنم حق مطلب در مورد آن ادا شده است.

من تنظیمات الماه را براساس iis سرور 2003 گذاشته بودم (قاعده‌تاً باید 6 می‌بود) و الان که ورژن iis رو دیدم ، با تعجب متوجه شدم شده ورژن 76,0 .

حالا اینکه این ورژن 6 یا 7 است با تغییر کانفیگ الماه مشخص خواهد گردید.

درود فراوان بر وب لاغ شما

نویسنده: Majid
تاریخ: ۱۳۸۹/۰۶/۰۹ ۰۱:۳۳:۱۴

سلام آقای نصیری :

نزدیک به 2 ساعت هر جور تغییری که بگید در web.config دادم اما نشد که نشد.

دایمیاً کامنت ها رو بر می‌داشتیم و دوباره می‌ذاشتیم.

من به سرور دسترسی remote کامل دارم و از این طریق حالت های مختلف web.config رو بررسی کردم.
(team viewer)

من یک ایمیل توسط گوگل ایجاد کردم و از اون برای ارسال ایمیل استفاده می‌کنم.

تنها تنظیماتی که انجام دادم در خود تگ error mail الما بود و بس.

وقتی پروژه در local کار می کنه ایمیل ارسال میشه ، اما وقتی پابلیش میشه میره رو سرور نمی دونم چرا کار نمی کنه؟

ویندوز سرور هم 2003 است و ورژن iis اوتو زده 76,0

فایروال سرور هم از کار انداختم و یه بار تست کردم ، اما از ایمیل خبری نشد.

آیا منظور از authenticate کردن این است که :

1-

یا در تگ error mail تنظیمات کانکت شدن به سروری که ایمیل مربوط به آن است را انجام دهیم (شامل user name و پسورد) که در این مورد gmail مورد نظر می باشد!

2-

یا در جای دیگری از وب کافیگ این تعاریف را انجام دهیم که برای کل پروژه عمومیت داشته باشه!

3-

یا در کد بیهایند جداگانه برای هر ایمیل زدن انجام دهیم!

این سوالو برای این پرسیدم چون گفته بودید از هاست خود user name و پسورد بگیرید.
این user name و پسورد مربوط به چیست ؟

از آنجا که سروری که من دارم استفاده می کنم تازه راه اندازی شده لذا مفاهیمی به نام mail server یا کنترل پنل برای آن وجود ندارد و فقط یک وب سرور است با یک دامین ثبت شده که به خوبی کار می کند.

کجای کار می لنگد؟

نویسنده: وحید نصیری

۰۹:۲۲:۰۹ ۱۳۸۹/۰۶/۰۹

تاریخ:

GMail نیاز به SSL دارد و پورت آن هم متفاوت است. برای تنظیمات مخصوص ELMAH در این مورد ویژه (که بهتر بود از ابتدا به آن اشاره می کردید) به مقاله زیر مراجعه کنید:

http://dotnetslackers.com/ASP_NET/re-205775_Sending_ELMah_Errors_Via_GMail.aspx

نویسنده: سیدخلیل

۹:۳۵ ۱۳۹۱/۰۴/۱۰

تاریخ:

سلام

آقای نصیری واقعا از مطالب مفیدتون تشکر میکنم
بی زحمت میشه لینک جدید دانلود elmah را بذارید؟
چون لینکها منقضی شدن!
ممnon

نویسنده: وحید نصیری

۹:۴۴ ۱۳۹۱/۰۴/۱۰

تاریخ:

نیازی نیست. به سادگی و بدون مشکل بسته بودن سورس اصلی، از NuGet استفاده کنید:

<http://nuget.org/packages/elmah>

نویسنده: سیروان عفیفی

۲۱:۴۶ ۱۳۹۱/۰۴/۱۰

تاریخ:

خیلی ممنون استاد عالی بود فقط دو سوال :

- 1- آیا امکان داره ظاهر errorLog که ایمیل میشود را تغییر داد تا خواناتر شود؟
- 2- برای امنیت بیشتر امکان Hash کردن UserName و Password SMTP وجود دارد؟

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۴/۱۰ ۲۲:۵

بله. برای اطلاعات بیشتر به مقالات زیر مراجعه کنید:

[Customizing ELMAH's Error Emails](#)

[Encrypting Web.config sections in ASP.NET](#)

نویسنده: حسین

تاریخ: ۱۳۹۱/۰۶/۲۰ ۱۲:۴۲

سلام

آیا ابزاری مشابه برای winforms هم وجود دارد؟

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۶/۲۰ ۱۳:۶

این دو مطلب را با هم ترکیب کنید تا به نمونه مشابهی در WPF و Winforms برسید:

([^](#)) و ([^](#))

نویسنده: ناصر

تاریخ: ۱۳۹۱/۰۷/۲۳ ۱۲:۲۱

آقای نصیری آیا باید این تگ حتما true باشه؟

```
<compilation debug="true"/>
```

این قضیه باعث پایین آمدن راندمان وب سایت ما نمیشه. چون وقتی میزارم روی false دیگه اطلاعات دقیق خطأ رو میل نمیکنه برام؟

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۷/۲۳ ۱۲:۴۶

خیر. در زمان ارائه نهایی [ناید به این ترتیب](#) عمل شود. در مورد دریافت جزئیات بیشتر در حالت release به مقاله لینک داده شده مراجعه کنید.

نویسنده: ناصر

تاریخ: ۱۳۹۱/۰۷/۲۳ ۱۶:۳۱

ممnon:

به نظر میارویشی که گفتید تو وب فرم جواب نمیده. جون من اون release رو پیدا نکردم. من میخام زمانی که <compilation> </> هستش شماره خطأ و جزئیاتش هم برام میل بشه. آیا امکانش هست؟

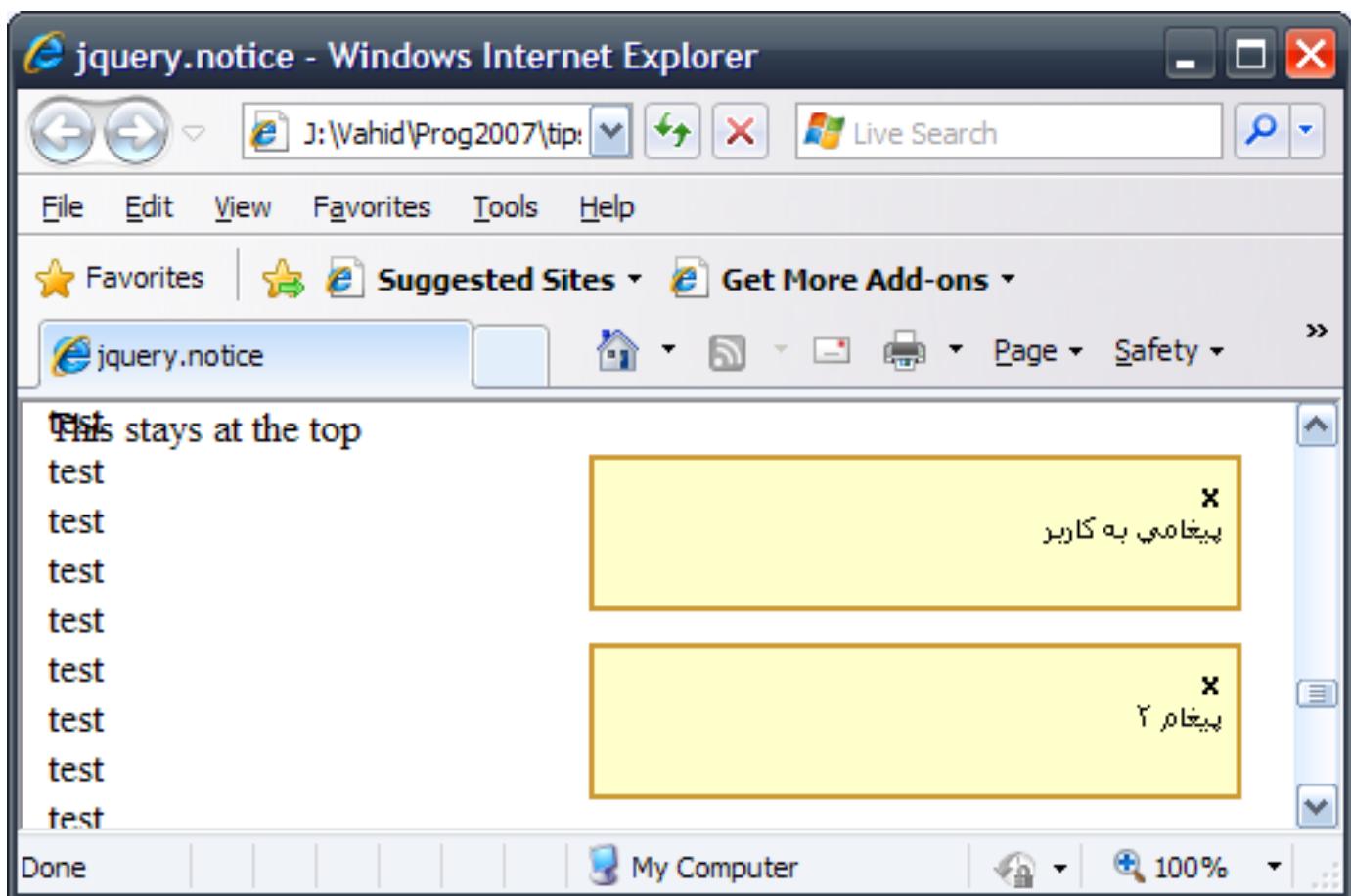
نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۰۷/۲۴ ۱۲:۲۸

خلاصه مطلب ذکر شده این است: در حالت release، باید فایل [pdb](#) منتظر با ۱۱ dll پروژه شما در کنارش باشد تا شماره خط

منتظر با استثنای رخ داده شده، قابل لاغ کردن باشد.

عموماً در برنامه‌های وب مرسوم است که پیغام به کاربر را در همان لابلای html صفحه نمایش می‌دهند. مثلًا یک برچسب و سپس تنظیم متن آن در کد برنامه به صورت پویا.
با استفاده از پلاگین‌های jQuery این کار را به صورت شکل‌تری می‌توان انجام داد. برای مثال:



پلاگین کم حجمی برای این منظور موجود است به نام [jQuery Notice](#) (یکی از چند ده نمونه موجود)

```
<script type="text/javascript">
$(document).ready(function()
{
jQuery.noticeAdd({
  text: 'پیغامی به کاربر',
  stay: false
});
});
</script>
```

کمی این پلاگین را [اصلاح کردم](#) تا مشکل عدم نمایش آن هنگام اسکرول طولانی صفحه در IE حل شود (به صورت پیش فرض با فایرفاکس مشکلی ندارد). برای مثال این div را در نظر بگیرید:

```
<div id="myElement" style="position: absolute">This stays at the top</div>
```

قصد داریم مکان آن را در بالای صفحه ثابت کیم (حتی با یک اسکرول طولانی مانند تصویر فوق، باز هم همان بالا باقی بماند و قابل مشاهده باشد).

با استفاده از jQuery این کار به صورت زیر قابل انجام است:

```
<script type="text/javascript">
$(document).ready(function()
{
$(window).scroll(function() {
    $('#myElement').css('top', $(this).scrollTop() + "px");
});
});
</script>
```

زمانیکه scroll ایی در window جاری صورت گیرد، ایی با id مساوی myElement یافت شده و سپس مقدار top آن تنظیم شده و در بالای صفحه نمایش داده می شود.

ولی این روش جهت نمایش پیغامی پویا به کاربر مشکل دارد.
نیاز است به ازای هر پیغام پویا یکبار به نحوی این اسکریپت به صفحه تزریق شود که روش انجام کار در ASP.Net به صورت زیر می تواند باشد:

```
using System;
using System.Web.UI;
using System.Web;

public class CAddJqueryNotice
{
    /// <summary>
    /// نمایش یک پیغام بر اساس پلاگین نوتیس
    /// </summary>
    /// <param name="title">عنوان</param>
    /// <param name="msg">پیغام</param>
    /// <param name="rtl">راست به چپ</param>
    /// <param name="duration">مدت زمان نمایش</param>
    /// <param name="autoHide">به صورت خودکار بسته شود؟</param>
    public static void Show(string title, string msg, bool rtl, int duration, bool autoHide)
    {
        string scriptBlock
            = string.Format(@"<script type=""text/javascript"">
                $(document).ready(function() {{
                    jQuery.noticeAdd({{
                        text: '<b>{0}</b><br/><div align=left dir={1}>{2}</div>',
                        stay: {3},
                        stayTime: {4}
                    }});
                }});
            </script>",
            title,
            (rtl ? "rtl" : "ltr"),
            msg,
            (autoHide ? "false" : "true"),
            duration);

        if (HttpContext.Current == null || HttpContext.Current.Handler == null) return;
        Page page = HttpContext.Current.Handler as Page;
        if (page != null)
            page.ClientScript.RegisterStartupScript(
                page.GetType(),
                "script" + new Guid().ToString("N"),
                scriptBlock,
                false);
    }
}
```

از آنجائیکه در یک کلاس دیگر خارج از صفحه اصلی مشغول به کار هستیم، دسترسی مستقیم به شیء Page و سپس متدها

آن جهت تزریق اسکریپت خود به صفحه نداریم. اما با استفاده از ClientScript.RegisterStartupScript می‌توان به این مقصود رسید و مشکل حل می‌شود.

برای آزمایش آن یک دکمه را در صفحه قرار داده و در روای رخداد گردن کلیک آن کد زیر را اضافه کنید:

```
CAddJqueryNotice.Show("مشکلی رخ داده است", "لطفاً دوباره سعی کنید", true, 2000, true);
```

بدیهی است قبل از استفاده از کد فوق، باید چند سطر زیر را به هدر page master سایت خود اضافه کنید:

```
<script src="jquery-1.3.2.js" type="text/javascript"></script>
<link href="jquery.notice.css" type="text/css" media="screen" rel="stylesheet" />
<script src="jquery.notice.js" type="text/javascript"></script>
```

نظرات خوانندگان

نویسنده: amin

تاریخ: ۰۵/۰۸/۲۶

نوبت داده شد

سلام . واقعا مطالب عال هستند .

اگر امکان دارد یک مثال کامل در این مورد با asp.net بگزارید .
موفق باشید .

نویسنده: مصطفی دیندار

تاریخ: ۱۳۸۹/۰۴/۲۳

نوبت داده شد

سلام ،

متاسفانه در UpdatePanel کار نمیکند

نویسنده: وحید نصیری

تاریخ: ۱۳۸۹/۰۴/۲۳

نوبت داده شد

تمکیل مبحث در اینجا:

[MS-Ajax-استفاده-از-هنگام-کاربر-در-پیغام-نمایش-به-Post/279/](#)

به ScriptManager ارائه شده در دات نت فریم ورک ۳ و نیم، سرویس پک یک، امکان ترکیب و یکی کردن اسکریپت‌های یک سایت نیز اضافه شده است. به مثال زیر دقت بفرمائید:

```
<asp:ScriptManager runat="server" ID="scMan1">
    <CompositeScript>
        <Scripts>
            <asp:ScriptReference Path("~/flash_detect.js" />
            <asp:ScriptReference Path "~/jquery.min.js" />
        </Scripts>
    </CompositeScript>
</asp:ScriptManager>
```

به صورت خودکار تمامی ScriptReference های قرار گرفته درون تگ CompositeScript ، یکی شده و توسط یک ScriptResource به صفحه اضافه خواهد شد. برای مثال:

```
<script src="/ScriptResource.axd?d=cAKu1KR-axYxbFwMRvm-I1AnHOZjn3-BvtfYzmKITFijWImCOZdjuDVGIFVqZLFX0"
type="text/javascript"></script>
```

مزایا:

- کم شدن رفت و برگشت‌ها به سرور. به این صورت مرورگر تنها یکبار درخواست دریافت اسکریپت فوق را به سرور ارسال می‌کند و نه چندین بار به ازای هر یک از اسکریپت‌های سایت.
- کش شدن خودکار اسکریپت حاصل. ([هدرهای لازم](#) اضافه می‌شوند)
- اگر مرورگر HTTP compression را پشتیبانی کند، نتیجه حاصل GZip شده ارائه می‌گردد.

و نهایتاً استفاده از این گزینه به بالا رفتن سرعت بارگذاری سایت، کمک شایانی را خواهد نمود.

[ماخذ](#)

نظرات خوانندگان

نویسنده: پیمان
تاریخ: ۱۳۸۸/۰۴/۱۳ ۱۹:۳۷:۲۴

امکان واقعا جالبی هست اما با توجه به اینکه اکثر ماهها در ایران هاست هایمان ۲. net هست درنتیجه خودمون باید این `merge` رو انجام بدیم ... مطالب این سری جالب بود لطفاروی مقالات `jquery` و `asp.net` بیشتر فوکوس کنید با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۴/۱۳ ۲۰:۲۵:۲۲

های دات نت فریم ورک رایگان هستند. اگر با هاست مکاتبه کنید مشکل لایسنس برای نصب آن خواهد داشت.

مزیت استفاده از یوزر کنترل‌ها، مازولار کردن برنامه است. برای مثال اگر صفحه جاری شما قرار است از چهار قسمت اخبار، منوی پویا، سخن روز و آمار کاربران تشکیل شود، می‌توان هر کدام را توسط یک یوزر کنترل پیاده سازی کرده و سپس صفحه اصلی را از کنار هم قرار دادن این یوزر کنترل‌ها تهیه نمود.

با این توضیحات اکنون می‌خواهیم یک یوزر کنترل ASP.Net را توسط Ajax.jQuery از بارگذاری کرده و نمایش دهیم. حداقل دو مورد کاربرد را می‌توان برای آن متصور شد:

الف) در اولین باری که یک صفحه در حال بارگذاری است، قسمت‌های مختلف آن را بتوان از یوزر کنترل‌های مختلف خواند و تا زمان بارگذاری کامل هر کدام، یک عبارت لطفاً منتظر بمانید را نمایش داد. نمونه‌ی آن را شاید در بعضی از CMS‌های جدید دیده باشید. صفحه به سرعت بارگذاری می‌شود. در حالیکه مشغول مرور صفحه جاری هستید، قسمت‌های مختلف صفحه پدیدار می‌شوند.

ب) بارگذاری یک قسمت دلخواه صفحه بر اساس درخواست کاربر. مثلاً کلیک بر روی یک دکمه و امثال آن.

روش کلی کار:

- 1) تهیه یک متد وب سرویس که یوزر کنترل را بر روی سرور اجرا کرده و حاصل را تبدیل به یک رشته کند.
 - 2) استفاده از متد Ajax جیکوئری برای فراخوانی این متد وب سرویس و افزودن رشته دریافت شده به صفحه.
- بدیهی است زمانیکه متد Ajax فراخوانی می‌شود می‌توان عبارت یا تصویر منتظر بمانید را نمایش داد و پس از پایان کار این متد، عبارت (یا تصویر) را مخفی نمود.

پیاده سازی:

قسمت تبدیل یک یوزر کنترل به رشته را قبل از مقاله " [تهیه قالب برای ایمیل‌های ارسالی یک برنامه ASP.Net](#)" مشاهده کردادید. در اینجا برای استفاده از این متد در یک وب سرویس نیاز به کمی تغییر وجود داشت KeyValueCollection‌ها درست سریالایز نمی‌شوند) که نتیجه نهایی به صورت زیر است. یک فایل Ajax.asmx را به برنامه اضافه کرده و سپس در صفحه Ajax.asmx.cs کد آن به صورت زیر می‌تواند باشد:

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Reflection;
using System.Text.RegularExpressions;
using System.Web;
using System.Web.Script.Services;
using System.Web.Services;
using System.Web.UI;
using System.Web.UI.HtmlControls;

namespace AjaxTest
{
    public class KeyVal
    {
        public string Key { set; get; }
        public object Value { set; get; }
    }

    /// <summary>
    /// Summary description for Ajax
    /// </summary>
    [ScriptService]
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    public class Ajax : WebService
    {
        /// <summary>
        /// Removes Form tags using Regular Expression
    }
}

```

```

    /// </summary>
    private static string cleanHtml(string html)
    {
        return Regex.Replace(html, @"><[/]?(<form>[^>]*?</form>)", string.Empty, RegexOptions.IgnoreCase);
    }

    /// <summary>
    /// تبدیل یک یوزر کنترل به معادل اج تی آم ال آن
    /// </summary>
    /// <param name="path"></param>
    /// <param name="properties"></param>
    /// <returns></returns>
    /// <exception cref="NotImplementedException"><c>NotImplementedException</c></exception>
    [WebMethod(EnableSession = true)]
    [ScriptMethod(ResponseFormat = ResponseFormat.Json)]
    public string RenderUserControl(string path,
        List<KeyVal> properties)
    {
        Page pageHolder = new Page();

        UserControl viewControl =
            (UserControl)pageHolder.LoadControl(path);

        viewControl.EnableViewState = false;

        Type viewType = viewControl.GetType();

        if (properties != null)
            foreach (var pair in properties)
            {
                if (pair.Key != null)
                {
                    PropertyInfo property =
                        viewType.GetProperty(pair.Key);

                    if (property != null)
                    {
                        if (pair.Value != null) property.SetValue(viewControl, pair.Value, null);
                    }
                    else
                    {
                        throw new NotImplementedException(string.Format(
                            "UserControl: {0} does not have a
                            path, pair.Key));
                    }
                }
            }

        //Form control is mandatory on page control to process User Controls
        HtmlForm form = new HtmlForm();

        //Add user control to the form
        form.Controls.Add(viewControl);

        //Add form to the page
        pageHolder.Controls.Add(form);

        //Write the control Html to text writer
        StringWriter textWriter = new StringWriter();

        //execute page on server
        HttpContext.Current.Server.Execute(pageHolder, textWriter, false);

        // Clean up code and return html
        return cleanHtml(textWriter.ToString());
    }
}

```

تا اینجا متد وب سرویسی را داریم که می‌تواند مسیر یک یوزر کنترل را به همراه خواص عمومی آن را دریافت کرده و سپس یوزر کنترل را رندر نموده و حاصل را به صورت HTML به شما تحویل دهد. با استفاده از reflection خواص عمومی یوزر کنترل یافت شده و مقادیر لازم به آن‌ها پاس می‌شوند.

چند نکته:

الف) وب کانفیگ برنامه ASP.NET شما اگر با VS 2008 ایجاد شده باشد مداخل لازم را برای استفاده از این وب سرویس توسط jQuery Ajax دارد در غیر اینصورت موفق به استفاده از آن نخواهید شد.

ب) هنگام بازگرداندن این اطلاعات با فرمت json = ResponseFormat.Json جهت استفاده در jQuery Ajax ، گاهی از اوقات بسته به حجم بازگردانده شده ممکن است خطای حاصل شده و عملیات متوقف شد. این طول پیش فرض را (maxJsonLength) در وب کانفیگ به صورت زیر تنظیم کنید تا مشکل حل شود:

```
<system.web.extensions>
  <scripting>
    <webServices>
      <jsonSerialization maxJsonLength="10000000"></jsonSerialization>
    </webServices>
  </scripting>
</system.web.extensions>
```

برای پیاده سازی قسمت Ajax آن برای اینکه کار کمی تمیزتر و با قابلیت استفاده مجدد شود یک پلاگین تهیه شده (فایلی با نام jquery.advloaduc.js) که سورس آن به صورت زیر است:

```
$ .fn.advloaduc = function(options) {
  var defaults = {
    webServiceName: 'Ajax.asmx',
    renderUCMethod: 'RenderUserControl',
    ucMethodJsonParams: '{path: "' + path + '"}',
    completeHandler: null
  };
  var options = $.extend(defaults, options);

  return this.each(function() {
    var obj = $(this);
    obj.prepend("<div align='center'>... <img src=\"images/loading.gif\"/></div>");

    $.ajax({
      type: "POST",
      url: options.webServiceName + "/" + options.renderUCMethod,
      data: options.ucMethodJsonParams,
      contentType: "application/json; charset=utf-8",
      dataType: "json",
      success:
        function(msg) {
          obj.html(msg.d);

          // if specified make callback and pass element
          if (options.completeHandler)
            options.completeHandler(this);
        },
      error:
        function(XMLHttpRequest, textStatus, errorThrown) {
          obj.html("امکان اتصال به سرور در این لحظه محدود نیست. لطفاً مجدداً سعی کنید");
        }
    });
  });
};
```

برای اینکه با کلیات این روش آشنا شوید می‌توان به مقاله "بررسی وجود نام کاربری با استفاده از jQuery در ASP.NET" مراجعه نمود که از ذکر مجدد آن‌ها خودداری می‌شود. همچنین در مورد نوشتن یک پلاگین جیکوئری در مقاله "افزونه جملات قصار jQuery" توضیحاتی داده شده است.

عمده کاری که در این پلاگین صورت می‌گیرد فراخوانی متد Ajax جیکوئری است. سپس به متد وب سرویس ما (که در اینجا نام آن به صورت پارامتر نیز قابل دریافت است)، پارامترهای لازم پاس شده و سپس نتیجه حاصل به یک شیء در صفحه اضافه می‌شود.

آن اختیاری است و پس از پایان کار متد اجکس فراخوانی می‌شود. در صورتیکه به آن نیازی نداشتید یا مقدار آن را null قرار دهید یا اصلاً آن را ذکر نکنید.

مثالی در مورد استفاده از این وب سرویس و همچنین پلاگین جیکوئری نوشته شده:

الف) یوزر کنترل ساده زیر را به پروژه اضافه کنید:

```
<%@ Control Language="C#" AutoEventWireup="true" CodeBehind="part1.ascx.cs"
Inherits="TestJQueryAjax.part1" %>
<asp:Label runat="server" ID="lblData" ></asp:Label>
```

بدیهی است یک یوزر کنترل می‌تواند به اندازه یک صفحه کامل پیچیده باشد به همراه انواع و اقسام ارتباطات با دیتابیس و غیره.

سپس کد آنرا به صورت زیر تغییر دهید:

```
using System;
using System.Threading;

namespace TestJQueryAjax
{
    public partial class part1 : System.Web.UI.UserControl
    {
        public string Text1 { set; get; }
        public string Text2 { set; get; }

        protected void Page_Load(object sender, EventArgs e)
        {
            Thread.Sleep(3000);
            if (!string.IsNullOrEmpty(Text1) && !string.IsNullOrEmpty(Text2))
                lblData.Text = Text1 + "<br/>" + Text2;
        }
    }
}
```

این یوزر کنترل دو خاصیت عمومی دارد که توسط وب سرویس مقدار دهی خواهد شد و نهایتاً حاصل نهایی را در یک لیبل در دو سطر نمایش می‌دهد.
عمداً یک sleep سه ثانیه‌ای در اینجا در نظر گرفته شده تا اثر آن را بهتر بتوان مشاهده کرد.

ب) اکنون کد مربوط به صفحه‌ای که قرار است این یوزر کنترل را به صورت غیرهمزمان بارگذاری کند به صورت زیر خواهد بود (مهمنترین قسمت آن نحوه تشکیل پارامترها و مقدار دهی خواص یوزر کنترل است):

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="TestJQueryAjax._Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>

    <script src="js/jquery.js" type="text/javascript"></script>
    <script src="js/jquery.advloaduc.js" type="text/javascript"></script>
    <script src="js/json2.js" type="text/javascript"></script>

    <script type="text/javascript">
        function showAlert() {
            alert('finished!');
        }
    </script>

    تشكيل پaramترهای متدهای وب سرویس جهت ارسال به آن/
    var fileName = 'part1.ascx';
    var props = [{ 'Key': 'Text1', 'Value': 'سطر یک' }, { 'Key': 'Text2', 'Value': '2' }];
    var jsonText = JSON.stringify({ path: fileName, properties: props });

    $(document).ready(function() {
        $("#loadMyUc").advloaduc({
            webServiceName: 'Ajax.asmx',
            renderUCMethod: 'RenderUserControl',
            ucMethodJsonParams: jsonText,
            completeHandler: showAlert
        });
    });
}
```

```
</script>  
</head>  
<body>  
  <form id="form1" runat="server">  
    <div id="loadMyUc">  
    </div>  
  </form>  
</body>  
</html>
```

نکته:

برای ارسال صحیح و امن اطلاعات json به سرور، از اسکریپت استاندارد [json2.js](#) استفاده شد.

ابزار دیبیاگ:

بهترین ابزار برای دیبیاگ این نوع اسکریپتها استفاده از افزونه فایرفاکس است. برای مثال مطابق تصویر زیر، یوزر کنترلی فراخوانی شده است که در سرور وجود ندارد:

The screenshot shows the Firebug Network panel. The 'Net' tab is selected, indicated by a red circle. Below it, the 'Response' tab is also circled in red. A POST request to 'RenderUserControl' is listed, showing a status of '500 Internal Server Error'. The response body contains a JSON string detailing the error message: "Message": "The file \u0027/monthcalendar.ascx\u0027 does not exist", "StackTrace": " at System.Web.UI.Util.CheckVirtualFileExists(VirtualPath virtualPath)\r\n at System.Web.Compilation.BuildManager.GetVPathBuildResultInternal(VirtualPath virtualPath, Boolean noBuild, Boolean allowCrossApp, Boolean allowBuildInPrecompile)\r\n at System.Web.Compilation.BuildManager.GetVPathBuildResultWithNoAssert(HttpContext context, VirtualPath virtualPath, Boolean noBuild, Boolean allowCrossApp, Boolean allowBuildInPrecompile)\r\n at System.Web.Compilation.BuildManager.GetVPathBuildResult(HttpContext context, VirtualPath virtualPath, Boolean noBuild, Boolean allowCrossApp, Boolean allowBuildInPrecompile)\r\n at System.Web.UI.TemplateControl.LoadControl(VirtualPath virtualPath)\r\n at System.Web.UI.TemplateControl.LoadControl(String virtualPath)\r\n at TestJQueryAjax.Ajax.RenderUserControl(String path, List`1 properties) in I:\\asp_net_works\\wwwroot\\1388\\AjaxTemplate\\TestJQueryAjax\\Ajax.asmx.cs:line 49", "ExceptionType": "System.Web.HttpException"}". At the bottom, it says '6 requests' and '77 KB'.

[دربیافت مثال فوق](#)

نظرات خوانندگان

نویسنده: نیما

تاریخ: ۲۲:۱۳:۴۲ ۱۳۸۸/۰۴/۱۵

سلام استاد نصیری

بسیار مقاله مفید و Trick جالبی بود. به تازگی افزونه‌ای برای فایر باگ عرضه شده به اسم Fire Query که مواردی را برای کار و دیباگ jQuery فراهم می‌کنه.

یه سوال از حضورتون داشتم چرا خروجی Execute در وب سرویس شامل تگ‌های html و body نیست؟ آیا این همیشه ثابت‌هه؟ و اینکه مواردی مثل `\n` و `\r\n` را هم از خروجی حذف کنیم بهتر نیست؟
با سپاس از شما استاد عزیز

نویسنده: Anonymous

تاریخ: ۲۲:۳۷:۳۶ ۱۳۸۸/۰۴/۱۵

سلام آقای نصیری

امکانش هست در مورد متدهای `stringify` و عملی که در اینجا انجام میده کمی توضیح بدھید؟ من اون لینک صفحه JSON رو خوندم ولی متوجه نشم چرا شما اینجا از این متدهای استفاده کردید.
ممnon

نویسنده: وحید نصیری

تاریخ: ۲۳:۱۰:۱۶ ۱۳۸۸/۰۴/۱۵

@نیما

بله. یک `trim` را هم می‌توانید اضافه کنید بعلاوه نکته حذف فاصله خالی بین تگ‌ها که در کاهش حجم نهایی مؤثر است.

@ناشناس

می‌شد این آرگومان‌های مورد نظر را به شکل یک رشته یک سطری هم در آورد (فرقی نمی‌کرد). اما با استفاده از اینتابع شما می‌توانید یک شیء جاوا اسکریپتی را تبدیل به json کنید که از این لحاظ خوب یک قدم پیشرفت است و در خطایابی کدها کمک بزرگی است (رشته قابل خطایابی نیست).

نویسنده: مهدی

تاریخ: ۱۷:۴۵:۰۹ ۱۳۸۸/۰۵/۰۹

ایده خیلی جالبی بود. اما چیزی که الان فکر من رو مشغول کرده، اینه که چطور ممکنه که از postback‌های این کنترل استفاده کرد.

اگر خودتون در این مورد کار کردن هم یک توضیحی بدین. اگر هم کاری نکردن که من خودم اگر وقت کردم یک تستی می‌کنم و اگر جواب داد به شما هم خبر میدم. نباید کار سختی باشه ولی بازم تا شروع نکنم نمی‌تونم قطعی نظر بدم.

نویسنده: وحید نصیری

تاریخ: ۱۹:۰۸:۲۳ ۱۳۸۸/۰۵/۰۹

اگر خطای viewstate is corrupted را دریافت می‌کنید به این علت است که الان در صفحه دو فیلد مخفی viewstate داریم. برای حذف این مورد دوم سطر زیر را به تابع `cleanHtml` اضافه کنید:

```
; (html = Regex.Replace(html, "<input.*name=__VIEWSTATE[^>]*>", string.Empty, RegexOptions.IgnoreCase
```

+ در این حالت چون اطلاعات کنترل‌ها به ViewState اضافه نمی‌شوند، ASP.NET Postback هم کار نمی‌کنه.

در این حالت برای ارسال دیتا و غیره امکان استفاده از وب سرویس و پاس کردن مقدار به آن هست (با استفاده از جیکوئری اجکس).

کمی بیشتر کار میبره نسبت به MS Ajax

نویسنده: sirasad
تاریخ: ۱۳۸۸/۰۵/۱۴ ۱۲:۳۲:۰۴

آقای نصیری عزیز من خواستم این کد رو در ۲.۰ net تست بکنم ، مقادیر اون تکست ها منتقل نمی شود ولی showalert رو فایرآپ میکنه .

یه مسئله دیگه اینکه این ajax شما بدون پارامتر ورودی کار میکنه ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۵/۱۴ ۱۳:۳۳:۳۷

بله. اگر پارامتر ورودی نداشت یک '{ }' قرار دهید کفايت میکنه.
برای دات نت ۲ کمی کار شما بیشتر میشود چون در VS2008 خیلی از موارد را خودکار اضافه میکند.

در این حالت یا باید ASP.NET AJAX Extensions روی سرور نصب باشد یا d11 های آنرا به صورت دستی به دایرکتوری bin برنامه اضافه کنید و یک ارجاع از آنها را نیز به برنامه اضافه کنید.

همچنین وب کانفیگ را نیز باید ویرایش کنید تا این تغییر در آن لحاظ شود. (در غیر اینصورت کار نمیکند)
یک نکته دیگر هم اینکه بجای msg.d شما فقط msg دارید و این d از ۳.۵ به بعد اضافه شده (در قسمت success و هنگام دریافت پاسخ از سرور توسط jQuery).

نویسنده: javad.mjt
تاریخ: ۱۳۸۹/۱۰/۲۱ ۱۵:۴۵:۳۸

سلام آقای نصیری
زمانی که یک دکمه یا کنترلی که postback ایجاد می کند را استفاده می کنم به خطای زیر بر می خورم
.The state information is invalid for this page and might be corrupted
Description: An unhandled exception occurred during the execution of the current web request. Please review the
.stack trace for more information about the error and where it originated in the code

Exception Details: System.Web.HttpException: The state information is invalid for this page and might be
.corrupted
لطفا راهنمایی کنید با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۰/۲۱ ۱۷:۱۵:۰۲

لطفا کامنت ها را مطالعه کنید در بالا توضیح دادم (جولای ۳۱، ۲۰۰۹، ساعت ۲۰:۰۸).
1- باید ViewState مربوط به این یوزر کنترل را خاموش کنید.
2- خاموش کردن ViewState سبب حذف فیلد مخفی آن به صورت کامل نمیشود، علاوه بر آن باید فیلد مخفی ViewState موجود در رشته را با استفاده از Regex ای که در بالا توضیح دادم کلا حذف کنید تا با ViewState اصلی صفحه تداخل نکند.

جهت بهینه سازی روش ارائه شده در مقاله " [بارگذاری یک یوزرکنترل با استفاده از جی کوئری](#)" ، می توان مبحث فشرده سازی را نیز به آن افزود.

برای این منظور نیاز است تا بتوان response حاصل را کاملا کنترل کرد و این مورد از طریق یک http module به خوبی قابل انجام است. مبحث http compression و پیاده سازی آن را احتمالا بارها در سایتها مختلف نیز دیده اید:

```
using System;
using System.IO;
using System.IO.Compression;
using System.Globalization;
using System.Web;

public class JsonCompressionModule : IHttpModule
{
    public JsonCompressionModule()
    {
    }

    public void Dispose()
    {
    }

    public void Init(HttpApplication app)
    {
        app.PreRequestHandlerExecute += new EventHandler(Compress);
    }

    private void Compress(object sender, EventArgs e)
    {
        HttpApplication app = (HttpApplication)sender;
        HttpRequest request = app.Request;
        HttpResponse response = app.Response;

        if (request.ContentType.ToLower(CultureInfo.InvariantCulture).StartsWith("application/json"))
        {
            if (!((request.Browser.IsBrowser("IE")) && (request.Browser.MajorVersion <= 6)))
            {
                string acceptEncoding = request.Headers["Accept-Encoding"];

                if (!string.IsNullOrEmpty(acceptEncoding))
                {
                    acceptEncoding = acceptEncoding.ToLower(CultureInfo.InvariantCulture);

                    if (acceptEncoding.Contains("gzip"))
                    {
                        response.Filter = new GZipStream(response.Filter, CompressionMode.Compress);
                        response.AddHeader("Content-encoding", "gzip");
                    }
                    else if (acceptEncoding.Contains("deflate"))
                    {
                        response.Filter = new DeflateStream(response.Filter, CompressionMode.Compress);
                        response.AddHeader("Content-encoding", "deflate");
                    }
                }
            }
        }
    }
}
```

در این مأذول تنها درخواست هایی از نوع application/json بررسی خواهد شد. هر چند این فشرده سازی را بر روی خروجی هر نوع WebMethod ای نیز می توان اعمال کرد. در این حالت، سطر بررسی json را حذف کرده و آن را به صورت زیر تغییر دهید:

```
if ( !request.Url.PathAndQuery.ToLower().Contains( ".asmx" ) )
```

```
return;
```

مرورگر IE6 و پایین‌تر نیز از این فشرده سازی معاف شده‌اند (چون یا پشتیبانی کاملی را ارائه نمی‌دهند یا باید منتظر کرشن مرورگر بود).

جهت اعمال این مازول به برنامه ASP.NET خود، کافی است سطر زیر را به قسمت httpModules وب کانفیگ افزود:

```
<httpModules>
  <add name="JsonCompressionModule" type="JsonCompressionModule"/>
</httpModules>
```

روش آزمایش مازول تهیه شده:

متاسفانه افزونه‌ی فایرباگ فایرفاکس اندازه‌ی نهایی response را نمایش می‌دهد و در گزارش آن حتی خبری از Content-encoding اضافه شده نیز نخواهد بود. بنابراین برای بررسی این روش مناسب نیست.
ابزار دیگری که اساساً برای این نوع آزمایشات طراحی شده است، برنامه معروف [فیدل](#) می‌باشد (که توسط مدیر پروژه تیم IE برنامه نویسی شده است).
برای استفاده از فیدل جهت دیباگ درخواست‌های local باید یک نکته‌ی کوچک را رعایت کرد:

/http://localhost .:25413

همانطور که در URL فوق مشاهده می‌کنید یک نقطه پس از localhost اضافه شده است تا خروجی محلی مربوطه قابل بررسی شود.

Fiddler - HTTP Debugging Proxy

File Edit Rules Tools View Extensions neXpert Help

Comment Reissue Remove Resume All Streaming Find Save Launch IE Clear Cache

Web Sessions <>

#	Result	Protocol	Host	URL
11	200	HTTP	localhost...	/testajax/Ajax.asmx/R...

neXpert Statistics Inspectors FiddlerScript

Headers TextView WebForms HexView Auth

No Proxy-Authorization Header is present.

No Authorization Header is present.

Response is encoded and may need to be decoded.

Transformer Headers TextView ImageView XML

Response Headers

HTTP/1.1 200 OK

- Cache
 - Cache-Control: private, max-age=0
 - Date: Mon, 13 Jul 2009 16:48:02 GMT
- Entity
 - Content-Length: 1196
 - Content-Type: application/json; charset=utf-8
- Miscellaneous
 - Server: Microsoft-IIS/5.1
 - X-AspNet-Version: 2.0.50727
 - X-Powered-By: ASP.NET
- Transport
 - Content-Encoding: gzip

Capturing All Processes Latency: Off 1 / 1 http://localhost./testajax/Ajax.asmx/Render

مطابق تصویر فوق، هم content-encoding اضافه شده مشخص است و هم حجم پاسخ دریافتی از 40 کیلوبایت (بر اساس یک تست معمولی روی صفحه‌ای مشخص) به نزدیک یک کیلوبایت و اندی کاهش یافته است.

مطلوب "نمایش یک پیغام به کاربر در ASP.Net" را احتمالاً به خاطر دارید. این روش در حالت postback کامل و یا در حالت page_load بدون مشکل کار می‌کند. اما اگر در حال استفاده از آپدیت پنل MS Ajax باشیم، دیگر کار نخواهد کرد چون تنها قسمتی از صفحه به روز رسانی می‌شود و یک postback کامل رخ نمی‌دهد تا از شیء صفحه جهت تزریق اسکریپت خود بتوانیم استفاده نماییم. خوب‌بختانه طراحان MS Ajax به این مورد هم توجه داشته‌اند و متدهای RegisterStartupScript و RegisterStartupScriptScriptManager قرار داده‌اند. برای مثال اگر نیاز باشد در پایان عملیات یک متدهای از آپدیت پنل کمک می‌گیرد، یک alert نمایش داده شود می‌توان از متدهای زیر استفاده کرد:

```
void jsCall(string alert)
{
    string script = @"alert('" + alert + "');";
    ScriptManager.RegisterStartupScript(this, this.GetType(), "jsCall", script, true);
}
```

با توجه به این نکته، مطلب "نمایش یک پیغام به کاربر در ASP.Net" را به صورت زیر نیز می‌توان با MS Ajax سازگار کرد:

```
if (page != null)
{
    ScriptManager sMgr = ScriptManager.GetCurrent(page);
    if (sMgr != null)
    {
        ScriptManager.RegisterStartupScript(page, page.GetType(),
            "script" + new Guid().ToString("N"),
            scriptBlock, false);
    }
    else
        // همان روش قبلی/
...
}
```

فرض کنید تعیین اعتبار یکی از فیلدهای فرم نیاز به انجام محاسباتی در سمت سرور دارد و این کار را می‌خواهیم با استفاده از jQuery انجام دهیم. مشکلی که در اینجا وجود دارد، این است که در A به معنای Ajax است. یعنی زمانیکه کاربر دکمه submit را فشرد، دیگر برنامه منتظر این نخواهد شد که پاسخ کامل دریافت شود، سایر پردازش‌ها صورت گیرد و سپس فرم را به سرور ارسال نماید (شبیه به ایجاد یک ترد جدید در برنامه‌های ویندوزی). مثال زیر را در نظر بگیرید:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="TestCustomValidation.aspx.cs"
Inherits="TestjQueryAjax.TestCustomValodation" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>

<script src="js/jquery.js" type="text/javascript"></script>

<script type="text/javascript">
    function validate() {
        var number1 = $("#<%=txtNumber1.ClientID %>").val();
        var number2 = $("#<%=txtNumber2.ClientID %>").val();
        var result = false;
        $.ajax({
            type: "POST",
            url: 'AjaxSrv.asmx/ValidateIt',
            data: '{"number1":"' + number1 + '","number2":"' + number2 + '"}',
            contentType: "application/json; charset=utf-8",
            dataType: "json",
            success:
                function(msg) {
                    if (msg.d) {
                        result = true;
                        alert('بسیار خوب');
                    } else {
                        result = false;
                        alert('دباره سعی کنید');
                    }
                },
            error:
                function(XMLHttpRequest, textStatus, errorThrown) {
                    result = false;
                    alert("خطای رخ داده است");
                }
        });
        //debugger;
        return result;
    }
</script>

</head>
<body>
<form id="form1" runat="server">
<div>
    number 1 :
    <asp:TextBox runat="server" ID="txtNumber1" />
    <br />
    number 2 :
    <asp:TextBox runat="server" ID="txtNumber2" />
    <br />
    <asp:Button ID="btnSubmit" Text="Submit" UseSubmitBehavior="false" runat="server"
        OnClientClick="if(!validate()){ return false;} OnClick="btnSubmitClick" />
</div>
</form>
</body>
</html>
```

این مثال یک نوع اعتبار سنجی سفارشی را در حین submit با استفاده از وب سرویس زیر انجام می‌دهد (حاصلضرب دو عدد دریافتی را بررسی می‌کند که باید مساوی 10 باشند. البته هدف از این مثال ساده، آشنایی با نحوه انجام این نوع عملیات است که می‌تواند شامل کار با دیتابیس و غیره هم باشد. و گرنه بدیهی است این بررسی را با دو سطر کد جاوا اسکریپت نیز می‌شد انجام داد):

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Web.Script.Services;

namespace TestJQueryAjax
{
    /// <summary>
    /// Summary description for AjaxSrv
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    [ScriptService]
    public class AjaxSrv : System.Web.Services.WebService
    {
        [WebMethod(EnableSession = true)]
        [ScriptMethod(ResponseFormat = ResponseFormat.Json)]
        public bool ValidateIt(int number1, int number2)
        {
            return number1 * number2 == 10;
        }
    }
}
```

در این مثال هنگام submit ، عملیات اعتبار سنجی با توجه به وضعیت asynchronous عملیات Ajax ، تمام و کمال رخ نداده ، یا اعتبار سنجی انجام نمی‌شود و یا پیغام خطای خواهیم کرد.

راه حل چیست؟

راه حل‌های فضای بسیاری را در وب در این مورد می‌توان پیدا کرد؛ اما راه حل استاندارد آن در این حالت ویژه، استفاده از Ajax در حالت غیرهمزمان است. یعنی این فریم ورک Ajax را وادار کنیم که تا پایان عملیات موردنظر، منتظر بماند و سپس فرم را ارسال کند. برای این منظور تنها کافی است یک سطر زیر را پیش از فراخوانی تابع Ajax ، اضافه و فراخوانی نمائیم:

```
$.ajaxSetup({async: false});
```

نکته:

UseSubmitBehavior دکمه ما را به شکل زیر رندر می‌کند (دکمه به یک button معمولی (بجای حالت submit) تبدیل شده و سپس یک doPostBack را اضافه خواهد کرد):

```
<input id="btnSubmit" type="button" onclick="if(!validate()){ return false;};__doPostBack('btnSubmit','') value="Submit" name="btnSubmit"/>
```

نظرات خوانندگان

نویسنده: Anonymous
تاریخ: ۰۹:۳۲:۴۸ ۱۳۸۸/۰۵/۱۲

سلام و خسته نباشید.

جناب نصیری من کلا هدف شما از این کار رو متوجه نشدم.

یعنی شما قصد اینو دارید که تا اعتبار سنجی انجان نشده و جواب فرستاده نشده فرم منتظر بمونه ؟

نویسنده: وحید نصیری
تاریخ: ۱۰:۰۴:۵۴ ۱۳۸۸/۰۵/۱۲

سلام،

دو هدف در این مثال مد نظر بود:

الف) تا پایان عملیات اعتبار سنجی توسط عملیات Ajax ای صبر شود و فرم به یکباره به سرور ارسال نگردد.
ب) اگر فرم اعتبار سنجی نشد، به سرور submit نشود.

جزو کنترل‌هایی در ASP.Net است که نمی‌توان RequiredFieldValidator استاندارد را بر آن اعمال کرد. به عبارتی اگر نیاز بود حداقل یک آیتم چک باکس لیست حتماً توسط کاربر انتخاب شود، راه حل آماده‌ای برای آن وجود ندارد. پیاده سازی این کار با استفاده از jQuery سادگی میسر است که در ادامه آنرا مرور خواهیم کرد.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="CheckBoxListValidator.Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <script src="jquery.min.js" type="text/javascript"></script>
    <script type="text/javascript">
        //<![CDATA[
        function CheckItems(sender, args) {
            //Get the total number of selected Checkboxes
            var num = jQuery("table#&lt;%=CheckBoxList1.ClientID%&gt; input:checked").length;
            args.IsValid = num &gt; 0;
        }
        //]]&gt;
    &lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
    &lt;form id="form1" runat="server"&gt;
        &lt;div&gt;
            &lt;asp:CheckBoxList ID="CheckBoxList1" runat="server"&gt;
                &lt;asp:ListItem&gt;item1&lt;/asp:ListItem&gt;
                &lt;asp:ListItem&gt;item2&lt;/asp:ListItem&gt;
            &lt;/asp:CheckBoxList&gt;
            &lt;asp:CustomValidator ClientValidationFunction="CheckItems" ID="ValidateIt"
                runat="server" ErrorMessage="(*)" &gt; &lt;/asp:CustomValidator&gt;
            &lt;asp:Button ID="Button1" runat="server" /&gt;
        &lt;/div&gt;
    &lt;/form&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```

توضیحات:

یک استاندارد را به فرم اضافه کرده‌ایم تا توسط تابعی که به آن معرفی می‌شود، کار اعتبار سنجی سمت کاربر را انجام دهد. این تابع یا همان CheckItems مثال فوق، امضای استاندارد و آشنایی دارد. اگر تعیین اعتبار صورت گرفته باشد، باید args.IsValid در آن به true تنظیم شود یا بر عکس. اصل قضیه هم، همین یک سطر کد زیر است:

```
var num = jQuery("table#<%=CheckBoxList1.ClientID%> input:checked").length;
```

کار این سطر که از جی‌کوئری استفاده می‌کند، پیدا کردن جدولی است که ID آن مساوی آی دی سمت کلاینت چک باکس لیست ما است (ASP.Net یک چک باکس لیست را به صورت یک جدول حاوی چک باکس‌ها رندر می‌کند). سپس در همان ناحیه مشغول به جستجوی چک باکس‌هایی می‌شود که تیک خورده‌اند. نهایتاً تعداد آن‌ها را بر می‌گرداند.

نظرات خوانندگان

نویسنده: Hossein Moradinia
تاریخ: ۱۳۸۸/۰۵/۱۹ ۲۰:۳۰:۵۶

با سلام خدمت استاد نصیری
می دونم که این مطلب به پست شما ربطی نداره ولی خواهش می کنم راهنمایی کنید.

آفای نصیری مطلبی رو پیدا کردم که توسط اون میشه از توابع نوشته شده در C# در Crystal report استفاده کرد:
اما هر کاری می کنم که اون رو در بخش فرمول نویسی در بخش Additional function بینم اصلاً گزینه مربوطه (همون فایل dll) نمایش داده نمی شه.

روی چند سیستم هم این کارو انجام دادم ولی گزینه مورد نظر وجود نداشته!!!

چیکار کنم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۵/۲۰ ۰۰:۱۶:۳۰

سلام،
این dll ها (یا اسمبلی های دات نتی) باید ComVisible شوند تا در Crystal report دیده شوند. بعلاوه در تنظیمات پروژه باید تیک مربوط به Register for COM Interop در برگه build گذاشته شود (مهمن).
توضیح قدم به قدم آن در اینجا هست:

[http://msdn.microsoft.com/en-us/library/ms227603\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/ms227603(VS.80).aspx)

نویسنده: Hossein Moradinia
تاریخ: ۱۳۸۸/۰۵/۲۵ ۲۰:۴۸:۲۹

ممnon وحید جان
مشکلم حل شد.
از اینکه جواب دادید واقعاً ممنون

نویسنده: مجید مهرتاش
تاریخ: ۱۳۸۸/۰۸/۲۱ ۱۳:۳۸:۰۰

ممnon ، خیلی جالب بود !

در نگارش‌های اخیر کتابخانه jQuery (از نگارش 1.3 به بعد) متذی به نام [live](#) به آن اضافه شده است که کاربرد آن را در ادامه مروء خواهیم کرد.
ابتدا مثال زیر را در نظر بگیرید:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="TestLive.aspx.cs"
Inherits="TestJQueryAjax.TestLive" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>

<script src="js/jquery.js" type="text/javascript"></script>

<script type="text/javascript">
$(document).ready(function() {
    $('a.mylink').click(function(e) {
        var $a = $(this);
        alert($a.attr('id'));
    });

    $('a#lnkLoad').click(function(e) {
        $('#dynContent').load('live.ashx');
    });
});
</script>

</head>
<body>
<form id="form1" runat="server">
<a href="#" id='lnk1' class='mylink'>link1</a>
<br />
<a href="#" id='lnkLoad'>load .ashx</a>
<div id='dynContent'>
</div>
</form>
</body>
</html>
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;

namespace TestJQueryAjax
{
    /// <summary>
    /// Summary description for $codebehindclassname$
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    public class live : IHttpHandler
    {

        public void ProcessRequest(HttpContext context)
        {
            context.Response.ContentType = "text/plain";
            context.Response.Write("<a href='#' id='lnk2' class='mylink'>link2</a>");
        }

        public bool IsReusable
        {
```

```
        get  
        {  
            return false;  
        }  
    }  
}
```

در این مثال ساده با کلیک بر روی لینک‌هایی با کلاس mylink id آن لینک نمایش داده خواهد شد. همچنین اگر بر روی لینکی با id مساوی lnkLoad کلیک شود، محتوای پویا از یک generic handler به نام live.ashx دریافت شده و به id مساوی div ایی با dynContent اضافه می‌گردد.

این محتوای دریافتی از generic handler mylink مساوی دارد، اما این بار هر چقدر بر روی لینک اضافه شده به صفحه کلیک کنیم کار نمی‌کند. چرا؟ چون در هنگام فراخوانی document.ready، این لینک وجود نداشته و روال رخدادگردانی به آن bind نشده است.

به صورت خلاصه می‌خواهیم روال کلیک بر روی لینک‌های با کلاس mylink همیشه کار کند. (چه در مورد عناصری در صفحه که از قبل وجود داشته‌اند و چه عناصری که توسط عملیاتی Ajax آیی بعداً اضافه خواهند شد) این مشکل با معرفی متدهای live حل شده است. برای این منظور تنها کافی است که ما به صورت زیر تغییر کند:

```
<script type="text/javascript">
    $(document).ready(function() {
        $('a.mylink').live("click", function() {
            var $a = $(this);
            alert($a.attr('id'));
        });
    });

```

اکنون `jQuery` کلیه لینک‌هایی با کلاس مساوی `mylink` را که از این پس اضافه خواهند شد، به صورت `live` و زنده تحت نظر قرار می‌دهد و عکس العمل نشان خواهد داد.

نظرات خوانندگان

نویسنده: مجتبی صحرائی
تاریخ: ۱۳۸۸/۰۵/۲۶ ۱۹:۱۵:۲۳

ممnon: نکته بسیار خوبی بود

نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۰۶/۱۹ ۰:۲۴

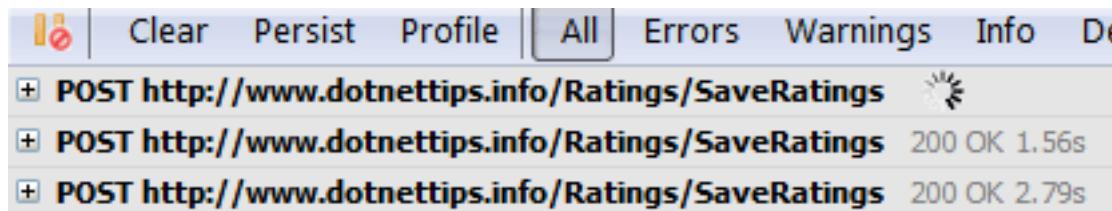
بعضی وقتا که از متده live استفاده می کنیم ، رویداد یک المنت چندبار اجرا میشه .
برای شما هم پیش اومده ؟
راه حل چیست ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۱۹ ۰:۴۱

نه این مشکل رو نداشتمن ولی راه حل اضافه کردن return false به انتهای روال رویدادگردان است.

نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۰۶/۲۴ ۱۳:۲۶

ظاهرا این مشکل در سایت جاری هم وجود دارد :



نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۴ ۱۳:۳۳

چند جایی رو تست کردم چنین چیزی مشاهده نشد. در کدام صفحه دقیقا؟ (تمام متدهای live سایت دارند)
آدرس Post یکی است؛ مقادیر Post ارسالی متفاوت (بسته به کلیک بر روی المانی خاص).

نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۰۶/۲۴ ۱۴:۱۸

هنگام امتیاز دهی به یک مطلب . مثلا همین الان 2 درخواست برای امتیازدهی همین پست ارسال شد .
در کنسول فایرباگ این کد رو اجرا کنید :

```
$(".postRating[sectiontype=BlogPost] .rating[value=5]").click();
```

نتیجه 2 درخواست بود برای من .

دفعات بعد هم شد ۳ درخواست.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۴ ۱۸:۵۴

ممنون. مشکل برطرف شد.

نگارش جدید [ELMAH](#) ارائه شده است. این نگارش کاملاً با نگارش قبلی سازگار بوده و تنها بازنویسی فایل `dll` آن با نگارش جدید کفایت می‌کند.

لیست موارد فیکس شده

- `System.Data.SQLite.dll` causes image problems on 64-bit IIS/ASP.NET
- Incorrect jscript filter sample

لیست تازه‌های آن

- Ability to set mail priority in `errorMail config` section
- Allow configuration-based filtering rules to be diffrent for e-mail and log
- Adjust behavior of `smtpPort` attribute in

دریافت از [گوگل کد](#) و یا دریافت از رپید شیر ([سورس](#) + [بایناری](#))

کنترل range validator در ASP.net امکان کنترل ورودی از نوع Int32 ، double و غیره را [فراهم می‌کند](#) . اما اگر کاربر حتما باید عددی صحیح را وارد کرده و این عدد از بازه‌ی اعداد Int32 خارج بود (مثلا اعداد Int64 یا حتی بزرگتر) این کنترل کاربرد خود را از دست می‌دهد.

در این حالت بجای استفاده از کنترل range validaotr می‌شود از RegularExpressionValidator معروف به صورت زیر استفاده کرد:

```
<asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server"  
ControlToValidate="TextBox1" ErrorMessage="RegularExpressionValidator"  
SetFocusOnError="True" ValidationExpression="\d+">></asp:RegularExpressionValidator>
```

در اینجا ValidationExpression آن عبارت باقاعده‌ای است که ورودی‌هایی را متشکل از رشته‌ای حاوی یک یا چند عدد، مجاز اعلام خواهد کرد (و در این حالت مشکلی برای تعیین اعتبار اعداد صحیح بزرگ وجود نخواهد داشت).

به نظر من jQuery به چندین دلیل از کتابخانه MS Ajax و ملحقات آن مهم‌تر است و باید به آن پرداخته شود:

- 1- داشش شما قابل انتقال است. اگر روزی به JSP یا PHP یا MJS یا موارد مشابه دیگری مهاجرت کردید، داشش jQuery شما باز هم قابل استفاده خواهد بود.
- 2- این کتابخانه بسیار سبک‌تر و کم حجم تر از MS Ajax است. (حجم آن کمتر از نصف است) (+)
- 3- تقریباً از اکثر فریم ورک‌های جاوا اسکریپت موجود سریع‌تر است. (+)
- 4- پشتیبانی آن از مرورگرهای مختلف بی‌نظیر است. لازم نیست عمر گرانمایه را صرف این کنید که چرا این اسکریپت در فایرفاکس کار می‌کنه اما در IE خیر.
- 5- کوهی از افزونه‌های مختلف برای آن موجود است. (jQuery plugin) jQuery را در گوگل جستجو کنید
- 6- استفاده از آن، میزان کد جاوا اسکریپتی را که باید نوشت تا حد بسیار قابل ملاحظه‌ای کاهش می‌دهد، که در نهایت سبب ایجاد کدی خواناتر در مدت زمانی کمتر می‌گردد.
- 7- کدنویسی با jQuery از کدنویسی JavaScript ای خالص بسیار ساده‌تر است. افرادی که تا دیروز حتی طرف جاوا اسکریپت هم نمی‌رفتند، امروزه پلاگین‌های jQuery می‌نویسند.
- 8- مستندات کاملی داشته به همراه مثال‌های بسیار کاربردی.
- 9- به دلیل استفاده شدن از آن در پلتفرم‌های مختلف، مثال‌های کاربردی بیشتری داشته، همچنین منابع آموزشی پر محتواتری را نیز می‌توان در این رابطه یافت. (می‌توان tutorial های مربوط به php را مشاهده و آن‌ها را به ASP.Net تبدیل کرد و صرفاً منحصر به یک عدد و یک پلتفرم نیست)
- 10- خلاقیت شما را نکشته و از شما فردی که هیچ‌گونه درکی نسبت به عملیات اجکسی رخ داده ندارد، نخواهد ساخت. (MS Ajax برای ASP.Net یک شاهکار است. چند عدد محصور کننده را استفاده کنید، برنامه نویسی شما با روش سابق "هیچ" تفاوتی نخواهد داشت. مسلماً این توانمندی تیم مایکروسافت را نمایش می‌دهد، نه توانمندی من و شما را! اما اگر نیاز به کمی ابتکار وجود داشت اینجا است که گیر خواهید افتاد و موارد 6 و 7 و 9 ذکر شده، jQuery را بر این کتابخانه مقدم خواهد ساخت)
- 11- با آمدن ASP.Net MVC تمایل به استفاده از jQuery به شدت افزایش یافته که نمود آن را در وبلاگ‌های مطرح در این زمینه می‌توان مشاهده کرد (اکثر مطالب ارسالی این روزهای بلاگ‌های ASP.Net MVC حول و حوش jQuery و ASP.Net است).

نظرات خوانندگان

نویسنده: Anonymous
تاریخ: ۱۳۸۸/۰۶/۱۱ ۲۱:۵۰:۰۲

قبلایک بحثی در سایت برنامه نویس در این باره شده بود من کاملا با حرفتون موافقم ajax .asp.net خیلی سنگینه

نویسنده: مهدی یوسفی
تاریخ: ۱۳۸۸/۰۶/۱۵ ۰۷:۱۰:۴۹

سلام

من مطلب جالبی از نویسنده کتاب Asp.net MVC خواندم که طبق تحقیقات انجام شده توسط ماکروسافت است:
اونها به عبارت Bloated Asp.net را دادند و این به این معنی است:

:The term bloated is often thrown at frameworks or technologies that either

People don't understand or aren't completely aware of and don't want to take the time to learn

Come from a provider people don't respect or like

People have heard bad things about

Are actually bloated

و این نظریه که اغلب مردم نسبت به Asp.net Ajax دارند.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۶/۱۵ ۰۸:۳۲:۴۸

Bloated رو در مورد هر چیزی بکار میبرند که زیاد شلوغش کرده باشه یا سنگین باشه و منابع زیادی را مصرف کند.

حالت forms authentication در ASP.Net، امکان تعریف کوکی‌های ماندگار را نیز جهت ورود خودکار کاربران در دفعات بعدی بازدید آن‌ها فراهم می‌کند. اما زمان منقضی شدن این کوکی‌های ماندگار در ASP.Net 1.1 و ASP.Net 2.0 به بعد کاملاً با هم متفاوت بوده و اگر برنامه نویس از این تغییر حاصل شده مطلع نباشد ممکن است بارها و بارها برنامه را آزمایش کند اما نتیجه‌ای نگیرد.

مدت زمان منقضی شدن کوکی‌های ماندگار ASP.Net 1.1 در forms authentication به صورت زیر است (بود):

زمان منقضی شدن کوکی سشن = زمان جاری + زمان منقضی شدن سشن کاربر

زمان منقضی شدن کوکی ماندگار (persistent) = زمان جاری + 50 سال

این عدد 50 سال بسیار غیرمنطقی بوده و در ASP.Net 2.0 به بعد به صورت زیر اصلاح شده است:

زمان منقضی شدن کوکی سشن = زمان جاری + زمان منقضی شدن سشن کاربر

زمان منقضی شدن کوکی ماندگار (persistent) = زمان جاری + زمان منقضی شدن سشن کاربر (پیش فرض آن 30 دقیقه است)

و جهت بیشتر کردن طول عمر این ماندگاری باید زمان مورد نظر را در وب کانفیگ به صورت زیر اعمال کرد:

```
<authentication mode="Forms">
<forms
    ...
    name="MyCookieName"
    slidingExpiration="true"
    timeout="43200"
  />
</authentication>
```

مطابق زمان تنظیم شده فوق که بر حسب دقیقه است، این کوکی به مدت یک ماه ماندگاری پیدا می‌کند در غیر اینصورت اعمال گزینه "مرا به خاطر بسپار" مطابق نظر شما کار نخواهد کرد و همانند ASP.Net 1.1 به صورت پیش فرض به 50 سال تنظیم نمی‌شود.

همچنین نکته‌ی مهم دیگری را که باید رعایت کرد، name ای است که در این فایل config عنوان می‌کنید (در قسمت تنظیمات form authentication). 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 منحصر بفرد انتخاب کنید، در غیر اینصورت تداخل رخ داده و باز هم گزینه "مرا به خاطر بسپار" شما کار نخواهد کرد.

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

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

مشکل: زمانیکه یک ASP.NET در آپدیت پنل AsyncPostback رخ دهد، پس از پایان کار، پلاگین جی‌کوئری که در حال استفاده از آن بودید و در هنگام بارگذاری اولیه صفحه بسیار خوب کار می‌کرد، اکنون از کار افتاده است و دیگر جواب نمی‌دهد.

- قبل از شروع، نیاز به یک سری پیش زمینه هست (شاید بر اساس روش استفاده شما از آن پلاگین جی‌کوئری، مشکل را حل کنند):
- رفع تداخل جی‌کوئری با سایر کتابخانه‌های مشابه.
 - آشنایی با `jQuery Live` مورد استفاده شما پس از هر AsyncPostback رخداده به عناصری که بعداً به صفحه اضافه خواهند شد.
 - تزریق اسکریپت به صفحه در حین یک AsyncPostback رخداده در آپدیت پنل

علت بروز مشکل:

علت رخدادن این مشکل (علاوه بر قسمت الف ذکر شده)، عدم فراخوانی `document.ready` تعریف شده، جهت بایند مجدد پلاگین `jQuery` مورد استفاده شما پس از هر AsyncPostback رخداده است. راه حل استاندارد جی‌کوئری هم همان مورد (ب) فوق می‌باشد، اما ممکن است جهت استفاده از آن نیاز به بازنویسی یک پلاگین موجود خاص وجود داشته باشد، که آنچنان مقرن به صرفه نیست.

مثالی جهت مشاهده‌ی این مشکل در عمل:

می‌خواهیم افزونه‌ی [Colorize - jQuery Table](#) را به یک گرید ویوو ASP.NET قرار گرفته درون یک آپدیت پنل اعمال کنیم.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="UpdatePanelTest.aspx.cs"
Inherits="TestjQueryAjax.UpdatePanelTest" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <asp:ScriptManager ID="sm" runat="server">
            <Scripts>
                <asp:ScriptReference Path("~/js/jquery.js" />
                <asp:ScriptReference Path "~/js/jquery.colorize-1.6.0.js" />
            </Scripts>
        </asp:ScriptManager>
        <asp:UpdatePanel ID="uppn1" runat="server">
            <ContentTemplate>
                <asp:GridView ID="GridView1" runat="server" AllowPaging="True"
                    OnPageIndexChanging="GridView1_PageIndexChanging">
                </asp:GridView>
            </ContentTemplate>
        </asp:UpdatePanel>
    </form>

    <script type="text/javascript">
        $(document).ready(function() {
            $('#<%=GridView1.ClientID %>').colorize();
        });
    </script>
</body>
</html>
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
```

```
using System.Web.UI.WebControls;
namespace TestJQueryAjax
{
    public partial class UpdatePanelTest : System.Web.UI.Page
    {
        void BindTo()
        {
            List<string> rows = new List<string>();
            for (int i = 0; i < 1000; i++)
                rows.Add(string.Format("row{0}", i));

            GridView1.DataSource = rows;
            GridView1.DataBind();
        }

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

        protected void GridView1_PageIndexChanging(object sender, GridViewEventArgs e)
        {
            GridView1PageIndex = e.NewPageIndex;
            BindTo();
        }
    }
}
```

مثل بسیار ساده‌ای است جهت اعمال این افرونه به یک گریدویو و مشاهده کار کردن این افزونه در هنگام بارگذاری اولیه صفحه و سپس از کار افتادن آن پس از مشاهده صفحه دوم گرید. در این مثال از نکته "[اسکریپت‌های خود را یکی کنید](#)" استفاده شده است.

راه حل:

از ویژگی‌های ذاتی ASP.Net Ajax باید کمک گرفت برای مثال:

```
<script type="text/javascript">
    $(document).ready(function() {
        $('#<%=GridView1.ClientID %>').colorize();
    });

    function pageLoad(sender, args) {
        if (args.get_isPartialLoad())
            $('#<%=GridView1.ClientID %>').colorize();
    }
</script>
```

متد استاندارد `pageLoad` به صورت خودکار پس از هر `AsyncPostback` رخداده در آپدیت پنل ASP.Net Ajax فراخوانی می‌شود (و همچنین پس از پایان پردازش و بارگذاری اولیه DOM صفحه). در این متد بررسی می‌کنیم که آیا یک `partial postback` رخداده است؟ اگر بله، مجدداً عملیات بایند افزونه به گردید را انجام می‌دهیم و مشکل برطرف خواهد شد.

[برای مطالعه بیشتر](#)

نظرات خوانندگان

نوبسند: نیما
تاریخ: ۱۳۸۸/۰۶/۲۱ ۱۰:۴۲:۲۸

سلام استاد عزیز
ممنون از مطلب بسیار مفیدتون.

برای استفاده از قابلیت‌های jQuery کتابخانه Ajax، شش متد زیر در اختیار برنامه نویس‌ها است:

```
$.ajax(), load(), $.get(), $.getJSON(), $.getScript(), and $.post()
```

که در حقیقت ۵ مورد آخر ذکر شده صرفا بیان اولین متد ajax فوق به نحوی دیگر می‌باشد و محصور کننده توانایی‌های آن هستند.

برای مثال کد زیر زمان جاری را از سرور دریافت کرده و نتیجه را در سه تکست باکس قرار داده شده در صفحه نمایش می‌دهد.
ابتدا وب سرویس ساده زیر را در نظر بگیرید که زمان شمسی جاری را بر می‌گرداند:

```
using System;
using System.Globalization;
using System.Web.Script.Services;
using System.Web.Services;

namespace TestjQueryAjax
{
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    [ScriptService]
    public class AjaxSrv : WebService
    {
        public class TimeInfo
        {
            public string Date { set; get; }
            public string Hr { set; get; }
            public string Min { set; get; }
        }

        [WebMethod(EnableSession = true)]
        [ScriptMethod(ResponseFormat = ResponseFormat.Json)]
        public TimeInfo GetTime()
        {
            DateTime now = new DateTime(DateTime.Now.Year, DateTime.Now.Month, DateTime.Now.Day);
            PersianCalendar pc = new PersianCalendar();
            string today = string.Format("{0}/{1}/{2}",
                pc.GetYear(now),
                pc.GetMonth(now).ToString("00"),
                pc.GetDayOfMonth(now).ToString("00"));

            TimeInfo ti = new TimeInfo
            {
                Date = today,
                Hr = DateTime.Now.Hour.ToString("00"),
                Min = DateTime.Now.Minute.ToString("00")
            };
            return ti;
        }
    }
}
```

سپس اگر از تابع Ajax کتابخانه jQuery جهت دریافت زمان جاری از وب سرویس استفاده نماییم، کد صفحه ما به صورت زیر خواهد بود:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="TestFillCtrls.aspx.cs"
Inherits="TestjQueryAjax.TestFillCtrls" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

غیرفعال کردن کش مرورگر هنگام استفاده از jQuery Ajax

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>

<script src="js/jquery.js" type="text/javascript"></script>

<script type="text/javascript">
    function validate() {
        $.ajax({
            type: "POST",
            url: 'AjaxSrv.asmx/GetTime',
            data: '{}',
            contentType: "application/json; charset=utf-8",
            dataType: "json",
            success:
                function(msg) {
                    $("#txtDate.ClientID").val(msg.d.Date);
                    $("#txtHr.ClientID").val(msg.d.Hr);
                    $("#txtMin.ClientID").val(msg.d.Min);
                },
            error:
                function(XMLHttpRequest, textStatus, errorThrown) {
                    alert("خطای رخداده است");
                }
        });
        //debugger;
    }
</script>

</head>
<body>
<form id="form1" runat="server">
<div>
    <asp:TextBox ID="txtDate" runat="server" />
    <br />
    <asp:TextBox ID="txtHr" runat="server" />
    <br />
    <asp:TextBox ID="txtMin" runat="server" />
    <br />
    <asp:Button ID="btnGetTime" runat="server" Text="Click here!" UseSubmitBehavior="false"
        OnClientClick="validate();return false;" />
</div>
</form>
</body>
</html>
```

تنها نکته‌ی جدید این اسکریپت، نحوه‌ی استفاده از خروجی JSON و ب متده است که از نوع TimeInfo تعریف شده است.
خروجی نمونه این وب متده به صورت زیر می‌تواند باشد:

```
{"d": {"__type": "TestJQueryAjax.AjaxSrv+TimeInfo", "Date": "1388/07/14", "Hr": "12", "Min": "59"}}
```

که نحوه‌ی دسترسی به اجزای آن را در متده validate ملاحظه می‌نمائید.

باید به خاطر داشت که برای هر 6 متده Ajax ایی jQuery، عملیات کش شدن اطلاعات در مرورگر کاربر به صورت پیش فرض فعال است. اما این نکته تنها زمانیکه dataType مورد استفاده از نوع jsonp یا script باشد، صادق نبوده و کش شدن به صورت خودکار غیرفعال می‌گردد.

روش سنتی غیرفعال کردن کش در حین عملیات اجکسی، استفاده از یک کوئری استرینگ متغیر در پایان url درخواستی است. به این صورت مرورگر درخواست صادره را جدید فرض کرده و از کش خود استفاده نمی‌نماید (همین مورد در حالت کش شدن تصاویر هم [صادق است](#)).

نیز همین عملیات را در پشت صحنه انجام داده اما تنظیم آن را به نحوی مطلوب‌تری ارائه می‌دهد. یا پارامتر cache را در تعريف متده ajax خود اضافه نموده و مقدار آن را مساوی false قرار دهید و یا جهت تاثیر گذاری بر روی کلیه متدهای مورد استفاده، پیش از استفاده از آن‌ها این تنظیم را مشخص سازید:

```
$.ajaxSetup({cache: false});
```

نظرات خوانندگان

نویسنده: شاهین کیاست
تاریخ: ۱۳۸۹/۱۱/۱۱ ۲۱:۵۶:۴۰

سلام.

علت `Return False` که آخر این دستور گذاشتید چیست؟

`;OnClientClick="validate();return false`

ممنون.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۱/۱۱ ۲۳:۲۰:۰۰

سلام، از `UseSubmitBehavior` هم استفاده شده. علت غیرفعال کردن `postback` است تا بشود نمایی از عملیات Ajax را مشاهده کرد.

در مورد کار با **j** و نحوه فراخوانی یک متاد وب سرویس توسط آن، چند مطلب پیشتر ارائه شدند:

[بررسی وجود نام کاربر با استفاده از j](#) در **jQuery Ajax**

و ...

تمام این مقالات یک ایراد مهم دارند که امروز با آن مواجه شدم و خلاصه آن به شرح زیر است:
پارامتر **data** متاد **Ajax** جی‌کوئری را به صورت زیر در نظر بگیرید:

```
data: {"username": '' + $('#<%= TextBox1.ClientID %>').val() + "'"},
```

این روش شاید با بسیاری از ورودی‌ها کار کند اما یک سری از کاراکترها زمانیکه با فرمت JSON ارسال می‌شوند باید اصطلاحاً **شوند** که لیست آن‌ها به صورت زیر است:

```
\b Backspace (ascii code 08)
\f Form feed (ascii code 0C)
\n New line
\r Carriage return
\t Tab
\v Vertical tab
' Apostrophe or single quote
" Double quote
\ Backslash character
```

برای حل این مشکل می‌توان از اسکریپت استاندارد [json2.js](http://json.org/json2.js) که در سایت json.org معرفی شده، استفاده کرد.

ابتدا ارجاعی از این اسکریپت باید به صفحه اضافه شود:

```
<script src="js/json2.js" type="text/javascript"></script>
```

سپس مقدار پارامتر **data** را باید به شکل زیر اصلاح کرد:

```
var jsonText = JSON.stringify({ username: $('#<%= TextBox1.ClientID %>').val() });
...
data: jsonText,
```

متاد **stringify** تمام ملاحظات لازم را در مورد **escape** کاراکترهای یاد شده و بسیاری از موارد دیگر به صورت خودکار اعمال می‌کند. همچنین از این لحاظ که یک ساختار داده‌ای جاوا اسکریپتی را نیز می‌تواند به معادل متنی آن تبدیل کند، بسیار جالب توجه است.

به عنوان مثالی دیگر، اگر متاد وب سرویس ما دو پارامتر داشت، **jsonText** به شکل زیر در خواهد آمد:

```
var jsonText = JSON.stringify({ param1: val1, param2: val2 });
```

لازم به ذکر است که پشتیبانی از این متاد در نگارش‌های آخر اکثر مرورگرهای امروزی در جهت سازگاری با آخرین استانداردهای مصوب جاوا اسکریپت به صورت بومی وجود دارد (از [IE8](#) به بعد و فایرفاکس 3 و یک به بعد)، اما از آنجاییکه کاربران ممکن است هنوز از نگارش‌های قدیمی آن‌ها استفاده کنند، کمک گرفتن از فایل [json2.js](#) ضروری به نظر می‌رسد.

نظرات خوانندگان

نویسنده: peyman naji
تاریخ: ۲۱:۳۰:۴۹ ۱۳۸۸/۰۸/۱۰

بسیار عالی بود . یه سوالی داشتم و اینکه برای کار با `jquery ajax` در `visualstudio` پروژه `Enabale Ajax` ایجاد نمود یا اینکه با پروژه معمولی هم کار میکنه چرا که هنگام پاس دادن اطلاعات بصورت `json` فایر باگ میگفت فرمت صحیح نیست و وقتی که با پروژه `Enable ajax` کار کردم مشکل حل شد در این مورد یه توضیحی میدید . با سپاس فراوان /

نویسنده: DotNetCoders
تاریخ: ۲۱:۳۵:۴۸ ۱۳۸۸/۰۸/۱۰

سلام !

خسته نباشید.

فکر کنم ۲ بار دیگه این مطلب رو گفته بودید.

/

(; happy new weblog name

نویسنده: وحید نصیری
تاریخ: ۲۱:۵۷:۳۳ ۱۳۸۸/۰۸/۱۰

peyman naji@
هنگامیکه پروژه Ajax enabled شود یک سری مدخل به فایل `web.config` در مورد تبادلات اجکسی اضافه میشود که حتما لازم است (در مورد فعل سازی `scripting` و `webServices` است). یک نمونه آن را در اینجا توضیح داده ام:
http://www.dotnettips.info/2009/07/blog-post_06.html

DotNetCoders@
- یکبار بود، البته نه به این صورت و واضحی.
- منون. اسم قبلی کمی زیاده روی بود (:)

نویسنده: علی یگانه مقدم
تاریخ: ۲۲:۳۱ ۱۳۹۴/۰۳/۱۰

من کد زیر رو ارسال میکنم ولی مشکلی که وجود داره این هست که نه پارامترها ارسال میشه و نه درخواست آزادکسی هست. صفحه رو واقعا هدایت میکنه به آدرس مورد نظر.

```
//new group
$("#NewGroup").on('click', function () {
    var catName = 'new';// $("#appendNodeText").val();
    $.ajax({
        dataType: "json",
        type: "POST",
        url: "@Url.Action(MVC.Categories.NewCategory())",
        data: { name: catName }

    }).done(function (data) {
        alert(data);
    });
});
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۳/۱۰ ۲۲:۵۷

را هم بررسی کنید. [event.preventDefault\(\)](#)

چه زمان‌هایی ممکن است ASP.NET یک صفحه `Page_Load` دوبار اجرا شود؟

عنوان:

نویسنده: وحید نصیری

تاریخ: ۱۳۸۸/۰۹/۱۴

آدرس: www.dotnettips.info

برچسب‌ها: ASP.NET

سؤال مربوط به حالت نخ نمای `Page.IsPostBack` نیست. مربوط به حالتی است که دقیقا در اولین بار مشاهدهی عادی یک صفحه، `Page_Load` دوبار یا بیشتر (!) اجرا می‌شود.

الف) برنامه‌ی `ASP.NET 1.x` خود را به نگارش‌های 2+ ارتقاء داده‌اید.

در این حالت هر چند `VS.NET` پیغام تبدیل با موفقیت یک پروژه‌ی قدیمی را به شما ارائه خواهد داد اما یک سری موارد را پس از تبدیل، باید اصلاح کرد.

پروژه‌های قدیمی `ASP.NET` در روای `InitializeComponent` خود سطر زیر را همانند یک پروژه `WinForm` و امثال آن برای معرفی روای رخداد گردان `Page_Load` دارند.

```
this.Load += new System.EventHandler(this.Page_Load);
```

اما در پروژه‌های `ASP.NET 2.0` به بعد دیگر از این روای خبری نیست. پس در این پروژه‌ها کامپایلر چگونه متوجه خواهد شد که `AutoEventWireup` واقعاً یک روای رخداد گردان است و نه یک روای معمولی؟ در پروژه‌های جدید `VS.NET`، خاصیت `Page_Load` صفحه به `true` تنظیم شده و دیگر نیازی به معرفی صریح روای‌های رخداد گردانی مانند `Page_Load` و یا `Page_Init` نبوده و تشخیص آن‌ها به صورت خودکار انجام می‌شود.

```
<% @Page AutoEventWireup="true" %>
```

بر این اساس با انتقال یک پروژه قدیمی به `VS.NET` جدید، هم `AutoEventWireup=true` را خواهیم داشت و هم سطرهای موجود در متد قدیمی `InitializeComponent` از پروژه حذف نشده‌اند. بنابراین متد `Page_Load` در دو نقطه، یکبار به صورت خودکار (متد `Page_Load` بر اساس نام آن با توجه به `AutoEventWireup=true` تشخیص داده شده و اجرا می‌شود) و یکبار هم بر اساس تعریف دستی موجود فراخوانی می‌شود.

برای حل این مشکل و سازگاری بهتر با نگارش‌های جدیدتر، سطر تعریف دستی روای رخداد گردان متد `Page_Load` را حذف کنید. لازم به ذکر است که این سیم کشی خودکار تنها برای متدهای زیر انجام خواهد شد و نسبت به حذف سایر موارد موجود اقدام نکنید!

```
Page_PreInit  
Page_Init  
Page_InitComplete  
Page_Preload  
Page_Load  
Page_LoadComplete  
Page_DataBind  
Page_SaveStateComplete  
Page_Preview  
Page_PreviewComplete  
Page_Unload  
Page_Error  
Page_AbortTransaction  
Page_CommitTransaction
```

ب) وجود هر نوع دکمه‌ی تصویری یا کلا تصویری با ویژگی `src` مقدار دهی نشده در صفحه مرورگر `IE` با این مساله مشکلی ندارد. اما فایرفاکس‌های جدید اگر به `src` مقدار دهی نشده‌ی تصویری برخورد کنند دقیقاً آدرس

جاری صفحه را بجای مقدار src قرار داده و مجدداً صفحه را درخواست می‌کنند (و البته این مورد ارتباط مستقیمی به ASP.NET یا PHP و امثال آن ندارد و یک مساله‌ی عمومی است). این مورد سبب خواهد شد که Page_Load صفحه، نه فقط دوبار بلکه به تعداد بار src خالی تصاویری که در صفحه وجود دارند، بر اساس درخواست‌های مجدد فایرفاکس از سرور اجرا شود. (مرورگر IE بجای فرآخوانی آدرس صفحه جاری، یک null/را به انتهای آدرس جاری اضافه کرده و آن را فرآخوانی می‌کند. بنابراین سبب اجرای مجدد هیچ روالی نخواهد شد).

ماخذ:

[Inside AutoEventWireup](#)
[How Firefox Handles Empty SRC tags](#)

نظرات خوانندگان

نویسنده: افشار محبی
تاریخ: ۱۳۸۸/۰۹/۱۸ ۱۰:۵۵:۴۵

در حالتی که از UpdatePanel استفاده می‌شود هم به نظر می‌رسد که Page_Load دو بار اجرا می‌شود. البته واقعیت این است که هر Postback هم خود یک Partial Postback مستقل محسوب شده و Page_Load را فرآخوانی می‌کند در نتیجه خیلی‌ها (به اشتباه) فکر می‌کنند که Postback دو بار (یا بیشتر) اجرا می‌شود...

قسمت‌های اول تا سوم این مقاله: [+ ۱](#) و [+ ۲](#) و [+ ۳](#)

در قسمت چهارم قصد داریم هدر مربوط به Content Expiration Date را توسط یک Http module به محتوای غیرپویای سایت مانند تصاویر، فایل‌های CSS و غیره اعمال کنیم. این روش از روش قسمت دوم ساده‌تر است و جامع‌تر. ابتدا یک پروژه‌ی Class library StaticContentCacheModule جدید را به نام StaticContentCacheModule ایجاد کرده و سپس ارجاعی را به اسمبلی استاندارد System.Web.dll به آن خواهیم افزود. سپس کدهای مرتبط با این مأذول به شرح زیر هستند:

```
//StaticCache.cs
using System;
using System.Web;

namespace StaticContentCacheModule
{
    public class StaticCache : IHttpModule
    {
        public void Init(HttpApplication context)
        {
            context.PreSendRequestHeaders += context_PreSendRequestHeaders;
        }

        static void context_PreSendRequestHeaders(object sender, EventArgs e)
        {
            //capture the current Response
            var currentResponse = ((HttpApplication)sender).Response;

            if (CacheManager.ShouldCache(currentResponse.ContentType))
            {
                currentResponse.AddHeader("cache-control", "public");
                currentResponse.AddHeader("Expires",
DateTime.Now.Add(TimeSpan.FromDays(30)).ToString());
            }
        }

        public void Dispose() { }
    }
}
```

در اینجا ContentType تک تک عناصری که توسط وب سرور ارائه خواهند شد، بررسی می‌شود. اگر نیازی به کش شدن آن‌ها وجود داشت (توسط کلاس CacheManager این امر مشخص می‌گردد)، هدر مربوطه اضافه می‌گردد.

```
//CacheManager.cs
using System;

namespace StaticContentCacheModule
{
    class CacheManager
    {
        public static bool ShouldCache(string contentType)
        {
            contentType = contentType.ToLower();
            string[] parts =
                contentType.Split(
                    new[] { ';' },
                    StringSplitOptions.RemoveEmptyEntries
                );

            if (parts.Length > 0)
                contentType = parts[0];

            bool cache = false;
```

```
switch (contentType)
{
    case "text/css":
        cache = true; break;
    case "text/javascript":
    case "text/jsonscript":
        cache = true; break;
    case "image/jpeg":
        cache = true; break;
    case "image/gif":
        cache = true; break;
    case "application/octet-stream":
        cache = true; break;
    default:
    {
        if (contentType.Contains("javascript"))
            cache = true;
        if (contentType.Contains("css"))
            cache = true;
        if (contentType.Contains("image"))
            cache = true;
        if (contentType.Contains("application"))
            cache = true;
    }
    break;
}
return cache;
}
}
```

در این کلاس، دریافتی بررسی می‌شود. اگر نوع محتوای قابل ارائه از نوع CSS، JavaScript، تصویر و یا Application بود، یک مقدار true بازگشت داده خواهد شد.
نهایتاً برای استفاده از این Http module جدید در IIS6 به قبل در وب کانفیگ برنامه خواهیم داشت:

```
<httpModules>
    <add name="StaticContentCacheModule" type="StaticContentCacheModule.StaticCache,
StaticContentCacheModule"/>
</httpModules>
```

و یا در IIS7 این تغییرات به صورت زیر می‌تواند باشد:

```
<system.webServer>
<modules>
    <add name="StaticContentCacheModule" type="StaticContentCacheModule.StaticCache,
StaticContentCacheModule"/>
</modules>
```

اکنون اگر یک پروژه‌ی آزمایشی جدید ASP.Net را گشوده و فایل css ساده‌ای را به آن اضافه کنیم، بررسی هدر نهایی توسط افزونه‌ی YSlow به صورت زیر خواهد بود:

URL	EXPIRES (Y/M/D)
http://localhost/TestStaticContentCacheModule/	no expires
http://localhost/TestStaticContentCacheModule/Stylesheet1.css	2010/2/10

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۰/۲۰ ۰۹:۴۴:۱۴

نکته: اگر کلاس‌های فوق را به برنامه‌ی خود اضافه کرده‌اید و مازول‌ها کار نمی‌کنند باید به این مطلب دقت داشته باشید: [\(+\)](#)

نویسنده: mohammad
تاریخ: ۱۳۸۹/۱۰/۲۰ ۱۹:۲۵:۱۰

سلام آقا بازم نمی‌شه من کد رو به vb تبدیل کردم و در وب کافیگ هم به صورت زیر گذاشتم :

```
add name="StaticContentCacheModule" type="StaticContentCacheModule.StaticCache, App_Code.7pqudgp-,  
"Version=0.0.0.0, Culture=neutral, PublicKeyToken=null
```

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۰/۲۰ ۲۲:۱۳:۳۳

سورس کامل مرتبط با این مطلب در همان زمان: [\(+\)](#)

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۰/۲۰ ۲۳:۴۳:۲۳

اگر روش فوق برای شما کار نمی‌کند، روش ذکر شده در این مطلب: [\(+\)](#) حتما کار خواهد کرد و حداقل برای اسکریپت‌های سایت می‌توانید از آن استفاده کنید.

نویسنده: مهدی پایروند
تاریخ: ۱۳۸۹/۱۰/۲۱ ۱۰:۴۰:۲۵

بنظرم، مشکل با عبارت This operation requires IIS integrated pipeline mode که هم در IIS6 و IIS7 وجود دارد

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۰/۲۱ ۱۱:۳۲:۳۱

فقط IIS های 7 به بعد integrated pipeline mode را دارند. به همین جهت بجای = currentResponse.Headers[something] استفاده شده. currentResponse.AddHeader از something

نویسنده: مهدی پایروند
تاریخ: ۱۳۸۹/۱۰/۲۲ ۱۳:۳۷:۳۴

من مشکل رو در iis7 در ویندوز 7 می‌بینم و نکته دیگه اینکه در iis6 و ویندوز سرور 2003 هم کار نکرده و با اینکه این چند روز خیلی دنبال این مسئله تو اینترنت بودم حدود 70 مقاله که اکثراً شبیه به هم بودند رو بررسی کردم حتی iis7 رو دوباره نصب کردم ولی مشکل همچنان باقیست.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۰/۲۲ ۱۳:۵۵:۳۰

اگر دسترسی به IIS دارید که نیازی به این مقاله نیست. باید به اینجا مراجعه کنید: [\(+\)](#)
در مورد IIS7 هم اطلاعات بیشتر اینجا است: [\(+\)](#)

در IIS7 این مورد توسط انتخاب application pool مناسب تنظیم می‌شود. برای مثال آن با Classic mode سازگار است.

پاسخ : هیچکدام!

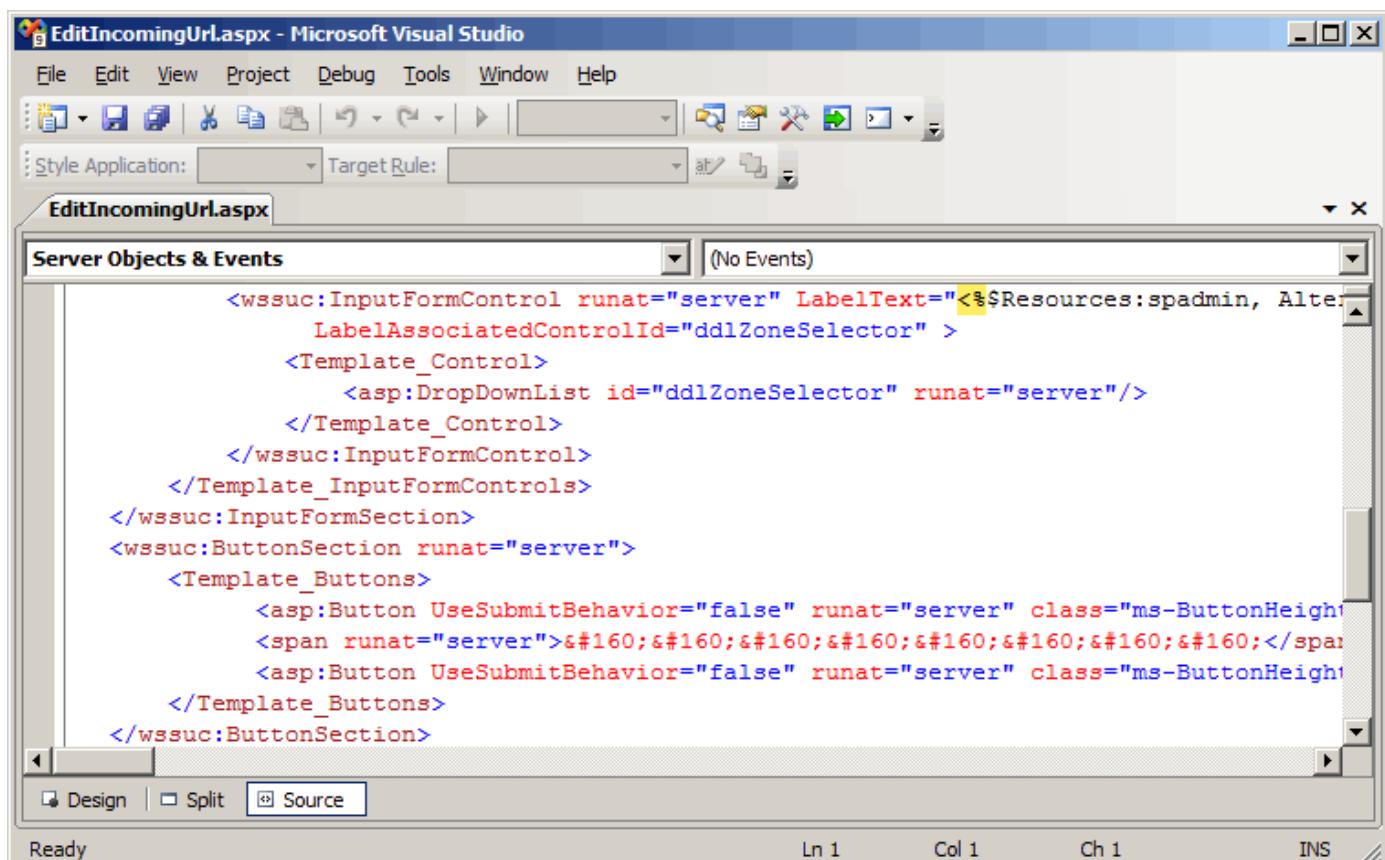
برای نمونه دو مورد از محصولات مهم تجاری و پر درآمد مایکروسافت در مقیاس سازمانی Exchange server و SharePoint هستند (البته اینجا منظور برنامه web access مربوط به Exchange server است). جالب اینجا است که هر دو محصول، مبتنی بر دات نت فریم ورک سه و نیم بوده و از ASP.Net WebForms استفاده می‌کنند. تفاوت مهم آن‌ها با نگارش سال 2007 هر کدام، استفاده از ASP.Net Ajax مایکروسافت در این محصولات است و همچنین استفاده‌ی وسیع از توانمندی‌های پاورشل 2 خصوصاً امکان مدیریت از راه دور پاور شل 2 که برای مثال در برنامه web access مربوط به exchange server 2010 ، امكان مدیریت خود است اما منسوخ شده در نظر گرفته می‌شود).

به علاوه هر دو محصول فقط با ویندوز‌های سرور 2008 به بعد، آن هم نسخه‌ی 64 بیتی کار می‌کنند. (البته از آنجائیکه هسته‌ی ویندوز 7 با هسته‌ی ویندوز سرور 2008 نگارش R2 یکی است (یا حداقل بر مبنای یک code base هستند)، SharePoint 2010 را بر روی ویندوز 7 شصت و چهار بیتی هم می‌توان جهت آزمایش و توسعه نصب کرد) یک دوره‌ی مدیریتی SharePoint 2010 را می‌توانید در آدرس زیر مشاهده نمائید:

Microsoft SharePoint 2010 Administration

جهت اثبات این مدعای (استفاده از WebForms و نه MVC) دو تصویر ذیل به اندازه‌ی کافی گویا هستند:

شیرپوینت 2010



Exchange server 2010 در Web Access

The screenshot shows the Microsoft Visual Studio interface with the title bar "EditAdminRoleGroup.aspx - Microsoft Visual Studio". The menu bar includes File, Edit, View, Project, Debug, Tools, Window, and Help. Below the menu is a toolbar with various icons. The main window displays the source code for "EditAdminRoleGroup.aspx". The code is written in C# and uses ASP.NET controls like `<asp:Content>`, `<ecp:properties>`, and `<div>`. The code is color-coded for syntax highlighting. At the bottom of the code editor, there are tabs for Design, Split, and Source, with Source selected. The status bar at the bottom shows "Ready", "Ln 1", "Col 1", "Ch 1", and "INS".

```
<%@ Page Language="C#" AutoEventWireup="false" Inherits="Microsoft.Exchange.Management.MasterPageFile=""~/CommonMaster.Master"" Title="<%$ Strings>EditRoleGroup %>" HelpId="EditRoleGroup" LocStringsResource="EditRoleGroup" EnableViewState="false" %>

<asp:Content ID="MainContent" ContentPlaceHolderID="ResultPanePlaceHolder" runat="server">
    <ecp:properties id="AdminRoleGroupProperties" runat="server" ServiceUrl="~/UsersGr...
        <sections>
            <ecp:Section ID="GroupInformationSection" runat="server" Title="<%$ Strings...
                <Content>
                    <div class="PropertyDiv">
                        <div>
                            <asp:Label runat="server" ID="displayName_label" Text="<%$ ...
                            <asp:Label runat="server" ID="displayNameContent_label" Da...
                        </div>
                    </div>
    
```

نظرات خوانندگان

نوبسند: ramin
تاریخ: ۱۴:۱۶:۲۴ ۱۳۸۹/۰۲/۰۸

ظاهرن microsoft نمی تونه از webform دست بکشه و نقش فرزند ناخوانده رو براش داره!

نوبسند: Sadegh
تاریخ: ۲۰:۱۳:۵۶ ۱۳۸۹/۰۲/۰۹

ناخوانده یا ناخواسته؟

بعد از مدت‌ها بالاخره یک باگ امنیتی در ASP.NET کشف شده! البته این "بعد از مدت‌هایی" که عنوان شد بر اساس آمار است که در سایت بی‌طرف Secunia قابل بررسی و مشاهده است (و اگر از یک سری کامپوننت و برنامه‌های جانبی سایر برنامه نویس‌ها صرفنظر کنیم، انصافاً ضریب امنیتی بالایی را ارائه داده):
<http://secunia.com/advisories/search/?search=asp.net>

البته این مورد جدید هم مرتبط با خود ASP.NET نیست؛ بلکه مرتبط است با نحوه پیاده سازی الگوریتم AES در دات نت فریم ورک (Padding Oracle / AES cookie encryption vulnerability). توضیحات بیشتر در سایت Scottgu [Important: ASP.NET Security Vulnerability](#)

خلاصه این توضیحات:
همانطور که هزاران بار به برنامه نویسان ASP.NET (و SharePoint) گوشزد شده است، لطفاً صفحه‌ی نمایش خطاهای سفارشی سایت را (customErrors) در web.config برنامه on کنید. این خطای امنیتی موجود در پیاده سازی الگوریتم AES نیاز به سعی و خطاهای زیاد روی سایت و بررسی پاسخ‌های داده شده یا ریزخطاهای منتشر شده دارد. اگر صفحه‌ی خطای سفارشی سایت شما تنظیم شده و روشن است، خطری سایت شما را تهدید نمی‌کند؛ زیرا شخص مهاجم هیچ خروجی خطایی را جهت بررسی مشاهده نخواهد کرد.

بله! همانطور که حدس زده می شد بالاخره مایکروسافت تکلیف خودش را با کتابخانه های Ajax ای تولید شده در طی این چند سال مشخص کرد و از این پس انتخاب اصلی جهت تولید برنامه های ASP.NET مبتنی بر Ajax ، تنها jQuery است.
اصل مطلب رو می تونید اینجا مطالعه کنید:

[Javascript Libraries and ASP.NET: A Guide to jQuery, AJAX and Microsoft](#)

[The Murky Future of ASP.NET AJAX](#)

خلاصه آن:

ASP.NET AJAX در آینده نیز کاملا پشتیبانی می شود، اما شهروند درجه یک محسوب نخواهد شد؛ مانند استفاده از ScriptManager و UpdatePanel و Ajax Control Toolkit که اکنون به صورت سورس باز در سایت CodePlex نیز قابل دریافت است، منسوخ شده در نظر گرفته شده و تنها در آینده هر از چندگاهی به رفع باغهای گزارش شده در آن پرداخته شود (اما دیگر به صورت فعال توسعه داده نخواهد شد).
Microsoft Ajax Library / ASP.NET Ajax Library هر چند تشابه اسمی با ASP.NET Ajax دارند اما جزئی از کتابخانه های AJAX بوده و این مورد نیز از این پس منسوخ شده در نظر گرفته خواهد شد (مانند استفاده از DataView یا Control Toolkit .(Sys.require

نظرات خوانندگان

نویسنده: farbod
تاریخ: ۱۳۸۹/۰۷/۲۶

سلام آقای نصیری. من، هم با این کتابخانه و updatepanel کار کردم و هم jquery . درسته که خروجی کار با jquery بسیار با کیفیت تر و انعطاف پذیری هم بیشتره ولی حقیقتش هر موقع از این روش استفاده می کنم احساس خوبی ندارم! به نظرم کدها بسیار نا مرتب و به قول معروف اسپاگتی هستن. من تمام تلاشم رو میکنم که در داخل فایل جاوااسکریپت بسیار با قاعده و طبقه بنده شده بنویسم، ولی باز هم .. به خصوص وقتی لازم میشه html رو داخل جاوااسکریپت و به صورت رشته ای تولید کنم. می خواستم بدونم واقعاً به همین صورته یا من از روش های نادرست استفاده میکنم.

نویسنده: محمد امین شریفی
تاریخ: ۱۳۸۹/۰۷/۲۶

سلام.

درست مثل linq to sql، که اولش تبلیغ می کرد از ADO.NET بیاین سمت Linq to SQL و بعدش گفت از Linq to SQL بیاین به سمت ADO.net Entity هم همینطور محجور مانده، ASP کلاسیک MFC

بدی محصولات ماکروسافت این هست که توسعه third party opensource همیشه پشتیبانی و به مرور بهنگام بشه و خوبیش هم این هست که یک فناوری خوش دست و راحتی میده.

مهندس نصیری، نظرت درباره جاوا چی هست؟، پستی را برای مقایسه این دو می نویسید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۷/۲۷

farbod@

برای کلا جاوا اسکریپت هم کتابخانه های Unit test وجود دارد تا این دلجرکین بودن خارج شوید:
[QUnit](#)

@محمد امین شریفی

اینبار انتخاب مایکروسافت درست بوده:

- 1- هزینه توسعه رو کاهش دادند. برای مثال هدف از Ajax Control Toolkit چیزی بوده شبیه به پلاگین های jQuery اما با هزینه بیشتر. اینبار شما هزاران پلاگین جی کوئری در اختیارتون هست و نه صرفا مجموعه ای Ajax Control Toolkit.
- 2- موردی رو که در مقاله بالا بهش اشاره شده و دلیل اصلی کنار گذاشتن ASP.NET Ajax بوده پایین بودن کارآبی است. هم حجم بالایی دارد و هم تبادلات شبکه ای حجمی را به همراه دارد که در هر دو مورد جی کوئری بهینه تر است.
- 3- انتخاب اصلی کتابخانه Ajax در الان همین جی کوئری است و در مقاله فوق به یک سری افزونه و کتابخانه کمکی برای آن اشاره شده (که در حال توسعه و تکمیل است).

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۷/۲۷

ضمنا یک مورد رو در باره‌ی ASP و LINQ to SQL entity framework L2S بجاشی اضافه کنم. جایگزین شدن entity framework به ASP.NET کلاسیک، یک روند سالم و سلامت توسعه است. LINQ to SQL فقط محدود است به SQL Server اما الان اکثر بانک‌های اطلاعاتی موجود پروايدر EF دارند و مدل توسعه‌ی آن بسته نیست. ASP کلاسیک رو نمی‌دونم باهاش کار کرده بودید یا نه؟ رسما

یک فاجعه بود! مخلوطی از کدهای برنامه داخل کدهای HTML و وابستگی آن به اشیاء COM و غیره (اگر می‌خواستید مثلاً رمزگاری را به آن اضافه کنید باید Active-X می‌نوشتید و در سرور登録 شود). این مورد اصلاً قابل قیاس نیست با ASP.NET و امکانات دات نت فریم ورک.

نوبسنده: shahin
تاریخ: ۱۳۸۹/۰۷/۲۷ ۱۷:۵۹:۳۷

سلام جناب نصیری یکی از مشکلاتی که برنامه نویسان تازه کار را اذیت می‌کند اجرای Jquery از سمت سرور هست.
برای حل این ضعف چه توصیه‌ای دارید
مثلاً Ajax پارامترها را سمت سرور سنت می‌کردیم و استفاده می‌کردیم.
اما Jquery باید سمت HTML همه کار را انجام داد.

نوبسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۷/۲۷ ۱۸:۳۴:۵۱

من در مورد jQuery در این سایت مطلب زیاد منتشر کردم. به عمد هم تمامشون با دید ASP.NET بوده.
به تگ‌های JavaScript و یا ASP.NET در کنار صفحه مراجعه کنید یا کلا فایل CHM خلاصه سایت رو جهت سهولت مطالعه دریافت کنید.
و بله. مايكروسافت اومده تمام اينها رو برای شما در ASP.NET Ajax کپسوله و از دید شما مخفی کرده. همين کپسوله سازی در پلاگین‌های jQuery هم با کیفیت بالا موجود است مثلاً : [\(+\)](#)

نوبسنده: Mehran Zand
تاریخ: ۱۳۸۹/۰۷/۲۷ ۲۱:۴۲:۱۵

با سلام،
تمام این موارد که مبفرا مایید درست، من خودم از طرفداران استقاده از jQuery هستم اما یک مسئله برای من هنوز وجود دارد اون هم RAD است. در یکی از کتابخانه‌های آژاکسی برای ASP.NET web forms مثل Anthem.Net asp.net می‌کنید که فقط منحصر است به آن هم نگارش‌های قبل از سه آن + پشتیبانی از این مسئله خیلی خوب رعایت شده و کار توسعه نرم افزار مبتنی بر استقاده از آژاکس به سرعت انجام می‌شود. این برای من هنوز سواله با jQuery چطور می‌توان به این سرعت نرم افزارهای Full-Ajax را توسعه داد.

نوبسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۷/۲۷ ۲۲:۴۷:۵۶

مايكروسافت اومده تمام اينها رو برای شما درست کرد، در مقابل jQuery بازنده اعلام کرده الان شما صحبت از Anthem.Net می‌کنید که فقط منحصر است به آن هم نگارش‌های قبل از سه آن + پشتیبانی از مرورگرهای مختلف هم در آن ضعیف است.
هدف مايكروسافت از اینکار مدیریت هر دو پروژه MVC و Web forms است آن هم با هزینه کم، کیفیت بالا و سازگار با تمام مرورگرهای کاره‌اند. فقط .
ضمنا این رو در نظر باشید که یکی از توانمندی‌های jQuery، Ajax است (از کار با DOM گرفته تا دستکاری CSS نمایش داده شده، تا Animation، مدیریت ساده‌تر رخدادها، اعمال قالب به سایت و غیره) و این مورد مزیت مهمی است نسبت به تمام کتابخانه‌هایی که فقط برای یک کار و آن هم سهولت تولید برنامه‌های مبتنی بر Ajax ایجاد شده‌اند و از چند مشکل مهم رنج می‌برند:
- تک کاره‌اند. فقط .
- مشکل سازگاری با مرورگرهای مختلف را دارند.
- به صورت فعل توسعه داده نمی‌شوند؛ رفع باگ نمی‌شوند و غیره. برای مثال فوائل به روز رسانی همان Anthem.Net را بررسی کنید.
- توسعه پذیر نیستند. برای مثال آیا می‌توان برای Anthem.Net افزونه نوشت؟
- حجم بالایی دارند.
- سرعت پایینی دارند.

jQuery در مورد تمام موارد عنوان شده حرف برای گفتن دارد از حجم کم تا سرعت بالاتر نسبت به اکثر کتابخانه‌های جاوا اسکریپتی دیگر تا توسعه‌ی منظم، سازگاری عالی با مرورگرها، توسعه پذیری و صدها و هزاران افزونه‌ی مهیا برای آن و غیره.

و RAD هزینه بر است. یعنی چی؟ یعنی حجم بالای کدهای اسکریپتی که باید به برنامه‌ی شما مثلاً توسط ASP.NET Ajax تزریق شود که مباداً شما بخواهید دست خودتان را به نوشتن چند سطر کد جاوا اسکریپت آلوه کنید. همچنین حجم تبادل اطلاعات ASP.NET Ajax را هم که مبتنی است بر RAD را هم با حجم اطلاعات مبادله شده توسط jQuery مقایسه کنید (با پلاگین فایرباگ مربوط به فایرفاکس). این حجم واقعاً زیاد است و قابل مقایسه نیست. (تمام این‌ها هزینه‌های RAD است) ضمناً وجود 100 ها افزونه و پلاگین نوشته شده برای jQuery کار شما را بسیار بسیار ساده می‌کنند، مانند پاسخ قبلی من در این مطلب.

فقط باید کمی وقت بگذرانید و چیزی را بیاموزید که واقعاً ارزش دارد. گیرم فردا نخواستید با ASP.NET کار کنید. تمام این اطلاعات در PHP هم به درد شما می‌خورد، چون قسمت سمت کلاینت آنچنان تفاوتی نمی‌کند و jQuery یک کتابخانه‌ی سمت کلاینت است.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۲/۱۹ ۰۱:۳۷:۵۳

در ادامه ... (+)

عنوان: انجمن سایت LLBLGEN سورس باز شد
نوبتندہ: وحید نصیری
تاریخ: ۱۳۸۹/۰۹/۱۰ ۰۸:۴۰:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net

LLBLGEN

یکی از ORM های تجاری بسیار با کیفیت دات نت است و علاوه بر اینکه هویت مستقلی دارد، امکان تولید کدهای مخصوص Entity framework و NHibernate را هم دارد است.
اخیرا این شرکت تصمیم گرفته است که سیستم پشتیبانی مشتریان خودش را که نمونه‌ای از آن را در [اینجا](#) می‌توانید ملاحظه کنید،
سورس باز کند.
حداقل پیشنبازهای نصب این انجمن به شرح زیر است:

- ASP.NET 2.0+
- SQL Server 2000 or higher / CE Desktop 3.5
- .NET 3.5+

لطفاً جهت دریافت آن به [این آدرس](#) و جهت ملاحظه قابلیت‌های آن به [این آدرس](#) مراجعه نمائید.

نظرات خوانندگان

نوبتند: سامان نام نیک
تاریخ: ۱۳۸۹/۰۹/۱۱ ۱۲:۰۲:۲۶

با سلام

آقای نصیری به نظر شما آپا LLBLGEN مناسب تر و بهتر از orm های دیگر مثل NHibernate و Entity framework هستش ممنون می شم یک مقایسه کلی کنید

نوبتند: وحید نصیری
تاریخ: ۱۳۸۹/۰۹/۱۱ ۱۴:۱۷:۰۶

سلام،

تقریباً بله!

- طول عمر آن از EF بیشتر است.

- رابط گرافیکی تولید نگاشتهای آن از EF و نمونه‌ی موجود در VS.NET بسیار بهتر و پایدارتر است.

- پشتیبانی از انواع بانک‌های اطلاعاتی دارد.

- code generator دارد.

- پشتیبانی کاملی از LINQ دارد.

و ...

ولی باز هم من ترجیح می‌دم سراغ راه حل‌های تجاری غیرمایکروسافتی نروم. حداقل می‌دونم مایکروسافت تا 10 سال دیگر هست. چندتا کتاب در مورد هر کدام از فریم ورک‌هاش موجود است، انواع و اقسام بلاگ‌ها، فوروم‌ها و غیره برای آن هست که قابل چشم پوشی نیست.

برای مطالعه بیشتر :

[\(+\)](#) , [\(+\)](#)

هر متغیر استاتیک تنها دارای یک مقدار، در یک AppDomain مشخص است (مگر اینکه با ویژگی `ThreadStatic` مزین شود). هر برنامه‌ی ASP.NET هم `AppDomain` جداگانه و منحصر به خود را دارد. بنابراین تعریف یک متغیر استاتیک در یک برنامه‌ی ASP.NET به معنای به اشتراک گذاری آن در بین تمامی درخواست‌های رسیده به سرور است. بنابراین عموماً استفاده از متغیرهای استاتیک در برنامه‌های چند کاربره ASP.NET یک اشتباه بزرگ است و در صورت استفاده از آن باید منتظر تخریب اطلاعات یا دریافت نتایج غیرمنتظره‌ای باشید (مگر اینکه واقعاً می‌دانید دارید چکار می‌کنید، برای مثال کش کردن نگاشته‌های NHibernate به این صورت و استفاده از الگوی `singleton` یا روش‌های مشابه که باید بین تمام کاربران به یک صورت و یک شکل به اشتراک گذاشته شود و در حین اجرای برنامه تغییری در آن حاصل نمی‌شود). برای مثال اگر کاربر یک، در صفحه‌ی یک، متغیر استاتیکی را مقدار دهی کند، کاربر 2 نیز با مقدار به روز شده‌ی کاربر یک کار خواهد کرد که به طور قطع این مورد نظر شما نیست (چون به احتمال زیاد طراحی شما بر اساس کار کاربر در یک `Session` است و نه یک مقدار برای تمام سشن‌های موجود در سایت) و همچنین باید دقت داشت که امنیت سیستم نیز در این حالت زیر سؤال است (زیرا در این حالت تمامی کاربران، صرفنظر از سطوح دسترسی تعریف شده برای آن‌ها، دسترسی به اطلاعاتی خواهند داشت که نباید داشته باشند).

نکته‌ی دیگری را هم که باید در مورد ASP.NET به خاطر داشت این است که ویژگی `ThreadStatic` نیز در اینجا کمکی نمی‌کند؛ زیرا مطابق طراحی آن از تردها استفاده‌ی مجدد می‌گردد. به عبارت دیگر در ASP.NET الزامی ندارد که آغاز یک درخواست جدید حتماً به همراه ایجاد یک ترد جدید باشد.

طول عمر این نوع متغیرها هم تا زمانی است که وب سرور یا برنامه‌ی استارت شوند. فقط در این حالت است که نمونه‌ی موجود تخریب شده و سپس با اجرای مجدد برنامه، بازسازی خواهد شد.

بنابراین متغیرهای استاتیک در ASP.NET همانند شیء `Application` عمل می‌کنند و از آن سریع‌تر هستند زیرا زمانیکه به آن‌ها ارجاع می‌شود نیازی به جستجو در یک جدول و یافتن آن‌ها نیست (برخلاف شیء `Application`) و همچنین در اینجا نیازی هم به عملیات تبدیل نوع داده‌ای وجود ندارد (برخلاف نوع شیء `Object` که به صورت `Object` تعریف شده است). وجود اشیاء `Application` در ASP.NET فقط به جهت حفظ سازگاری آن با ASP کلاسیک است و [توصیه شده است](#) در ASP.NET به دلایلی که ذکر شد، اگر و تنها اگر نیاز به اشیایی در سطح برنامه داشتید از متغیرهای استاتیک استفاده کنید. شیء `Cache` نیز در ASP.NET همین کاربرد را دارد با این تفاوت که می‌توان برای آن مدت زمان منقضی شدن تعريف کرد یا اینکه وب سرور بسته به حق تقدم و اهمیتی که برای آن تعريف شده است، مجاز به حذف کردن آن در زمانی است که با کمبود منابع مواجه می‌شود. همچنین باید دقت داشت که تنها مکان ذخیره سازی متغیرهای استاتیک حافظه است اما امکان ذخیره سازی کش بر روی فایل سیستم تا بانک اطلاعاتی و غیره نیز مهیا است.

سؤال: آیا تعریف `SqlConnection` به صورت استاتیک جزو مواردی است که "مگر واقعاً می‌دانید دارید چکار می‌کنید؟"؟ پاسخ: خیر. در اینجا هم واقعاً این شخص نمی‌داند که دارد چکار می‌کند! یعنی در مورد سازوکار درونی ADO.NET اطلاعاتی ندارد. باز کردن یک کانکشن در ADO.NET به معنای مراجعه به استخراج (pool) کانکشن‌ها و بازکردن یکی از آن‌ها و در مقابل، بستن یک کانکشن هم به معنای علامتگذاری یک کانکشن به صورت غیرفعال است و آماده سازی آن برای استفاده در درخواست بعدی. به معنای دیگر این عملیات سربار آنچنانی ندارد که بخواهید آن را استاتیک تعريف کنید.

همچنین مورد دیگری را هم که این برنامه نویس نمی‌داند این است که متغیرهای استاتیک `thread safe` نیستند. به عبارتی حين استفاده از آن‌ها در یک برنامه‌ی چند کاربره ASP.NET حتماً باید مکانیزم‌های قفل‌گذاری بر روی این نوع متغیرها و اشیاء اعمال شود (که این هم خود یک سربار اضافی است در مقایسه چند 10 یا چند 100 کاربر همزمان). این مشکلات همزنمانی به چه معنا است؟ فرض کنید کاربر یک، شیء استاتیک `SqlConnection` ای را باز کرده است و با آن مشغول کوئری گرفتن است. کاربر 2 نیز همزمان شروع به استفاده از این کانکشن باز در حال استفاده می‌کند (`SqlConnection` استاتیک یعنی استفاده‌ی تمام کاربران فقط و فقط از یک کانکشن باز شده)، نتیجه این خواهد بود که برای مثال پیغام خطایی را دریافت می‌کند مانند: `Fileld مورد نظر در جدول موجود نیست!` چرا؟ چون روی شیء استاتیک `SqlConnection` تعریف شده قفل گذاری صورت نگرفته است و در حین استفاده از آن هر کاربری در سایت نیز همان را استفاده خواهد کرد یا از آن بدتر ممکن است یک کاربر زودتر از کاربر دیگری آن را بیندد!

کاربر سوم در وسط کار با پیغام غیرمعتبر بودن کانکشن مواجه می‌شود، یا اینکه به صورت پیش فرض یک datareader را بیشتر نمی‌توان بر روی یک کانکشن باز شده اعمال کرد. کاربر 4 مشغول خواندن اطلاعات است، کاربر 5، پیغام غیرمعتبر بودن کوئری را دریافت می‌کند.

نظرات خوانندگان

نوبسند: Farhad
تاریخ: ۱۳۸۹/۰۹/۱۸ ۲۱:۰۶:۱۷

مطلوب بسیار جالبی بود.

نوبسند: Mehran Zand
تاریخ: ۱۳۸۹/۰۹/۱۸ ۲۲:۱۸:۵۷

در لایه بیزینس تعریف مند ها استاتیک مشکلی ایجاد میکند؟

نوبسند: وحید نصیری
تاریخ: ۱۳۸۹/۰۹/۱۸ ۲۳:۲۹:۰۰

لطفاً جهت تکمیل بحث به این مطلب مراجعه کنید: [\(+\)](#)

نوبسند: A
تاریخ: ۱۳۸۹/۰۹/۱۹ ۰۰:۵۲:۵۴

خیلی ممنون.

من برای معماری دسترسی به داده‌ام از روشی استفاده کرده بودم که لازم داشتم متن داده‌ام برای هر ریسمان یکتا باشد.
این کار را با استفاده از متدهای

Thread.AllocateDataSlot
Thread.GetData
Thread.SetData
انجام داده بودم.

اکنون استفاده از یک متغیر Static دارای ویژگی ThreadStatic را تست کردم و به خوبی جواب گرفتم.

فقط یک نکته کوچک وجود دارد. من یک متغیر bool نیز دارم که چون ValueType است در اولین بار که مورد دسترسی قرار می‌گیرد مقدار Default که همان false است را خواهد داشت. درحالی که باید از دسترسی برای اولین بار به این متغیر آگاهی پیدا کنم. البته یک روش ساده که فعلاً به ذهنم رسید استفاده از bool است.

باز هم ممنون.

نوبسند: وحید نصیری
تاریخ: ۱۳۸۹/۰۹/۱۹ ۰۱:۳۳:۰۲

ببینید محور اصلی بحث این تاپیک ASP.NET است و اگر از ویژگی ThreadStatic استفاده کرده‌اید، اشتباه است چون بحث استفاده مجدد از یک ترد موجود در ThreadPool در اینجا مطرح است (در بالا ذکر کردم).
در ASP.NET اگر می‌خواهید اطلاعاتی را صرفاً برای استفاده در طول عمر یک درخواست ذخیره کرده و به اشتراک بگذارید از [\(+\)](#) استفاده کنید که برای این منظور طراحی شده است: HttpContext.Current.Items

نوبسند: A
تاریخ: ۱۳۸۹/۰۹/۱۹ ۰۲:۰۷:۵۲

بله کاملاً متوجه این موضوع بودم. در اینجا محیطی که برای آن طراحی می‌کنم نیز ASP.NET است. اما من حتماً اسرار دارم که به

ازاء هر Thread یک شی `DataContext` در حافظه داشته باشم و نه به ازاء هر درخواست. حال اگر Thread ASP.NET از استفاده مجدد می‌کند چه بهتر، از `Dispose` و `New` اضافی خودداری می‌شود. و البته علاوه بر این در نهایت باعث خواهد شد که طراحی این لایه از لایه `Presentation` بیشتر جدا شود و حتی برای Win نیز کاربرد داشته باشد.

البته! در ASP.NET اگر امکان داشت که دو همزمان از یک `Thread` دو خروجی متفاوت بگیرند با مشکل مواجه می‌شدم؛ که فکر می‌کنم چنین چیزی ممکن نیست.

در کل تنها مشکلی که داشتم این بود که وقتی فکر این اگوی طراحی به ذهنم خطور کرد از `ThreadStaticAttribute` اطلاع نداشتمن و با آن متدها که گفتم کار کردم. که توسط شما از این موضوع مطلع شدم.

باز هم ممنون برای نکات و ارجاء.

نویسنده: وحید نصیری

تاریخ: ۰۸:۴۷:۱۱ ۱۳۸۹/۰۹/۱۹

در ASP.NET این استفاده مجدد از یک `Thread` منحصر به یک سشن نیست. همچنین از یک ترد مشخص الزاماً برای درخواست بعدی استفاده نمی‌شود.

به علاوه `thread safe` نیز `DataContext` نیست و مباحثی را که در بالا ذکر شد در نظر داشته باشد. ممکن است وسط کار توسط یک کاربر دیگر استفاده شود.

+ این کار کردن با یک مرورگر و `load` یک کاربر ... روش صحیحی برای آزمودن نیست.

اگر خیلی علاقمند به انجام اینکار هستید باید از روش یک `DataContext` به ازای هر درخواست (per request) استفاده کنید، یعنی همان روش استفاده از `Current.Items` ذکر شده. اصطلاحاً به این الگو، الگوی `UnitOfWork` گفته می‌شود. یک پیاده سازی خوب در این مورد اینجا هست: [\(+\)](#) و [\(+\)](#)

نویسنده: A

تاریخ: ۱۱:۲۹:۱۹ ۱۳۸۹/۰۹/۱۹

آیا این امکان وجود دارد که یک متغیر که دارای ویژگی `ThreadStatic` است توسط دو همزمان مورد استفاده قرار بگیرد؟ جواب Microsoft خیر است. پس اینچنین متغیری نیاز به ThreadSafe بودن و حتی `lock` (فارق از محیط اجرا) ندارد. مگر اینکه راه دیگری برای اجرای چند کد به صورت موازی در یک `AppDomain` وجود داشته باشد. به عبارت دیگر من Thread را `Atomic` ترین شی موازی‌سازی می‌دانم. اشتباه می‌کنم؟

البته اینکه یک `Thread` لزوماً برای درخواستی بعدی استفاده نشود نیز مشکلی ایجاد نمی‌کند.

لینک‌ها را مطالعه کردم. ولی بازهم فکر می‌کنم مشکلی در استفاده از این روش نیست! با توجه به جمله‌ای که در بالا گفتم نمی‌دانم چه مشکلی می‌تواند از نظر Concurrency اتفاق بیفتد؟

در مورد تست، من هم با شما موافقم و تست کردن فقط با یک کاربر در یک مرورگر را مطمئناً تست کاملی نمی‌دانم. درواقع فقط با بررسی نقاط بحرانی و حالت‌های بحرانی در ذهنم موارد `Concurrent` را چک کرده‌ام. که البته حتماً سعی می‌کنم با استفاده از یک نرم‌افزار خوب یک محیط چند کاربره را شبیه‌سازی کنم.

راستی یک نکته خیلی مهم. در این روش هر شبی که `DataContext` را برای اولین بار به وجود آورده باید بعد از اتمام کارش خودش آنرا `Dispose` کند. تمامی عملیات کار با داده در یک بلاک `using` صورت می‌گیرد اما شبی که در `using` استفاده شده مانند یک شی `Repository` است که مدیریت `DataContext` را بر عهده دارد اما در داخل تفاوت‌هایی برای بالابردن کارایی دارد.

البته اسرار من بر این روش به جای استفاده از `HttpContext` این است که: لایه دسترسی به داده به صورت BlackBox کاملاً از لایه `Presentation` جدا باشد. درواقع بدون پیاده‌سازی یک `Interface` یا چیزی شبیه آن بتوان از یک لایه داده «همzman» توسط

چند لایه دیگر (در یک پروژه) استفاده شود و البته برنامه‌نویس به دردسر نیفتند.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۹/۱۹ ۱۱:۵۲:۴۱

Atomic Operation - داریم ولی شء؟

- بحث استفاده مجدد از یک ترد در ASP.NET به این معنا است:
- ترد یک در سشن کاربر یک، یک `DataContext` استاتیک را ایجاد کرده. حتی آنرا با ویژگی `ThreadStatic` هم مزین کرده است. اکنون به نظر در پایان درخواست کارش به پایان رسیده است. در این لحظه ASP.NET این ترد یک را در اختیار سشن کاربر 2 قرار می‌دهد. این `DataContext` استاتیک شما که با توجه به ویژگی `ThreadStatic` بودن آن در این ترد زنده است و جهت `Tracking` بسیاری از موجودیت‌ها از آن استفاده شده، اطلاعات خود را در اختیار کاربر 2 قرار داده است. این `DataContext` نه لزوماً می‌تواند معتبر باشد (شاید `dispose` شده) و یا شاید حاوی اطلاعات حساس و غیرضروری. هر دو مورد در یک برنامه چند کاربره مشکل ساز است.
- زمانیکه از ORM استفاده می‌کنید، لایه دسترسی به داده همان ORM است و از دید لایه‌های دیگر مخفی است. شما یک لایه دیگر به نام `BLL` برای جداسازی اعمال انجام شده توسط آن از لایه نمایش باید ایجاد کنید.

پ.ن.

- اسرار با اصرار کمی متفاوت است.

نویسنده: A
تاریخ: ۱۳۸۹/۰۹/۱۹ ۱۳:۴۶:۵۷

منظورم از Atomic این است که بیشتر از این نمی‌توانیم یک موجودیت را به اجزاء ریزتر تقسیم کنیم. مثل یک نقش(Role) اتمیک.

صحبت شما درست است ولی در این مورد این مشکلاتی که گفتید ایجاد نمی‌شود. عرض کردم که قبل از اینکه Thread بخواهد به کاربر 2 سپرده شود `DataContext` نابود شده است. معماری من کمی متفاوت است. فکر می‌کنم اگر بخواهیم بحث کنیم باید بر سر معماری بحث کنیم.

پاسخ اینکه آیا این معماری درست کار می‌کند یا نه در این جواب است که آیا امکان دارد دو Thread همزمان به یک شی که دارای ویژگی `ThreadStatic` است دسترسی پیدا کنند؟ اگر جواب خیر است پس همه چیز مرتب است (التبه کمی توضیح دارد).

و همچنین بله صحبت شما در مورد `BLL` درست است اما گاهی برای پروژه‌های کوچک بهتر است لایه‌ها با یک شبه `BLL` که داخل `DAL` قرار دارد صحبت کنند. البته فکر می‌کنم نمی‌توانم به درستی بیان کنم.

اما در کل شما به یک نکته خیلی خوب اشاره کردید و آن هم امنیت است. چک نکردم اما فکر می‌کنم در صورتی که «برنامه‌نویس خراب کاری کند» ممکن است این سیستم امنیت خوبی نداشته باشد. سعی می‌کنم برای این موضوع فکری کنم. اگر این مورد جدی باشد و اشتباهات برنامه‌نویس بتواند موجب مشکل شود حتماً از این روش صرف نظر کرده و از همان `HttpContext.Current` استفاده خواهم کرد.

دیکته من هرگز خوب نبوده و نیست. منظورم از اسرار، رازها نبوده منظورم پافشاری بوده (اصرار):

ممnon.

نویسنده: Nima
تاریخ: ۱۳۸۹/۰۹/۱۹ ۱۸:۱۴:۴۲

بسیار عالی، استفاده کردم و دقیقاً همچین اشتباهی رو داشتم. میرم که بیشتر تحقیق کنم
ممnon

نوبنده: میثم نوایی
تاریخ: ۱۳۸۹/۱۰/۱۴ ۱۷:۱۵:۴۱

من یکامپوننت چت آنلاین نوشتم و چون کاربران هر چهارثانیه یکبار لیست پیغام‌های دریافتی خود را چک میکند برای اینکه سرعت برنامه بالا باشه مجبور شدم کلیه پیغام‌هارو در یک لیست استاتیک قرار بدم. آیا کسی پیشنهاد بهتری نسبت به لیست استاتیک برای اینکار سراغ دارد؟ بازم سپاسگزارم.

نوبنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۰/۱۴ ۱۷:۳۱:۴۴

این مطلب جاری به معنای نفی استفاده از متغیرهای استاتیک نبود. اگر بد است چرا اصلاً در زبان قرار داده شده؟ بنابراین با دید صرفه منفی به این قضیه نگاه نکنید.
در کار شما آیا این لیست برای تمام کاربران یکسان است؟ آیا سطح دسترسی در کار نیست؟ آیا همه موارد مشابهی را مشاهده می‌کنند؟ اگر بله مشکلی ندارد، فقط در نظر داشته باشید که متغیرهای استاتیک thread safe نیستند. برای این موارد کلاس Cache قرار گرفته در فضای نام System.Web.Caching، مطابق مستندات آن Thread safe است و read/write کاربره مشکل زا نیست: [\(+\)](#)

ضمیرنا در مورد طراحی سیستم چت خوب در ASP.NET در مورد COMET تحقیق کنید. این روش سربار کمی دارد چون سرور به کلاینت پیغام ارسال می‌کند نه اینکه کلاینت متناظراً سرور را چک کند که آیا پیغام جدیدی هست یا نه: [\(+\)](#)

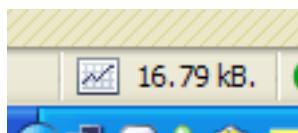
نوبنده: میثم نوایی
تاریخ: ۱۳۸۹/۱۰/۲۶ ۰۹:۵۸:۲۵

سپاسگزارم. راهنمایی‌های شما مخصوصاً معرفی این کامپوننت آخری خیلی بدرد بخور بود. احتمالاً بهترین پیشنهاد برای پیاده سازی سیستم چت استفاده از همین کامپوننت می‌باشد.

امکان اندازه گیری دقیق حجم ViewState در برنامه‌های ASP.NET WebForms وجود دارد (+)، اما خوب، این روش یک ایراد مهم هم دارد. چند نفر حاضرند تمام صفحات خود را ویرایش کرده و ارت بری ذکر شده را پیاده سازی کنند؟ یک روش دیگر اعمال آن به تمام صفحات، استفاده از پوششی استاندارد App_Browsers و سپس ایجاد فایلی مانند ViewStateManager.browser می‌باشد:

```
<!--Applies to all pages-->
<browsers>
<browser refID="Default">
  <controlAdapters>
    <adapter controlType="System.Web.UI.Page" adapterType="ViewStateManagerCore.SomeClass" />
  </controlAdapters>
</browser>
</browsers>
```

به این صورت ارت بری مورد نظر به صورت خودکار بر روی تمام صفحات اعمال خواهد شد. علاوه بر این دو روش (ارت بری دستی و ارت بری خودکار)، افزونه‌ای هم برای فایرفاکس جهت نمایش حجم ViewState صفحات طراحی شده است که از آدرس زیر قابل دریافت می‌باشد :



[Viewstate Size](#)

این موضوع چه اهمیتی دارد؟ ممکن است کاربران سایت شما گاهی از اوقات در بعضی از صفحات با خطای "Validation of viewstate MAC failed" مواجه و متوقف شوند. عموماً حجم بالای ViewState این مشکل را درست می‌کند. حجم ViewState بالا است (چند صد کیلوبایت...). صفحه دیر رندر می‌شود یا هنوز کامل نشده، شخص صفحه را متوقف می‌کند. ASP.NET در این حالت اجازه ارسال اطلاعات از این صفحه‌ی ناقص را به دلایل امنیتی نمی‌دهد که خوب است (شاید جعل شده باشد?).

پ.ن. راه حل پیشنهادی مايكروسافت جهت حل این مشکل (شروع شده از سال 2007)، ارتقاء برنامه‌های شما به ASP.NET MVC می‌باشد (:)

نظرات خوانندگان

نوبتند: shahin
تاریخ: ۱۳۸۹/۱۰/۱۹ ۱۲:۵۲:۰۷

سلام.

جناب نصیری حجم ViewState تا چقدر باشد ایرادی ندارد؟

مثلا حجم 7 کیلو بایت زیاد هست؟

شنیدم راههایی جهت فشرده سازی آن است.

آقای نصیری آیا ما برنامه نویسان ASP.NET MVC Web Form اگر به مهاجرت نکنیم ضرر می کنیم؟

ممnon

نوبتند: وحید نصیری
تاریخ: ۱۳۸۹/۱۰/۱۹ ۱۳:۰۰:۵۷

سلام،

- بستگی دارد. اگر شبکه داخلی است. کلا بحث اندازه گیری آن را فراموش کنید. چون سرعت دریافت بالا است. اگر برای اینترنت برنامه می‌نویسید مهم است و بالای 30 کیلوبايت به ازای هر صفحه را باید زیاد دانست.

- همان لینک داده شده در ابتدای مقاله را دنبال کنید. یک سری لینک در کامنت‌های آن برای فشرده سازی هست به همراه پیاده سازی کامل.

- هم...، باید منتظر نگارش بعدی ASP.NET بود یعنی نگارش 5 . اگر در نگارش 5 هم تغییرات آنچنانی حاصل نشده بود (مثل نگارش 4 فعلی که فقط یک سری موارد جزئی در آن تغییر کرده، مثلا فشرده سازی سشن، مشخص کردن ID سمت کلاینت و از این نوع بحث‌هایی که خیلی وقت است خود برنامه نویس‌ها برای آن راه حل پیدا کرده‌اند و ... به نظر من جدید نیستند)، آنگاه ... MVC تنها راه حل "زنده" و در حال پیگیری از طرف مایکروسافت تلقی خواهد شد.

در مورد ELMAH پیشتر مطالبی در این سایت منتشر شده است:

[معرفی ELMAH](#)

[ELMAH 1.1](#)

[یکپارچه کردن ELMAH با WCF RIA Services](#)

از آن می‌توان جهت ثبت استثناهای مدیریت نشده در انواع و اقسام برنامه‌های وب دات نتی استفاده کرد. از برنامه‌های ASP.NET Webforms ، ASP.NET MVC و WCF RIA سرویس‌های وغیره. سپس این Http module ویژه امکان مرور خطاهای ثبت شده را از راه دور میسر می‌سازد، به همراه فید RSS از خطاهای مشاهده‌ی مقادیر فیلدها در لحظه‌ی بروز خطا، ارسال ایمیل خطاهای امکان ذخیره سازی آن‌ها در فایل‌های XML تا انواع بانک‌های اطلاعاتی. برای نمونه، سایت Stackoverflow از یک نمونه‌ی سفارشی شده‌ی آن استفاده می‌کند ([+](#)).

همانطور که در این معرفی آمده است : ثبت استثناهای "مدیریت نشده". اما آیا امکان ثبت استثناهای مدیریت شده هم توسط آن وجود دارد؟ پاسخ: بله. به صورت زیر:

```
...
} catch (Exception ex)
{
    Elmah.ErrorSignal.FromCurrentContext().Raise(ex);
...
}
```

فقط کافی است ارجاعی به اسمبلی ELMAH اضافه شود و از سطر فوق استفاده گردد. به این صورت خروجی شبیه به همان خطاهای مدیریت نشده در لیست خطاهای ثبت شده توسط ELMAH ظاهر خواهد شد.

نظرات خوانندگان

نوبنده: Iman تاریخ: ۱۳۸۹/۱۱/۱۰ ۲۳:۴۲:۵۰

آیا سایت stackoverflow سورس بازه سورسشو از کجا می شد دانلود کرد؟

نوبنده: وحید نصیری تاریخ: ۱۳۸۹/۱۱/۱۱ ۰۰:۴۵:۴۷

نه. یا نه هنوز : (+)
ولی دیتابیس سایتشون رو میشه دریافت کرد و هر ماه یک نسخه جدید ارائه می‌دان: (+)

البته نسخه‌های مشابه نوشته شده : (+) و (+)
و یک سایت ایرانی نمونه: (+)

نوبنده: وحید نصیری تاریخ: ۱۳۸۹/۱۱/۱۶ ۱۰:۲۶:۱۹

تکمیل این بحث: در یک سری از متدهای فایل global.asax عموما Context ای وجود ندارد (دقیقاً مانند برنامه‌های دسکتاب). در این حالت از Elmah.ErrorLog.GetDefault(null).Log(new Error(ex)) استفاده کنید.

نوبنده: احسان تاریخ: ۱۳۹۲/۱۰/۱۴ ۱۳:۹

با سلام.
دستور فوق رو گذاشتم اما به Application_Error یا ErrorLog_Logged منتقل نشد. منظور از ارجاع به اسمبلی رو متوجه نشدم.
میشه یک نمونه مثال بزنید.
ممnon...

نوبنده: وحید نصیری تاریخ: ۱۳۹۲/۱۰/۱۴ ۱۳:۳۱

- به: منتقل نمی‌شود به روال‌های رویدادگردن یاد شده. سطر یاد شده در متن فوق، صرفاً خطای مدنظر را به لیست لاغ‌های قابل گزارشگیری ELMAH اضافه می‌کند.
- یک فایل DLL دارد به نام Elmah.dll. به این فایلهای DLL در دات نت، اصطلاحاً اسمبلی گفته می‌شود. افزودن ارجاعی به اسمبلی، یعنی مراجعه به منوی پروژه در VS.NET و سپس انتخاب گزینه‌ی Add reference در آنجا و انتخاب DLL یاد شده.
- در قسمت «مطالب مرتبط» ذیل هر بحث، یا حتی مقدمه‌ی بحث جاری، مثال‌های بیشتری ارائه شده‌اند.

چه زمان‌هایی یک برنامه‌ی ASP.NET ری استارت می‌شود؟

عنوان: وحید نصیری
نوبت‌نده: ۱۳۸۹/۱۱/۱۴
تاریخ: ۱۳:۳۳:۰۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net

- برای ری استارت کردن یک برنامه‌ی ASP.NET حتماً نیازی نیست تا IIS را متوقف و سپس راه اندازی کرد یا تنظیمات App pool را در IIS تغییر داد. روش‌های دیگری نیز وجود دارند که عدم آگاهی از آن‌ها می‌تواند سبب بروز مشکلات عدیده‌ای گردد و گاه‌ها خطایابی آن‌ها بسیار مشکل است؛ زیرا ری استارت شدن برنامه = از دست رفتن آنی تمام سشن‌های InProc تمام کاربران سایت؛ پاک شدن کش برنامه در IIS؛ از دست رفتن تمام متغیرهای استاتیک، Application State و مواردی از این دست:
- نوشتمن در پوشه bin برنامه (ایجاد فایل یا ایجاد پوشه یا تغییر نام پوشه‌ها و مواردی از این دست). (بنابراین قرار دادن بانک اطلاعاتی برنامه در این پوشه، اشتباه بوده و به همین جهت پوشه‌ی استاندارد App_Data طراحی شده است)
 - نوشتمن در فایل web.config برنامه (به صورت دستی، حتی در حد اضافه کردن یک فاصله یا توسط خود برنامه و یا استفاده از متده SetLastWriteTime FileSystemWatcher). در این حالت ASP.NET در این فایل، بروز خود فقط خواندنی با آن است نه ذخیره سازی تنظیمات پویای برنامه در آن. برای ذخیره سازی تنظیمات پویا، از بانک اطلاعاتی و یا یک فایل XML مجزای از وب کانفیگ استفاده کنید یا مواردی مشابه)
 - هر گونه تغییری در فایل‌ها و یا پوشه‌های ، App_WebReferences ، App_Code ، Global.asax ، machine.config ، App_LocalResources و App_GlobalResources
 - با ایجاد، حذف یا تغییر نام یکی از ساب دایرکتوری‌های واقع شده در ریشه برنامه. بنابراین اگر برنامه‌ی شما به صورت پویا پوشه‌هایی را ایجاد یا حذف می‌کند باید منتظر ری استارت‌های پی در پی باشید (البته این مورد با از کار انداختن FileChangesMonitor مربوط به HttpRuntime قابل حل می‌باشد ([+](#))، ولی همانطور که عنوان شد به صورت پیش فرض همواره فعال است)
 - فرآخوانی متده System.Web.HttpRuntime.UnloadAppDomain در برنامه‌های دسکتاب است و بلافاصله سبب خاتمه‌ی برنامه می‌شود. قرار دادن فایل App_Offline.htm در پوشه اصلی برنامه نیز چنین رفتاری را سبب خواهد شد. علاوه بر آن تگ httpRuntime در وب کانفیگ نیز دارای گزینه‌ی enable است و تنظیم آن به false، سبب خاتمه‌ی سرویس دهی برنامه خواهد شد.
 - رسیدن به عدد numRecompilesBeforeAppRestart تعریف شده در فایل machine.config که عموماً به عدد 15 تنظیم شده است. اگر تغییرات زیادی را در فایل‌های (مرتبط با ASP.NET مانند aspx، asmx و غیره) برنامه داده باشید (بیشتر از 15 مورد) و نیازی به ری کامپایل اساسی وجود داشته باشد، ASP.NET FileSystemWatcher به صورت خودکار برنامه را ری استارت خواهد کرد.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۱ ۱۰:۹

چند نکته‌ی تکمیلی

- اگر هاست شما تمام سایت‌ها را با یک Application pool مدیریت می‌کند، کرش یکی از چند ده سایت دیگر می‌تواند سبب ری استارت شدن سایت شما هم بشود؛ چون برنامه‌ها از همه ایزووله نشده‌اند. راه حل آن ایجاد یک Application pool مجزا به ازای هر سایت هست (توسط هاستدار). زمانیکه تمام سایت‌ها با یک Application pool واحد مدیریت می‌شوند، تمام آن‌ها توسط یک وله از w3wp.exe اجرا خواهند شد. با تعریف Application pool‌های مجزا، هر سایت، یک وله‌ی مجزا از w3wp.exe را به خود اختصاص خواهد داد. یعنی اگر سرور را بررسی کنید، به ازای هر سایت، یک pid با w3wp.exe مجزا قابل مشاهده است. به این ترتیب اگر pid=1234 کرش کرد، تاثیری روی pid=4321 نخواهد داشت.
- یک برنامه‌ی ASP.NET پس از مدتی بیکاری (قابل تنظیم در Application pool برنامه)، به صورت خودکار توسط IIS از حافظه خارج می‌شود. با درخواست بعدی که به آن برنامه می‌رسد (مثلاً گشودن یک صفحه‌ی آن توسط یک کاربر)، مجدداً از صفر اجرا خواهد شد. این مورد نیز به معنای ری استارت کامل برنامه است.
- در تنظیمات Application pool موارد زیادی را می‌توان تنظیم کرد که سبب ری استارت شدن برنامه می‌شوند. برای مثال اگر مصرف CPU و یا حافظه‌ی برنامه به حد مشخصی رسید، برنامه ری استارت شود و امثال آن.

با استفاده از AutoComplete TextBoxes می‌توان گوشاهی از زندگی روزمره‌ی کاربران یک برنامه را ساده‌تر کرد. مشکل مهم این است که در یک برنامه‌ی وب، عدم امکان تایپ قسمتی از متن مورد نظر و سپس نمایان شدن آیتم‌های متناظر با آن در اسرع وقت می‌باشد. همچنین با تعداد بالای آیتم‌ها هم حجم صفحه و زمان بارگذاری را افزایش می‌دهند. راه حل‌های بسیار زیادی برای حل این مشکل وجود دارند و یکی از آن‌ها ایجاد AutoComplete TextBoxes است. پلاگین‌های متعددی هم جهت پیاده سازی این قابلیت نوشته شده‌اند منجمله [jQuery Autocomplete](#). این پلاگین دیگر توسط نویسنده‌ی اصلی آن نگهداری نمی‌شود اما توسط برنامه نویسی دیگر در [github](#) ادامه یافته است. در ادامه نحوه استفاده از این افزونه را در ASP.NET Webforms بررسی خواهیم کرد.

الف) دریافت افزونه

لطفا به آدرس GitHub [ذکر شده](#) مراجعه نمائید.

سپس برای مثال پوشه‌ی js را به پروژه افزوده و فایل‌های jquery.autocomplete.js ، jquery-1.5.min.js ، indicator.gif و jquery.autocomplete.css را در آن کپی کنید. فایل indicator.gif به همراه مجموعه‌ی دریافتی ارائه نمی‌شود و یک آیکن [loading](#) معروف می‌تواند باشد. علاوه بر آن یک فایل جدید custom.js را نیز جهت تعاریف سفارشی خودمان اضافه خواهیم کرد.

ب) افزودن تعاریف افزونه به صفحه

در ذیل نحوه افزودن فایل‌های فوق به یک master page نمایش داده شده است. در اینجا از قابلیت‌های جدید ScriptManager (موجود در سرویس پک یک دات نت سه و نیم و یا دات نت چهار) جهت یکی کردن اسکریپت‌ها کمک گرفته شده است. به این صورت تعداد رفت و برگشت‌ها به سرور بهجای سه مورد (تعداد فایل‌های اسکریپت مورد استفاده)، یک مورد (نهایی یکی شده) خواهد بود و همچنین حاصل نهایی به صورت خودکار به شکلی فشرده شده به مرورگر تحویل داده شده، سرآیندهای کش شدن اطلاعات به آن اضافه می‌گردد (که در سایر حالات متدائل اینگونه نیست)؛ به علاوه URL نهایی آن هم بر اساس hash فایل‌ها تولید می‌شود. یعنی اگر محتوای یکی از این فایل‌ها تغییر کرد، چون URL نهایی تغییر می‌کند، دیگر لازم نیست نگران کش شدن و به روز نشدن اسکریپت‌ها در سمت کاربر باشیم.

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs"
Inherits="AspNetjQueryAutocompleteTest.Site" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <asp:PlaceHolder Runat="server">
        <link href="<%= ResolveClientUrl("~/js/jquery.autocomplete.css")%>" rel="stylesheet" type="text/css" />
    </asp:PlaceHolder>
    <asp:ContentPlaceHolder ID="head" runat="server">
        </asp:ContentPlaceHolder>
    </head>
<body>
    <form id="form1" runat="server">
        <asp:ScriptManager ID="ScriptManager1" runat="server">
            <CompositeScript>
                <Scripts>
                    <asp:ScriptReference Path "~/js/jquery-1.5.min.js" />
                    <asp:ScriptReference Path "~/js/jquery.autocomplete.js" />
                </Scripts>
            </CompositeScript>
        </asp:ScriptManager>
    </form>
</body>
</html>
```

```

        <asp:ScriptReference Path="~/js/custom.js" />
    </Scripts>
</CompositeScript>
</asp:ScriptManager>
<div>
    <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
        </asp:ContentPlaceHolder>
    </div>
</form>
</body>
</html>

```

علت استفاده از ResolveClientUrl در حین تعریف فایل css در اینجا به عدم مجاز بودن استفاده از ~ جهت مسیر دهی فایل‌های header در css صفحه بر می‌گردد.

ج) افزودن یک صفحه‌ی ساده به برنامه

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="default.aspx.cs" Inherits="AspNetjQueryAutocompleteTest._default" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <asp:TextBox ID="txtShenas" runat="server" />
</asp:Content>

```

فرض کنید می‌خواهیم افزونهی ذکر شده را به TextBox استاندارد فوق اعمال کنیم. ID این TextBox در نهایت به شکل ContentPlaceHolder1_txtShenas در ASP.NET 4.0 رender خواهد شد. البته در تنظیم ClientIDMode=Static می‌توان ID انتخابی خود را به این ID خودکار درنظر گرفت و اعمال کرد. اهمیت این مساله در قسمت (۵) مشخص می‌گردد.

د) فراهم آوردن اطلاعات مورد استفاده توسط افزونهی AutoComplete به صورت پویا

مهمنترین قسمت استفاده از این افزونه، تهیه‌ی اطلاعاتی است که باید نمایش دهد. این اطلاعات باید به صورت فایلی که هر سطر آن حاوی یکی از آیتم‌های مورد نظر است، تهیه گردد. برای این منظور می‌توان از فایل‌های ASHX یا همان Generic handlers استفاده کرد:

```

using System;
using System.Data.SqlClient;
using System.Text;
using System.Web;

namespace AspNetjQueryAutocompleteTest
{
    public class AutoComplete : IHttpHandler
    {
        public void ProcessRequest(HttpContext context)
        {
            string prefixText = context.Request.QueryString["q"];
            var sb = new StringBuilder();

            using (var conn = new SqlConnection())
            {
                //todo: این مورد باید از فایل کانفیگ خوانده شود
                conn.ConnectionString = "Data Source=(local);Initial Catalog=MyDB;Integrated Security = true";
                using (var cmd = new SqlCommand())
                {
                    cmd.CommandText = @" select Field1 ,Field2 from tblData where Field1 like @SearchText + '%' ";
                    cmd.Parameters.AddWithValue("@SearchText", prefixText);
                    cmd.Connection = conn;
                    conn.Open();
                    using (var sdr = cmd.ExecuteReader())
                    {
                        if (sdr != null)

```

```
        while (sdr.Read())
        {
            string field1 = sdr.GetValue(0) == DBNull.Value ? string.Empty :
sdr.GetValue(0).ToString().Trim();
            string field2 = sdr.GetValue(1) == DBNull.Value ? string.Empty :
sdr.GetValue(1).ToString().Trim();
            sb.AppendLine(field1 + "|" + field2);
        }
    }
    context.Response.Write(sb.ToString());
}
public bool IsReusable
{
    get
    {
        return false;
    }
}
```

در این مثال از ADO.NET کلاسیک استفاده شده است تا به عمد نحوه تعریف پارامترها یکبار دیگر مرور گردد. اگر از SQL یا Entity framework یا NHibernate و موارد مشابه استفاده می‌کنید، جای نگرانی نیست؛ زیرا کوئری‌های SQL تولیدی توسط این ORMs به صورت پیش فرض از نوع پارامتری هستند (+).

در این مثال اطلاعات دو فیلد یک و دوی فرضی از جدولی با توجه به استفاده از `like` تعریف شده دریافت می‌گردد. به عبارتی همان‌متد `StartsWith` معروف `LINQ` بکار گرفته شده است.

به صورت خلاصه افزونه، کوئری استرینگ `q` را به این فایل `ashx` ارسال می‌کند. سپس کلیه آیتم‌های شروع شده با مقدار دریافتی، از بانک اطلاعات، در یافته شده و هر کدام قارگ فته در یک سطح جدید بازگشت داده می‌شوند.

اگر دقت کرده باشید در قسمت `sb.AppendLine`، با استفاده از "`"`" دو مقدار دریافتی از هم جدا شده‌اند. عموماً یک مقدار کفایت می‌کند (در 98 درصد موارد) ولی اگر نیاز بود تا توضیحاتی نیز نمایش داده شود از این روش نیز می‌توان استفاده کرد. برای مثال یک مقدار خاص، به همراه توضیحات آن به عنوان یک آیتم نمایش، داده شده مدنظر است.

۵) اعماء، نهایه، افزونه به

در ادامه می‌باده سازی `custom.js`، فایل `index.html` استفاده از امکانات فریم‌کاری شده در قسمت‌های قبل، ارائه گردیده است:

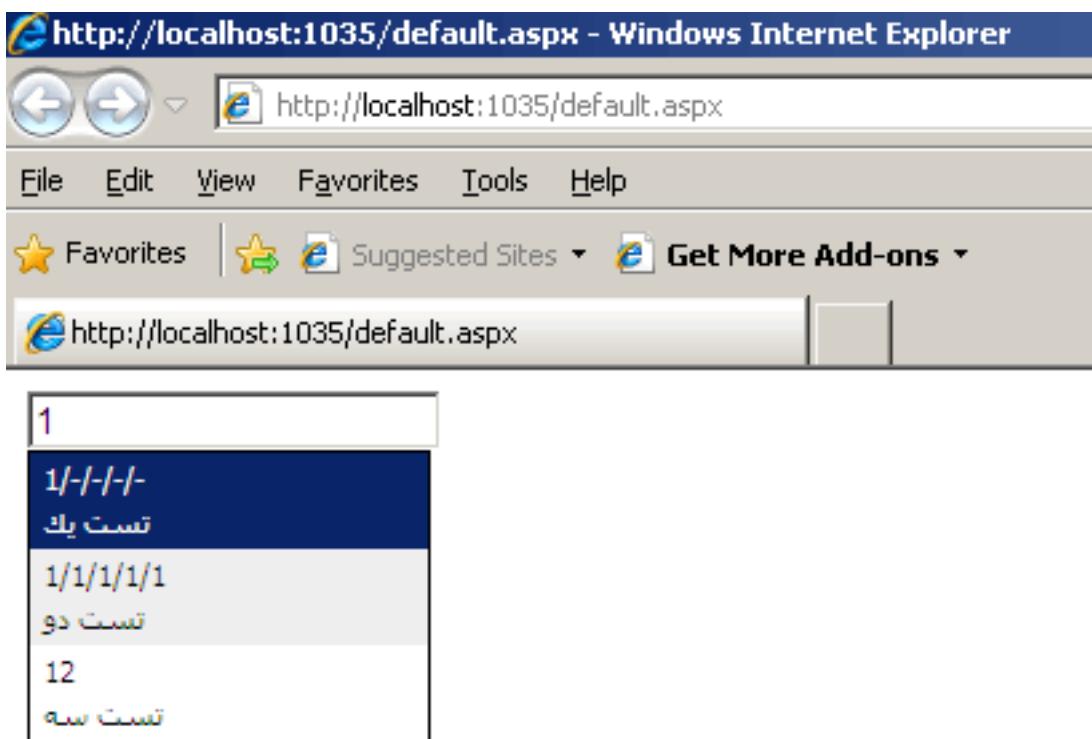
```
function formatItem(row) {
    return row[0] + "<br/><span style='text-align:justify;' dir='rtl'>" + row[1] + "</span>";
}

$(document).ready(function () {
    $("#ContentPlaceHolder1_txtShenas").autocomplete('AutoComplete.ashx', {
        //Minimum number of characters a user has to type before the completer activates
        minChars: 0,
        delay: 5,
        //Only suggested values are valid
        mustMatch: true,
        //The number of items in the select box
        max: 20,
        //Fill the input while still selecting a value
        autoFill: false,
        //The comparison doesn't looks inside
        matchContains: false,
        formatItem: formatItem
    });
});
```

بیس از این مقدمات، اعمال افزونه‌ی `autocomplete` به `ContentPlaceHolder1` مساوی `txtShenas` است اما `id` با `txtBox` ساده

است. اطلاعات از فایل AutoComplete.ashx دریافت می‌گردد و تعدادی از خواص پیش فرض این افزونه در اینجا مقدار دهی شده‌اند. لیست کامل آن‌ها را در فایل js/jquery.autocomplete.js می‌توان مشاهده کرد.

تنها نکته‌ی مهم آن استفاده از پارامتر اختیاری formatItem است. اگر در حین تهیه‌ی AutoComplete.ashx خود تنها یک آیتم را در هر سطر نمایش می‌دهید و از "|" استفاده نکرده‌اید، نیازی به ذکر آن نیست. در این مثال ویرثه، فیلد یک در یک سطر و فیلد دو در سطر دوم یک آیتم نمایش داده می‌شوند:



نظرات خوانندگان

نویسنده: شاهین کیاست
تاریخ: ۱۳۸۹/۱۱/۲۳ ۲۰:۱۵:۳۴

سلام آقای نصیری.
بسیار ممنونم :)
چند تا سوال دارم.

- 1) این روش چه برتری بر پلاگین UI jQuery دارد؟
- 2) می‌شود از Web Service هم استفاده کرد. آیا Generic handlers سرعت و کارایی بهتری می‌دهد؟
چگونه می‌توان ID رکوردی که بازگشته را در جایی نگه داشت تا جهت انجام عملیات سمت سرور از آن استفاده کرد.
با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۱/۲۳ ۲۱:۲۲:۵۱

سلام،
- برتری خاصی ندارد. فقط اگر کسی نمی‌خواهد از UI jQuery و فایل‌های آن استفاده کند، این روش هم هست و سبک‌تر است. در این مورد باز هم بگردید افزونه هست. البته بعدش باید بگردید به چه صورت باید دیتا به آن‌ها پاس کرد و روش‌ها یکی نیست.
- بله. حتی می‌شود از یک صفحه معمولی aspx هم استفاده کرد. در page_load آن کافی است این Response.Write ذکر شده قرار گیرد. در این مثال فوق از یکی از چندین روش ممکن، استفاده شده با سربار حداقل ممکن.
- سوال خوبیه. به عبارتی اگر قرار است بجای یک dropdown که قبلاً text هر ردیف آن متن خاصی بود و آن id value آن یا همان primary key رکورد جاری و از دید کاربر مخفی بود، الان باید چکار کرد تا همان id را بتوان به سرور ارسال کرد. چون به ظاهر فقط text دریافت می‌شود.
شما در اینجا از همان روش row_0 و row_1 که ذکر شد استفاده کنید. row_1 را با id در سمت سرور پر کنید field1+"|"+id .
سپس این پلاگین متده‌ی result دارد (مقدار انتخاب شده توسط کاربر را بازگشت می‌دهد). به این صورت فرصت خواهد بود تا یک فیلد مخفی را جهت ارسال به سرور مقدار دهی کرد:

```
(result(function (evt, row, formatted) {
    $("[hiddenIDbox]").val(row[1#]);
})
```


لزومی هم ندارد تا این id یا همان row_1 را در تابع formatItem نمایش داد. برای نمایش همان row_0 کافی است.

نویسنده: محسن نجف زاده
تاریخ: ۱۳۹۲/۰۹/۳۰ ۱۰:۲۵

پیشنهادات امنیتی تون برای استفاده از فایل‌های ASHX یا Generic handlers چیست؟
ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۳۰ ۱۰:۲۸

در مثال جاری با توجه به اینکه ورودی دریافتی از کاربر به کمک SQL پارامتری توسط ADO.NET پردازش می‌شود، مقاومت در برابر تزریق اس‌کیوال آن افزایش یافته است.

نویسنده: محسن نجف زاده
تاریخ: ۱۳۹۲/۰۹/۳۰ ۱۰:۴۶

منظورم در حالت کلی است.

یک موردش : "چنانچه نیاز به استفاده از Session در فایل ASHX است، واسط IReadonlySessionState است، به کار گرفته شود. "

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۳۰ ۱۰:۵۱

در حالت کلی تفاوت خاصی وجود ندارد با سایر قسمت‌ها و [مسایل دیگر امنیتی](#) ASP.NET. بیشتر، ورودی‌های دریافتی توسط آن را باید صحیح بکار بگیرید و مراقب آن‌ها باشید.

نکات نصب برنامه‌های ASP.NET 4.0 بر روی IIS 6

عنوان:

وحید نصیری

تاریخ:

۱۳۸۹/۱۱/۲۵ ۰۸:۴۱:۰۰

آدرس:

www.dotnettips.info

ASP.Net

گروه‌ها:

سه نکته مهم حین توزیع برنامه‌های ASP.NET 4.0 IIS 6.0 نسبت به سایر نگارش‌های قبلی وجود دارند که باید در نظر گرفته شوند:

الف) پس از انتقالی برنامه از نگارش‌های قبلی به دات نت 4 (با فرض اینکه دات نت 4 بر روی سرور نصب است)، پیغام 404 یا به عبارتی فایل مورد نظر بر روی سرور یافت نشد را دریافت می‌کنید (با تمام فایل‌های موجود):

در کنسول IIS ، ذیل قسمت Web Services Extensions ، باید دو مورد از حالت prohibited خارج شوند:

All unknown ISAPI extensions

ASP.NET 4.0

ب) پس از اجرای برنامه پیغام غیر معتبر بودن تگ‌های جدید فایل Web.Config را ملاحظه می‌کنید:

- در برگه‌ی خواص سایت در IIS 6.0 ، اکنون امکان انتخاب ASP.NET 4.0 هم میسر است که حتماً باید این مورد انتخاب گردد (تا دات نت سه و نیم این نام تنها 2.0 ASP.NET بود). در غیر اینصورت تگ‌های جدید فایل Web.Config شناخته نخواهند شد.

ج) بلافاصله پس از اجرای برنامه، پیغام Server Application Unavailable قابل مشاهده است:

نکته‌ی مهم دیگری که به همراه برنامه‌های دات نت 4 باید به آن توجه داشت، ضرورت اجرای آن‌ها در یک پروسه جدید است. پروسه جدید در IIS 6.0 به معنای یک Application pool جدید است. به عبارتی اگر هم اکنون بر روی IIS 6.0 شما برای مثال 2 برنامه‌ی دات نت سه و دات نت 4 قصد استفاده از یک Application pool را داشته باشند، پیغام Server Application Unavailable ظاهر خواهد شد، زیرا نمی‌توان این دو نگارش را تحت یک پروسه اجرا کرد. برای حل این مشکل باید یک Application pool جدید و اختصاصی را جهت برنامه‌ی ASP.NET 4.0 خود تعریف و انتساب دهید.

نظرات خوانندگان

نویسنده: حمید
تاریخ: ۱۳۹۲/۱۰/۰۲ ۱۱:۶

خیلی ممنون از آموزش‌های مفید و گرانقدرتون.
در مورد Application Pool ممکنه بیشتر توضیح بدین. اینکه فرمودین به ازای هر برنامه باید یک Application Pool ایجاد کرد، ازدیاد آنها معایبی را در بر نداره؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۰۲ ۱۱:۵۵

خیر. «باید» به ازای هر برنامه یک [application pool](#) مجزا درست کنید. برای مطالعه بیشتر:

- برای مدیریت نگارش‌های مختلف دات نت مورد استفاده (مطلوب جاری)
- جهت مدیریت بهتر میزان مصرف حافظه یک برنامه: [مرواری بر تاریخچه محدودیت حافظه مصرفی برنامه‌های ASP.NET در IIS](#)
- برای جلوگیری از ری‌استارت شدن بی‌جهت سایر برنامه‌ها: [چه زمان‌هایی یک برنامه‌ای ASP.NET ری‌استارت می‌شود؟](#)
- جهت کنترل میزان مصرف CPU یک برنامه: [بررسی علت CPU Usage بالای برنامه در حال اجرا](#)

مدتی قبل مطلبی تحت عنوان "[What's coming in the next version of ASP.NET Webforms](#)" منتشر شد (که نویسنده آن دقیقاً مشخص نیست این اطلاعات را از کجا آورده و همچنین تکذیبیه‌ای هم جایی در مورد آن صادر نشد...); بنابراین خلاصه‌ای از آن را با هم مرور خواهیم کرد:

اخیراً تمام توجه تیم ASP.NET معطوف نسخه‌ی MVC آن شده است؛ هر چند هنوز تعداد قابل توجهی از پروژه‌های ASP.NET بر اساس Webforms تهیه شده‌اند یا می‌شوند. همچنین برخلاف مطالب منتشره در انجمن‌ها یا بلاگ‌های مرتبط، تیم ASP.NET، نگارش Webforms را فراموش نکرده و حتی نگارش 4 آن نیز تعدادی از قابلیت‌های MVC مانند URL Routing، حجم کمتر ViewState و کنترل Viewstate بیشتر بر روی HTML نهایی را به همراه داشته است.
به روز رسانی‌های متوالی MVC (که اکنون به نگارش 3 رسیده است)، شاید این تصور را پیش آورده باشد که دیگر Webforms مرده است! اما مهترین دلیل به روز رسانی‌های دیر هنگام نسخه‌ی Webforms، یکی بودن اسمبلی‌های آن با مجموعه‌ی اصلی دات نت فریم‌ورک است (برخلاف نسخه‌ی MVC که به صورت افزونه‌ای برای این مجموعه ارائه شده است).

نسخه‌ی بعدی Webforms (حداقل) شامل تازه‌ها و پیشرفت‌های زیر خواهد بود:

MVC ModelBinders

در نسخه‌ی MVC مفهومی به نام Model binders وجود دارد. کار آن مقدار دهی مدل برنامه به صورت خودکار بر اساس اطلاعات وارد شده توسط کاربر در رابط کاربری برنامه است. برای مثال در Webforms داریم `employee.Name = txtName.Text`. به این معنا که مقدار Text یک جعبه‌ی متنی به نام txtName را به خاصیت Name شیء employee نسبت بد. اینکار (انقیاد اطلاعات رابط کاربر به مدل برنامه) با وجود Model binders در نسخه‌ی MVC به صورت خودکار انجام می‌شود. این مورد دو مزیت عمده را به همراه خواهد داشت: (الف) سادگی و حجم کمتر کد (ب) امکان تهیه ساده‌تر unit test جهت قسمت‌های مختلف برنامه (چون دیگر به `txtName` گره نخواهد خورد).
امکانات Model binders، گفته شده (مطابق مرجع فوق!) که قرار است جزئی از نگارش بعدی Webforms باشد ... (امیدوارم!)

بهبودهای حاصل شده در اعتبار سنجی نسخه‌ی بعدی Webforms شامل پیشرفت‌های اعتبارسنجی نسخه‌ی MVC نیز خواهد بود. به این معنا که امکان کنارگذاشتن کنترل‌های اعتبار سنجی Webforms و استفاده یکپارچه از امکانات jQuery فراهم خواهد شد (به این صورت دیگر شما محدود به یک سری کنترل از پیش تعیین شده نخواهید بود و امکان دسترسی به کوهی از [افزونه‌های اعتبار سنجی](#) jQuery را خواهید داشت).

CSS Sprites

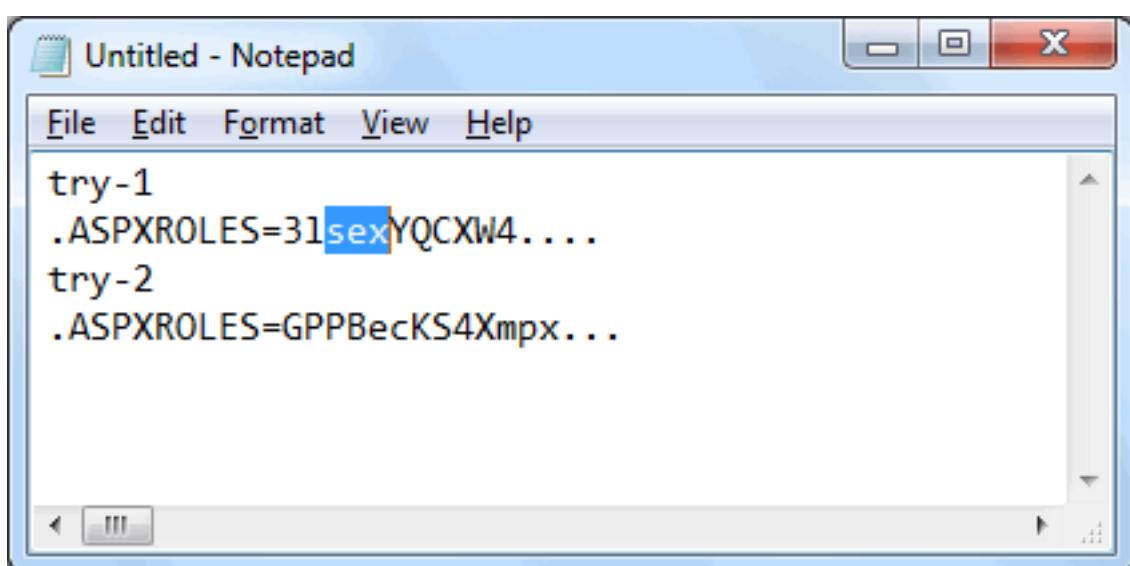
CSS Sprites که در نگارش بعدی Webforms پشتیبانی خواهد شد (+)، تکنیکی است جهت کاهش تعداد رفت و برگشت‌های به سرور با ارائه‌ی یک فایل حاوی تمام تصاویر قرار گرفته شده در یک شبکه یا گرید. به این صورت بجای دها یا صدها رفت و برگشت به سرور جهت دریافت تصاویر یک صفحه، تنها یک رفت و برگشت انجام خواهد شد.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۱/۲۷ ۱۶:۰۹

[Web Forms: Reports of my Death are Greatly Exaggerated](#)

از دیروز نمی‌توانستم به یک سایت آزمایشی ASP.NET وصل بشم. مشکل هم با فایرفاکس بود. با IE تست کردم و سایت وارد شد. بنابراین سایت فیلتر نشده بود. مشکل از کجا بود؟



بله ... سیستم فیلترینگ به هدرها هم حساس شده. کوکی را دستی پاک کردم. مجدداً پس از لاگین، با ساخته شدن کوکی جدید، مشکل برطرف شد.

این حروف هم به صورت اتفاقی تولید می‌شوند؛ بنابراین ممکن است با این نوع مسایل زیاد مواجه شوید! اگر یک سایت را با دو مرورگر متفاوت، یکی فیلتر شده و دیگری بدون مشکل باز کردید، یعنی یکبار باید کوکی‌های سایت را در مرورگری که با آن مشکل وجود دارد، پاک کنید!

خطای را هم که مشاهده خواهید کرد این است:

IMB2-6

[http://10.10.34.34?type=Invalid Pattern&policy>MainPolicy](http://10.10.34.34?type=Invalid%20Pattern&policy>MainPolicy)

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۱/۳۰ ۱۳:۰۵:۳۸

من امروز کارم شده پاک کردن کوکی‌های ASPXROLES . از این افزونه هم استفاده می‌کنم تا سریعتر باشد: ([^](#))

نویسنده: Salar
تاریخ: ۱۳۹۰/۱۱/۳۰ ۲۰:۲۸:۳۱

این مشکل تو کوکیهای یاهو هم هست. پس یه خاطر خودشون هم شده درستش می‌کنن!

نویسنده: Mohsen Ara
تاریخ: ۱۳۹۰/۱۲/۰۲ ۰۸:۴۸:۰۱

جالب بود چند روز پیش ولنتاین رو تو یاهور سرچ کردم فیلتر بود ولی x/e/ d رو سرچ کردم فیلتر نبود !!
معلوم نیست واقعاً چی به چیه !!!

نویسنده: nile
تاریخ: ۱۳۹۰/۱۲/۰۲ ۱۲:۲۵:۴۳

:)) چقدر باحال!!!!!!
چطوریه با هر بار لاغین یک کوکی جدید ایجاد میشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۲/۰۲ ۱۳:۰۶:۱۰

به این مبحث مراجعه کنید:
[Role Cookies & Cryptographically Random Keys](#)

چند وقتی هست که در کنار بدنه اصلی دات‌نت فریم‌ورک چندین کتابخونه به صورت متن باز در حال توسعه هستند. این مورد در ASP.NET بیشتر فعاله و مثلاً دو کتابخونه **WebApi** و **SignalR** توسط خود مایکروسافت توسعه داده می‌شود. **SignalR** هموطن‌طور که در سایت بسیار خلاصه و مفید یک صفحه‌ای! خودش توضیح داده شده ([^](#)) یک کتابخونه برای توسعه برنامه‌های وب «زمان واقعی»! ([real-time web](#)) است:

Async library for .NET to help build real-time, multi-user interactive web applications.

برنامه‌های زمان واقعی به صورت خلاصه و ساده به صورت زیر تعریف می‌شوند ([^](#)):

The real-time web is a set of technologies and practices that enable users to receive information as soon as it is published by its authors, rather than requiring that they or their software check a source periodically for updates.

یعنی کاربر سیستم ما بدون نیاز به ارسال درخواستی صریح! برای دریافت آخرین اطلاعات به روز شده در سرور، در برنامه کلاینتی از این تغییرات آگاه بشود. مثلاً برنامه‌هایی که برای نمایش نمودارهای آماری داده‌ها استفاده می‌شوند (بورس، قیمت ارز و طلا و ...) و یا مهمترین مثالش می‌توانه برنامه «چت» باشد. متاسفانه پروتوكول HTTP مورد استفاده در وب محدودیت‌هایی برای پیاده‌سازی این گونه برنامه‌ها دارد. روش‌های گوناگونی برای پیاده‌سازی برنامه‌های زمان واقعی در وب وجود دارد که کتابخونه **SignalR** فعلای از موارد زیر استفاده می‌کند:

تکنولوژی جدید **WebSocket** ([^](#)) که خوشبختانه پشتیبانی کاملی از اون در **دات نت 4.5** (چهار نقطه پنج! نه چهار و نیم!) وجود دارد. اما تمام مرورگرها و تمام وب سرورها از این تکنولوژی پشتیبانی نمی‌کنند و تنها برخی نسخه‌های جدید قابلیت استفاده از آخرين ورژن **WebSocket** را دارند که می‌شوند که کروم 16 به بالا و فایرفاکس 11 به بالا و اینترنت اکسپلورر 10 اشاره کرد (برای استفاده از این تکنولوژی در ویندوز نیاز به **IIS 8.0** است که متاسفانه فقط در ویندوز 8.0 موجود است):

Chrome 16, Firefox 11 and Internet Explorer 10 are currently the only browsers supporting the latest (. specification ([RFC 6455](#))

یه روش دیگه **Server-sent Events** نام داره که داده‌های جدید رو به فرم رویدادهای **DOM** به سمت کلاینت می‌فرسته ([^](#)).

روش دیگه‌ای که موجوده به **Forever Frame** معروفه که در این روش یک **iframe** مخفی درون کد **html** مسئول تبادل داده‌هاست. این **iframe** مخفی به صورت یک بلاک **Chunked** ([^](#)) به سمت کلاینت فرستاده می‌شود. این **iframe** که مسئول رندر داده‌های جدید در سمت کلاینت هست ارتباط خودش رو با سرور تا ابد! (برای همین بهش **forever** می‌گن) حفظ می‌کند. هر وقت رویدادی سمت سرور رخ میده با استفاده از این روش داده‌ها به صورت تگ‌های **script** به این فریم مخفی فرستاده می‌شوند و چون مرورگرها محتوای **html** رو به صورت افزایشی (incrementally) رندر می‌کنند بنابراین این اسکریپتها به ترتیب زمان دریافت اجرا می‌شوند. (البته ظاهرا عبارت **forever frame** در صنعت عکاسی! معروف‌تره بنابراین در جستجو در زمینه این روش ممکنه کمی مشکل داشته باشین) ([^](#)).

روش آخر که در کتابخونه **SignalR** ازش استفاده می‌شود **long-polling** نام داره. در روش **polling** معمولی پس از ارسال درخواست توسط کلاینت، سرور بلاfacله نتیجه حاصله رو به سمت کلاینت می‌فرسته و **ارتباط قطع می‌شود**. بنابراین برای داده‌های جدید درخواست جدیدی باید به سمت سرور فرستاده بشود که تکرار این روش باعث **افزایش شدید بار بر روی سرور** و کاهش کارآمدی اون می‌شود. اما در روش **long-polling** پس از برقراری ارتباط کلاینت با سرور این ارتباط تا مدت زمان معینی (که توسط یه مقدار تایم اوت مشخص می‌شود و مقدار پیش‌فرضش 2 دقیقه است) برقرار می‌ماند. بنابراین کلاینت می‌تواند بدون ایجاد مشکلی در کارایی، داده‌های جدید رو از سرور دریافت کند. به این روش در برنامه‌نویسی وب اصطلاحاً **برنامه‌نویسی کامت (Comet)** ([^](#) [^](#)) می‌گن ([Programming](#)).

(البته روش‌های دیگری هم برای پیاده‌سازی برنامه‌های زمان اجرا وجود داره مثل کتابخونه `node.js` که جستجوی بیشتر به خواندنگان واگذار می‌شود)

SignalR برای برقراری ارتباط ابتدا بررسی می‌کنے که آیا هر دو سمت سرور و کلاینت قابلیت پشتیبانی از WebSocket را دارند. در غیراینصورت سراغ روشن Server-sent Events میره. اگر باز هم موفق نشد سعی به برقراری ارتباط با روشن forever frame می‌کنے و اگر باز هم موفق نشد در آخر سراغ long-polling میره.

با استفاده از SignalR شما میتوانید از سرور، متدهایی را در سمت کلاینت فراخونی کنید. یعنی درواقع با استفاده از کدهای سی شارپ می‌شود متدهای جاوااسکریپت سمت کلاینت را صدای زد!

بطور خلاصه در این کتابخونه دو کلاس پایه وجود دارد:

PersistentConnection سطح پایین

کلاس سطح بالای Hub

علت این نامگذاری به این دلیله که کلاس سطح پایین پیاده‌سازی پیچیده‌تر و تنظیمات بیشتری هم در اختیار برنامه‌نویس قرار می‌دهد.

خوب پس از این مقدمه نسبتاً طولانی برای دیدن یک مثال ساده میتوانید با استفاده از نوگت (Nuget) مثال زیر را نصب و اجرا کنید (اگه تا حالا از نوگت استفاده نکردیدن قویا پیشنهاد می‌کنم که کار رو با دریافت از [اینجا آغاز کنید](#)):

```
PM> Install-Package SignalR.Sample
```

پس از کامل شدن نصب این مثال اون رو اجرا کنید. این یک مثال فرضی ساده از برنامه نمایش ارزش آنلاین سهام برخی شرکتهاست. میتوانید این برنامه رو همزمان در چند مرورگر اجرا کنید و نتیجه رو مشاهده کنید.
حالا میریم سراغ یک مثال ساده. میخوایم یک برنامه چت ساده بنویسیم. ابتدا یک برنامه وب اپلیکیشن خالی رو ایجاد کرده و با استفاده از دستور زیر در خط فرمان نوگت، کتابخونه SignalR را نصب کنید:

```
PM> Install-Package SignalR
```

پس از کامل شدن نصب این کتابخونه، ریفرنس‌های زیر به برنامه اضافه می‌شوند:

```
Microsoft.Web.Infrastructure
Newtonsoft.Json
SignalR
SignalR.Hosting.AspNet
SignalR.Hosting.Common
```

برای کسب اطلاعات مختصر و مفید از تمام اجزای این کتابخونه به [اینجا](#) مراجعه کنید.
همچنین اسکریپت‌های زیر به پوشش Scripts اضافه می‌شوند (این نسخه‌ها مربوط به زمان نگارش این مطلب است):

```
jquery-1.6.4.js
jquery.signalR-0.5.1.js
```

بعد یک کلاس با نام SimpleChat به برنامه اضافه و محتوای زیر را در اون وارد کنید:

```
using SignalR.Hubs;
namespace SimpleChatWithSignalR
{
    public class SimpleChat : Hub
    {
        public void SendMessage(string message)
        {
            Clients.receiveMessage(message);
        }
    }
}
```

```
}
```

دقیق کنین که این کلاس از کلاس Hub مشتق شده و همچنین خاصیت **Clients** از نوع **dynamic** است. (در مورد جزئیات این کتابخونه در قسمت‌های بعدی توضیحات مفصل‌تری داده می‌شود) سپس یک فرم به برنامه اضافه کرده و محتوای زیر را در آون اضافه کنید:

```
<input type="text" id="msg" />
<input type="button" value="Send" id="send" /><br />
<textarea id='messages' readonly="true" style="height: 200px; width: 200px;"></textarea>
<script src="Scripts/jquery-1.6.4.min.js" type="text/javascript"></script>
<script src="Scripts/jquery.signalR-0.5.1.min.js" type="text/javascript"></script>
<script src="signalr/hubs" type="text/javascript"></script>
<script type="text/javascript">
    var chat = $.connection.simpleChat;
    chat.receiveMessage = function (msg) {
        $('#messages').val($('#messages').val() + "-" + msg + "\r\n");
    };
    $.connection.hub.start();
    $('#send').click(function () {
        chat.sendMessage($('#msg').val());
    });
</script>
```

همونطور که می‌بینید برنامه چت ما آماده شد! حالا برنامه رو اجرا کنید و با استفاده از دو مرورگر مختلف نتیجه رو مشاهده کنید. نکته کلیدی کار SignalR در خط زیر نهفته است:

```
<script src="signalr/hubs" type="text/javascript"></script>
```

اگر محتوای آدرس فوق رو دریافت کنید می‌بینید که موتور این کتابخانه تمامی متدهای موردنیاز در سمت کلاینت رو با استفاده از کدهای جاوااسکریپت تولید کرده. البته در این کد تولیدی از نامگذاری camel Casing استفاده می‌شود، بنابراین متدهای SendMessage در سمت سرور به صورت sendMessage در سمت کلاینت در دسترسه. امیدوارم تا اینجا توانسته باشید علاقه شما به استفاده از این کتابخونه رو جلب کرده باشید. در قسمت‌های بعد موارد پیشرفته‌تر این کتابخونه معرفی می‌شود.

اگه علاقه‌مند باشید می‌توانید از [این ویکی](#) اطلاعات بیشتری بدست بیارید.

به روز رسانی در دوره‌ای به نام SignalR در سایت ، به روز شده‌ای این مباحث را می‌توانید مطالعه کنید.

نظرات خوانندگان

نویسنده: روح الله
تاریخ: ۱۳۹۱/۰۳/۲۹ ۰:۸

جالب بود. اتفاقاً امروز با دوستان صحبت از همچین نیازمندی بود. ممنون از زحمات شما.

نویسنده: افشار محب
تاریخ: ۱۳۹۱/۰۳/۲۹ ۹:۱۷

به نظر فریمورک جذاب و مفیدی میاد. همینجا به آقای نصیری و همکارانش به خاطر راه اندازی و بلاگ جدید تبریک می‌گویم.

نویسنده: علیرضا جهانشاهلو
تاریخ: ۱۳۹۱/۰۳/۲۹ ۹:۳۹

خیلی عالی بود! اگه بدونید این کتابخونه پایه اساس خیلی از ایده‌های تجاریه؟! که من در حال حاضر مشغول به کار بر روی یکی از همین ایده‌ها هستم.

نویسنده: احمد
تاریخ: ۱۳۹۱/۰۳/۲۹ ۱۹:۵۶

سلام و تبریک برای سایت جدید
از اینکه مطلب بسیار مفید و خوبی گذاشتید بسیار سپاسگزارم.اما اسکریپت `signalr/hubs` کجا قرار داره؟

نویسنده: Mona
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۰:۳۵

با سلام خدمت دوستان عزیز
ممنون از مطالب دقیق و کاربردی شما
لطفاً اگر امکان دارد مقایسه ای بین SignalR و Nod.js انجام دهید، زیرا در هر کجای وب که قدم می‌گذارید بحث از Nod.js است.
ممنون

نویسنده: یوسف نژاد
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۱:۴۶

این مسیر درواقع آدرس پیشفرض `signalr` برای کلاس پایه Hub هست که البته می‌توانه عوض هم بشه. این آدرس درواقع به یه هندرل مپ شده و این هندرل به محض دریافت یه درخواست از سمت کلاینت شروع به گشتن کل اسمبلی برای یافتن کلاس‌های مشتق شده از Hub می‌گردد و برای تمام اجزای public اوナ جهت در دسترس بودن در سمت کلاینت متدهای جاوا اسکریپت متناظر تولید کرده و در متن پاسخ ارسالی مینویسه (البته یکسری کد دیگه هم برای کار با کلاس Hub تولید می‌کنه).
برای دریافت این اسکریپت تولیدی، شما می‌توانید این مسیر رو در نوار آدرس مرورگرتون وارد کنید و کد جاوا اسکریپت تولیدی رو مشاهده کنید. حتی می‌توانید این کد تولیدشده رو در یک فایل ذخیره کرده و به جای ریفرنس اصلی استفاده کنید. البته به شرطی که در اجزای پابلیک کلاس‌هاتون بعداً تغییری ایجاد نشه.

نویسنده: ramin
تاریخ: ۱۳۹۱/۰۳/۳۰ ۱۰:۵۹

سلام

یعنی مسیر و محتوای `signalr/hubs` در هتگام اجرا تولید می‌شه؟
من مثال شما رو دنبال کردم ولی خروجی مورد نظر رو نگرفتم

نویسنده: یوسف نژاد
تاریخ: ۱۳۹۱/۰۳/۳۰ ۱۱:۱۰

من خودم تخصصی در زمینه [node.js](#) ندارم. البته ازش خوشم هم نمی‌یاد:)
اینچوری که خودشون می‌گن این یه پلتفرم که از [Chrome's JavaScript runtime](#) استفاده می‌کنه و شونصدتا ویژگی دیگه هم داره!
برای استفاده ازش باید برنامشو رو که بهش node می‌گن دریافت و نصب کنین (حدود 3 مگابایت نسخه 0.6.19). بعد کد سمت سرور کاملا با استفاده از زبان جاوااسکریپت نوشته می‌شه (عیب) بنابراین کاملا cross platform هستش (مزیت). یعنی با استفاده از جاوااسکریپت می‌توینی مثلای وب سرور بسیار سبک (مزیت) ایجاد کنین که به یکسری درخواستا پاسخ میده. حالا یکسری اومدن برای این پلتفرم مازولهایی نوشتن مثل [nowJS](#) و [socket.io](#) که برقراری ارتباطی راحت و آسان با سرور و تبادل داده‌ها رو به عهد می‌گیره (مزیت).

در مقابل [SignalR](#) یه کتابخونه غیرهمزمان (Async) هستش که برای دات نت نوشته شده (مزیت)، مخصوص وب زمان واقعی چند کاربره و شونصدتا ویژگی دیگه هم نداره (مزیت):). بنابراین سمت سرور از مزایای بیشماری برخورداره، مثل یک IDE قدرتمند (VS) (مزیت 2x) و یک کتابخونه کامل (.NetFx) (مزیت 2x).
در ضمن یه فریمورک دیگه هم برای ASP.NET وجود داره: [PokeIn](#) که نسخه تجاری (پولی) هم داره و از WebSocket استفاده می‌کنه.
جایی خوندم که کاراییش خوبه در حد 10 تا 50 هزار کلاینت همزمان ([^](#)).

برای SignalR یه اپلیکیشن توسط خود نویسنده‌گانش نوشته شده ([SignalR-Crank](#)) که برای آزمایش بار روی سرور استفاده می‌شود. طبق گفته خودشون تا 100k (صد هزار!) کلاینت رو توپنستن بدون هیچ مشکلی راه بندازن (البته با مصرف 5 گیگ رم).
نمیدونم این تست چی بوده ولی این عدد خیلی زیاده.

در مورد بحث کارآئی خودم دارم یه تستایی انجام میدم. نمیدونم بتونم آزمایش مشابهی رو بقیه فریمورکها انجام بدم یا نه. اگه نتیجه‌ای حاصل شد اینجا می‌زارم.

با تشکر

نویسنده: یوسف نژاد
تاریخ: ۱۳۹۱/۰۳/۳۰ ۱۱:۱۴

اگه برنامه چت داره درست اجرا می‌شه، مسیر مورد نظر برای دریافت این کد جاوااسکریپت مثلای رو سیستم من اینه:
<http://localhost:16869/signalr/hubs>

آره در زمان اجرا تولید می‌شه و بار کوچیکی روی سرور می‌زاره. برای همین پیشنهاد می‌شه تو نسخه ریلیز برنامه‌ها این کد تولیدی تو
یه فایل ذخیره بشه و به جای اون مسیر ریفرنس داده بشه.

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۰۷/۲۹ ۱:۲

سلام

ممnonم بابت مطلب خوب کاربردیتون
می تونم بگم یکی از سئوالهای ذهن من من پاسخ داده شد و این امکان راه حل مشکل چند هفته ای من بود
تشکر

نویسنده: hossein
تاریخ: ۱۳۹۲/۰۳/۰۸ ۱۲:۲۶

با سلام میخواستم بپرسم ایا این روش (signalr) برای لود کردن مثل 50-40 تا عکس اسلاید شو کارایی بالاتری داره یا روش
jquery lazy loading پلاگینی برای اون وجود داره
با تشکر

SignalR

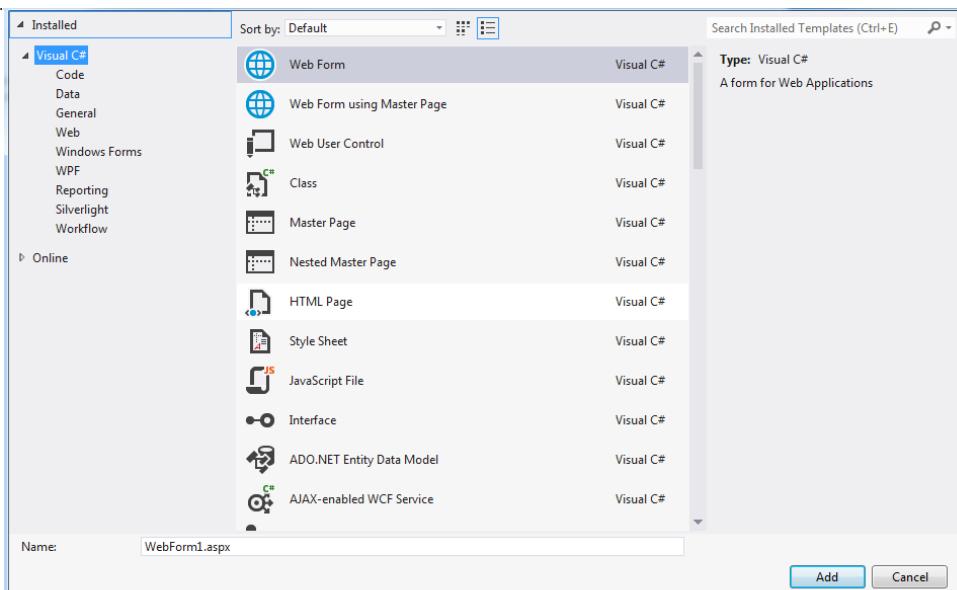
نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۳/۰۸ ۱۲:۳۳

سئوالت ربطی به بحث نداره. [کار SignalR به زبان ساده](#) فرستادن پیغام از طرف سرور به کلاینت‌ها هست. مثلًا ۲ تا ایمیل جدید داری. رکورد جدیدی به صفحه اضافه شده. یا برای نمونه درصد پیشرفت عملیات سمت سرور مثلاً ۵۰ درصد بوده و ارسالش به تمام کلاینت‌های متصل. هدف تبادل اطلاعات همزمان و بلادرنگ و بدون معطلی هست؛ خصوصاً از طرف سرور به کلاینت.

نویسنده: ابوالفضل روشن ضمیر
تاریخ: ۱۳۹۲/۰۵/۳۰ ۱۶:۲۵

سلام

من پکیج Jquery و SignalR را در برنامه نصب کردم ... ولی وقتی روی نام پروژه راست کلیک می‌کنم تا کلاس Class را اضافه کنم وجود نداره
ممnon



نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۳۰ ۱۷:۵۹

- آپدیت سوم VS 2012 را نصب کنید.
+ هیچ نیازی به این پیشنبازها نیست، با ۲۰۱۰ VS [کل مطالب SignalR](#) قابل پیاده سازی هستند. فقط باید بتونید با نیوگت کار کنید و بسته‌های لازم رو اضافه کنید. بعد از آن [کلاس هاب](#)، یک کلاس ساده است که امضای مشخص داره و از کلاسی به نام Hub مشتق می‌شود و همچنین ویژگی HubName هم می‌تواند داشته باشد.

نویسنده: mostafa
تاریخ: ۱۳۹۲/۰۹/۳۰ ۲۳:۱۶

سلام

میشه این رو برای website معمولی visual studio استفاده کرد؟ یعنی غیر از وب اپلیکشن ممنون

نویسنده: bahman

تاریخ: ۱۱:۵۴ ۱۳۹۲/۱۰/۰۹

نوبت داده شد

تاریخ:

سلام.
آیا این امکان وجود دارد که SignalR رو با RestFull سرویس در سمت سرور ترکیب کنم و از سرویس در سمت کلاینت، داخل یک برنامه WPF استفاده کنم؟

نویسنده: وحید نصیری

تاریخ: ۱۳:۴۴ ۱۳۹۲/۱۰/۰۹

نوبت داده شد

نگاهی به گزینه‌های مختلف مهیایی جهت میزبانی SignalR

نویسنده: عرفان

تاریخ: ۱۱:۲۲ ۱۳۹۳/۰۳/۲۵

نوبت داده شد

(Clients.receiveMessage(message
خطا میده. همچین کلاسی وجود نداره حتی اگه غلط املایش رو هم درست کنیم!

نویسنده: محسن خان

تاریخ: ۱۱:۳۶ ۱۳۹۳/۰۳/۲۵

نوبت داده شد

شما که به غلط املایش دقت کردی، به انتهای بحث که نوشته شده این مباحثت به روز شده‌اش در دوره SignalR سایت ارائه شدند، دقت نکردی؟ اون متدهای dynamic هست؛ یعنی اصلاً نیازی نیست وجود خارجی داشته باشد. فقط کمی در نگارش‌های جدید، Refactoring انجام دادن، بعدش باید مشخص کنی به All یا به گروه خاصی این پیام‌ها ارسال بشه. کلیاتش یکی هست. فقط کمی تعاریف اولیه رو Refactor کردن. در بحث معرفی hubs دوره‌ای که نام برده شد این‌ها هست.

یکی از ضروریات دنیای برنامه نویسی امروز، داشتن یک الگوی مناسب می‌باشد. یکی از الگوهای مناسب برای وب فرم‌ها، استفاده از الگوی MVP است.

اگر در خلال پیاده سازی، گاهی اوقات نیاز به handle کردن رخدادها را داشته باشید بدین منظور به روش زیر عمل می‌کنیم:
(توجه: شیء مورد نظر ما در این پست RadGrid از کنترلهای Telerik در نظر گرفته شده است).

```
// ASPX page
<telerik:RadGrid ID="RadGrid1" runat="server"></telerik:RadGrid>
// Asp.Net Code Behind
protected void Page_Load(object sender, EventArgs e)
{
    GridPresenter presenter = new GridPresenter(this);
}
// view interface
public interface IGridView
{
    Telerik.Web.UI.RadGrid myGrid { get; }
}
// presenter
protected readonly IGridView _view;
public GridPresenter(IGridView view)
{
    _view = view;
    _view.myGrid.UpdateCommand += new Telerik.Web.UI.GridCommandEventHandler(onUpdateCommand);
    _view.myGrid.InsertCommand += new Telerik.Web.UI.GridCommandEventHandler(onInsertCommand);
    _view.myGrid.EditCommand += new Telerik.Web.UI.GridCommandEventHandler(onEditCommand);
}
private void onUpdateCommand(object sender, Telerik.Web.UI.GridCommandEventArgs e)
{
    // Code for updating
}
private void onInsertCommand(object sender, Telerik.Web.UI.GridCommandEventArgs e)
{
    // Code for inserting
}
private void onEditCommand(object sender, Telerik.Web.UI.GridCommandEventArgs e)
{
    // Code for editcommand
}
```

نظرات خوانندگان

نویسنده: حسین مرادی نیا
تاریخ: ۱۳۹۱/۰۳/۲۹ ۰:۴۹

خوبه

اما من یه قسمت رو متوجه نشد. ببینید ما میگیم (MVP) Model View Presenter بیبینید ما میگیم (MVP) (Model View Presenter).
خب حالا سوال اینجاست که در بخش Asp.net Code Behind از چه کدی استفاده میکنیم؟
یعنی کدهای View و Presenter چه موقع فراخوانی میشوند؟
از کجا اجرا میشود و به کجا ختم میشود؟
مدل ما کجاست؟
...

نویسنده: امیر هاشم زاده
تاریخ: ۱۳۹۱/۰۳/۲۹ ۱۱:۱

ابتدا وب فرم مورد نظر اینترفیس IGridView را پیاده سازی میکند و سپس در Page_Load یک وله از presenter با پارامتر this ایجاد میکند، در نتیجه کلیه رخدادها در presenter مدیریت میشوند.
نکته در این است که چون هر بارگذاری صفحه نیاز به متصل کردن رخدادها به شی هستیم مجبوریم در Page_Load یک وله از presenter ایجاد کنیم.
در این پست هدف پیاده سازی مدل نبوده است، ولی شما میتوانید مدل خود را با توجه به الگوی MVP طراحی کنید و از آن در presenter استفاده کنید.

نویسنده: حسین مرادی نیا
تاریخ: ۱۳۹۱/۰۳/۲۹ ۱۴:۲۶

مرسی

مشکل قطعه کد زیر بود که جا افتاد بود و باعث شده بود که نحوه استفاده رو متوجه نشم!

```
GridPresenter presenter = new GridPresenter(this);
```

در مورد Model هم شما درست میگید. هدف این پست استفاده از Model نبود.

نویسنده: جلال
تاریخ: ۱۳۹۱/۰۷/۲۲ ۱۲:۴۱

سلام،

با این قضیه ارجاع مستقیم به یک کنترل خاص در IView میتونم کنار بیام، ولی با ارجاع به اون در CPresenter نه!
دلیل خاصی داشته؟ آخه معمولا برای رخدادگردانی، EventHandler ها رو در IView معرفی میکنن و در CPresenter استفاده میکنند.

البته در [این مقاله](#)، کلا منکر استفاده از EventHandler ها در IView میشه و ترجیح میده فقط تابع Presenter رو از Code Behind.

نویسنده: امیر هاشم زاده
تاریخ: ۱۳۹۱/۰۸/۱۶ ۱۸:۴۸

یکی از دلایل استفاده از این روش، تمیز نگه داشته داشتن ویو (با توسعه برنامه پارامترهای لازم برای 4 عمل اصلی اضافه میشد

که نگهداری پارامترهای رخداد گردانها پیچیده و دشوار می‌شد) بود، زمانی که داشتم این پیاده سازی رو در پروژه انجام می‌دادم به این نتیجه رسیده بودم بهتر است کلیه عملیات‌های افزودن، حذف، ویرایش و حذف شی تریک گرید ویو رو در presenter انجام بدم چون قرار بود یک سری عملیات منطقی بر روی 4 عمل اصلی انجام بدم.

ذخیره سازی اطلاعات بازدیدهای کاربران، در طول زمان حجم بالایی از بانک اطلاعاتی را به خود اختصاص خواهد داد؛ به علاوه کند شدن کوئری‌های مرتبط با آن، به همراه مصرف بالای منابع سیستم. به همین جهت اکثر سایتها از [Google analytics](#) برای مدیریت جمع آوری بازدیدهای کاربران خود استفاده می‌کنند و این ابزار واقعاً عالی و حرفه‌ای طراحی شده و پیاده سازی همانند آن شاید در حد یک پروژه‌ی چندساله باشد.

اضافه کردن [Google analytics](#) به یک سایت، بسیار ساده است. در آن ثبت نام می‌کنید؛ سپس آدرس دومین خود را وارد کرده و یک قطعه کد جاوا اسکریپت را دریافت خواهید کرد که باید به انتهای تمام صفحات سایت خود اضافه نمایید و ... همین. اضافه کردن این کد در ASP.NET MVC می‌تواند در فایل layout یا همان master page سایت انجام شود تا به صورت خودکار به تمام صفحات اعمال گردد.

مشکل!

من نمی‌خواهم که صفحات غیرعمومی سایت نیز دارای کدهای Google analytics باشند و بی‌جهت Google به اینجاها نیز سرکشی زاید کند! چکار باید کرد؟

احتمالاً عنوان می‌کنید که باید یک if و else به همراه آرایه‌ای از نام‌ها و آدرس‌های صفحات غیرعمومی سایت تهیه کرد و بر این اساس کدهای Google analytics را در master page درج کرد یا خیر.

بله. این روش کار می‌کنه ولی بهینه نیست و همچنین نگهداری آن در طول زمان مشکل است. سایت توسعه خواهد یافت، صفحات غیرعمومی بیشتر خواهند شد و ممکن است در این بین فراموش شود که کدهای مرتبط به روز شوند.

روش بهتر :

آیا می‌توان در یک View مشخص کرد که فیلتر Authorize در اکشن متداشتی که آنرا رندر کرده است بکار گرفته شده است یا خیر؟

صفحات غیرعمومی سایت در ASP.NET MVC با فیلتر Authorize محافظت می‌شوند. این فیلتر را می‌توان به کل یک کنترلر اعمال کرد تا به تمام اکشن‌های آن اعمال شود؛ یا فقط به یک اکشن متداشتی View مخصوص که آنرا رندر می‌کند.

نحوه پیاده سازی تشخیص وجود فیلتر Authorize را در یک View رندر شده، در متداشت مشاهده کرد:

```
@helper IncludeGoogleAnalytics(WebViewPage page)
{
  var controller = page.ViewContext.Controller;
  var controllerHasAuthorizeAttribute =
controller.GetType().GetCustomAttributes(typeof(AuthorizeAttribute), true).Any();
  var currentActionName =
page.ViewContext.Controller.ValueProvider.GetValue("action").RawValue.ToString();
  var actionHasAuthorizeAttribute = controller.GetType().GetMethods()
    .Where(x => x.Name == currentActionName &&
x.GetCustomAttributes(typeof(AuthorizeAttribute), true).Any())
    .Any();
  if (!controllerHasAuthorizeAttribute && !actionHasAuthorizeAttribute)
  {
    string trackingId = ConfigurationManager.AppSettings["GoogleAnalyticsID"];
<script type="text/javascript">
  var _gaq = _gaq || [];
  _gaq.push(['_setAccount', '@trackingId']);
  _gaq.push(['_trackPageview']);
  (function () {
    var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
    ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') +
'.google-analytics.com/ga.js';
    var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);
  })();
</script>
}
```

توضیحات:

این متد در فایلی به نام `HtmlUtils` قرار گرفته در پوشه `app_code` تعریف شده است و بکارگیری آن در یک فایل `master page` به نحو زیر خواهد بود:

```
@HtmlUtils.IncludeGoogleAnalytics(this)
```

در این متد به کمک خاصیت `page.ViewContext.Controller` می‌توان به کنترلری که در حال رندر کردن `View` جاری است دسترسی یافت. اکنون که به کنترلر دسترسی داریم، به کمک `Reflection`, `Attributes` ویژگی‌ها یا `AuthorizeAttribute` است یا خیر. بر این اساس می‌توان تصمیم گرفت که آیا `View` در حال نمایش عمومی است یا خصوصی. اگر عمومی بود، کدهای اسکریپتی `Google analytics` به صورت خودکار به صفحه تزریق می‌شوند. همچنین در اینجا فرض بر این است که `Id` منتبه به دومین جاری در کلیدی به نام `GoogleAnalyticsID` در فایل کانفیگ برنامه در قسمت `app settings` آن تعریف شده است.

نظرات خوانندگان

نویسنده: سعید
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۱:۴

خیلی خوشحالم که مجددا شروع به نوشتمن کردید.
یه ایراد کوچیک درباره سایت جدید میخوام بگیرم. در سایت جدید دیگه نمیشه به راحتی به مطالب قبلی دسترسی داشت. در سایت قبل، گزینه‌های ارسال‌های قبلي و ارسال‌های جدیدتر وجود داشت که میشد به راحتی به ترتیب ارسال پست، سایت رو مرور کرد.
در صورتی که این امکان گداشته بشه، از مطالب بهتر میشه استفاده کرد

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۱:۱۰

ممnon. در این سایت هم هست. در پایین صفحه اصلی سایت سه لینک هست به نام‌های

[آخرین نظرات](#)
[آخرین مطالب ارسالی](#)
[آخرین اشتراک‌ها](#)

که همین کار را انجام می‌دهند.

نویسنده: Mohammad
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۱:۲۰

ازتون تشکر میکنم. فقط احساس میکنم یه باگی تو این سیستم جدیدتون مربوط به تعداد نظرات هستش. به عنوان مثال همین مطلب تو صفحه اصلی سایت ۰ نظر داره ولی داخل خود پست که میشی میبینی ۲ تا نظر داره. ولی بعد از چند دقیقه درست میشه. یا شایدم کش میشن.

نویسنده: Ara
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۱:۳۴

سلام

تبریک میگم وحید جان !
خیلی خوشحالم باز میتونیم از مطالب خوبتون استفاده کنیم !

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۲:۳۶

بله. از کش استفاده شده.

نویسنده: مجتبی حسینی
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۲:۵۸

سلام مهندس جان

از اینکه میتوونم اولین کار روزانه خودم را چک کردن سایت قرار بدم خیلی خوشحالم

ایکاش مانند گذشته فایل chm سایت رو نیز در اختیار مراجعه کنندگان برای استفاده آفلاین قرار میدادی

نکته دیگر آنکه تعداد برچسب‌های سایت خیلی کم است و این دسته بندی یافتن مطالب رو سخت میکنه

و دیگر آنکه چند کاربره شدن سایت دامنه مطالب ارسالی رو بازتر کرده و ممکنه اولاً سطح مطالب افت کنه و یا زیادی پراکنده بشه اگر برای نویسنده‌گان سایت راهبری هایی انجام بشه شاید مطالب هماهنگ‌تر و مفیدتر و مورد نیازتر بشن متشرکم

نویسنده: حمید نصیری
تاریخ: ۲۳:۳ ۱۳۹۱/۰۳/۲۹

سلام؛ ممنون. برچسب‌های سایت کم نیست. لیست کامل آن‌ها در اینجا قابل مشاهده است:

<http://www.dotnettips.info/search/label>

نویسنده: مومن
تاریخ: ۸:۵۳ ۱۳۹۱/۰۳/۳۰

مبارک است

هر روز بهتر از دیروز!

نویسنده: مهدی
تاریخ: ۱۲:۳۹ ۱۳۹۱/۰۳/۳۱

واقعاً عالی

عجب انژیکی آقا وحید شما...

نویسنده: مجتبی حسینی
تاریخ: ۲۱:۲ ۱۳۹۱/۰۳/۳۱

جناب مهندس عزیز

مثل اینکه چند روزی هست که google analytic نیز ایران رو تحریم کرده چاره چیه؟ جایگزین داره؟

نویسنده: حمید نصیری
تاریخ: ۲۱:۲۸ ۱۳۹۱/۰۳/۳۱

مثل بقیه سرویس‌های آن با V.P.N کار می‌کنه. شما فقط نیاز به یک tracking id دارید. ثبت سایت و دریافت آن کار جماعتیک تا دو دقیقه است.

نویسنده: آریا
تاریخ: ۱۲:۴ ۱۳۹۱/۰۴/۰۱

ممnon بابت مطلب خوبتون

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۸:۱۶ ۱۳۹۲/۰۵/۰۱

با سلام. امکان دارد در مورد نحوه استخراج اطلاعات از این سرویس و نمایش در سایت کمی راهنمایی کنید؟ باتشکر.

نوبنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۰۱ ۱۸:۲۶

[Google APIs Client Library for .NET](#)
[DotNET Client Library Developer's Guide](#)

نوبنده: ایاک
تاریخ: ۱۳۹۳/۰۲/۲۳ ۱۱:۱۹

با تشکر.

من دقیقاً روش بالا را استفاده کرده ام. در کلاینت خوب جواب می‌دهد ولی وقتی بر روی سرور آپلود می‌کنم، نتیجه‌ای حاصل نمی‌شود.

```
string username = ConfigurationManager.AppSettings["GoogleAnalyticsUsername"];
string password = ConfigurationManager.AppSettings["GoogleAnalyticsPassword"];
string profileId = ConfigurationManager.AppSettings["GoogleAnalyticsProfileId"];
string yesterdayDate = DateTime.Today.AddDays(-1).ToString("yyyy-MM-dd");
var service = new AnalyticsService("WebSiteAnalytics");

service.setUserCredentials(username, password);
var dataQuery = new DataQuery()
{
    Ids = profileId,
    Metrics = "ga:pageviews",
    Sort = "ga:pageviews",
};
DataFeed dataFeed = null;

//--- total visit
dataQuery.GAStartDate = new DateTime(2012, 3, 1).ToString("yyyy-MM-dd");
dataQuery.GAEndDate = DateTime.Now.ToString("yyyy-MM-dd");
dataFeed = service.Query(dataQuery);
var totalEntry = (dataFeed.Entries == null || dataFeed.Entries.Count == 0) ? "0" :
    (dataFeed.Entries[0] as DataEntry).Metrics[0].Value;

//--- yesterday visit
dataQuery.GAStartDate = DateTime.Now.AddDays(-1).ToString("yyyy-MM-dd");
dataQuery.GAEndDate = DateTime.Now.AddDays(-1).ToString("yyyy-MM-dd");
dataFeed = service.Query(dataQuery);
var yesterdayEntry = (dataFeed.Entries == null || dataFeed.Entries.Count == 0) ? "0" :
    (dataFeed.Entries[0] as DataEntry).Metrics[0].Value;

//--- today visit
dataQuery.GAStartDate = DateTime.Now.ToString("yyyy-MM-dd");
dataQuery.GAEndDate = DateTime.Now.ToString("yyyy-MM-dd");
dataFeed = service.Query(dataQuery);
var todayEntry = (dataFeed.Entries == null || dataFeed.Entries.Count == 0) ? "0" :
    (dataFeed.Entries[0] as DataEntry).Metrics[0].Value;

//--- last week visit
dataQuery.GAStartDate = DateTime.Now.AddDays(-7).ToString("yyyy-MM-dd");
dataQuery.GAEndDate = DateTime.Now.ToString("yyyy-MM-dd");
dataFeed = service.Query(dataQuery);
var lastWeekEntry = (dataFeed.Entries == null || dataFeed.Entries.Count == 0) ? "0" :
    (dataFeed.Entries[0] as DataEntry).Metrics[0].Value;

//--- last month visit
dataQuery.GAStartDate = DateTime.Now.AddDays(-30).ToString("yyyy-MM-dd");
dataQuery.GAEndDate = DateTime.Now.ToString("yyyy-MM-dd");
dataFeed = service.Query(dataQuery);
var lastMonthEntry = (dataFeed.Entries == null || dataFeed.Entries.Count == 0) ? "0" :
    (dataFeed.Entries[0] as DataEntry).Metrics[0].Value;
```

نوبنده: وحید نصیری

۱۱:۲۰ ۱۳۹۳/۰۲/۲۴

تاریخ:

- این روش منسوخ شده است و مربوط است به پروتکل های قدیمی گوگل.
- کار کردن با آن امن نیست چون باید اطلاعات حساس اکانت گوگل خود را در فایل کانفیگ برنامه قرار دهید.
- برای کار با این نوع پروتکل ها، در بسیاری از اوقات گوگل نیاز دارد تا شما توسط مرورگر سیستم جاری (در اینجا سرور وب)، یکباره به اکانت خود لاگین کنید.
- visits با ga:pageviews ارتباطی ندارد. فقط به این معنا است که صفحات سایت شما چندبار مشاهده شده اند و نه به معنای اینکه چند نفر آن هارا مشاهده کرده اند.

[روش بهتر در اینجا](#)

نویسنده: امیر
تاریخ: ۹:۳۱ ۱۳۹۳/۰۳/۰۷

با سلام و درود؛ یک سوال ؟ برای بدست آوردن کاربران Unique از session استفاده میکنیم و اما گوگل با استفاده از کدهای جاوا اسکریپت چطور میتوانه کاربران unique را بدست بیاره ؟!
و یک سوال دیگه اگه راهنمایی کنید ممنون میشم.
من یک آمارگیر برای سایتم نوشتم که یک مشکل با کاربرانی داره که با موبایل یا تبلت از سایت دیدن میکنند و با هر بار بازدید چند بار session اجرا میشه ! یعنی با یک بازدید به جای اینکه یک عدد به کاربران unique اضافه بشه چند عدد اضافه میشه ! که مشکل برای کاربرانی که از طریق ویندوز سایت را مشاهده میکنند وجود نداره.

نویسنده: وحید نصیری
تاریخ: ۹:۳۸ ۱۳۹۳/۰۳/۰۷

در Google Analytics (نه راه حل های شخصی):
تعداد بار مراجعه کاربران: «sessions»
کاربران منحصر بفرد مراجعه کننده به سایت: «users»
- [تفاوت بین ga:users و ga:sessions در Google Analytics](#)
- آنالیز محتوای کوکی ها utma و utmz + بررسی IP کاربر و تهیه یک چکیده از user-agent مرورگر و خیلی از مسایل ریز دیگر او برای تشخیص منحصر بفرد بودن کاربر و نه فقط اینکه سشنی شروع شده، بنابراین کاربر جدید است.

نویسنده: سینا س
تاریخ: ۱:۱۹ ۱۳۹۳/۱۲/۱۴

سلام؛ میخواستم بدونم در صورتیکه قصد داشته باشیم آنالیز ترافیک بازدید سایت رو بدون استفاده از سرویس های مختلف (PersianStat, Google Analytic)

وب سایت بر پایه 5 MVC ساخته شده. روشی که خودم در نظر دارم، یک جدول در بانک اطلاعاتی هست که شامل یک سری فیلد مانند IP و User-Agent و تاریخ و ساعت و ...

به ازای هر کاربر یک رکورد اضافه بشه در دیتابیس. ولی به صورتی باشه که فقط یکبار اضافه بشه در یک مدت زمان محدود. یعنی به ازای هر Request که از یک کاربر مشخص ارسال میشه نیاد پشته سر هم هی رکورد درج کنه و به قول معروف با Refresh کردن هی رکورد ثبت بشه توی دیتابیس.

در فایل Global.asax گزینه های خوبی هست برای اینکار ولی در وب سایت از Session_Start استفاده نمیکنم و غیرفعال ش کردم در وب کانفیگ، بنابراین جوابگو نیست.
ممنون میشم اگر بتونید راهنمایی کنید.

نویسنده: وحید نصیری

- از رخداد Application_BeginRequest باید استفاده کنید.
- همچنین اگر سرور متعلق به خودتان است، [روش لاغ‌های IIS](#) ، امکاناتی مشابه google analytics می‌دهد (_ و ^ و __).

در [قسمت اول](#) بحث‌های مقدماتی درباره وب زمان واقعی (real time web) و معرفی کتابخونه SignalR به همراه یک مثال ساده رو با هم دیدیم. در ادامه به جزئیات ریزی از کتابخونه SignalR که توسط آقای [David Fowler](#) توسعه داده میشود میپردازیم.

همونطور که قبل این اشاره شد قلب این کتابخونه در سمت سرور دو کلاس پایه PersistentConnection و Hub هستند که اولی سطحی پایینتر داره یعنی به تنظیمات و کدنویسی (بسیار) بیشتری برای پیاده‌سازی نیاز داره اما در عوض امکانات سطح پایینتری هم در اختیار برنامه نویس قرار میده که در برخی موارد موردنیاز هستن. در مورد بخش‌های مختلف این دو کلاس و نحوه پیاده‌سازی هر دو کلاس فوق، تو آدرسی که قبل اشاره کردم (آدرس پروژه متن باز این کتابخونه در [github](#)) راهنمایی‌های نسبتاً مفصلی ارائه شده و نیازی به تکرار این مطالب در اینجا نیست. پیشنهاد میکنم که این مطالب رو با دقت مطالعه کین.

به صورت توکار از 4 روشی که در قسمت قبل به اون اشاره شد برای برقراری ارتباط استفاده میکنند. که به بسترها جدیدی نیاز داره. Server-sent Events که تنها در مرورگرهایی که پشتیبانی کاملی از html5 دارند قابل استفاده است (بنابراین ie9 نمیتوانه از این روش استفاده کنه). forever frame (بخشی از استاندارد 1.1 (<http://http://www.w3.org/TR/2011/WD-html5-20110525/>)) داده‌ها رو به صورت chunked دریافت میکنند و روش آخر که Long-polling نام داره از هموش روش قدیمی Ajax (Ajax) بهره میبرند و با استفاده از آبجکت XMLHttpRequest کار ارتباط و تبادل داده‌ها رو انجام میدهند.

در اینجا برای بررسی این ارتباطات ابتدا برنامه چت ساده قسمت قبل رو در اینترنت اکسپلورر اجرا کنید و با استفاده developer tool (کلید F12) به درخواستهای مختلف ارسال شده به سرور نگاه کنید. پس از اجرای برنامه و قبل ارسال هرگونه داده به سرور در تب Network این ابزار چیزی شبیه به شکل زیر مشاهده خواهید کرد (البته باید قبل از ورود به صفحه برنامه چت روی دکمه Start capturing کلیک کنید):

File Find Disable View Images Cache Tools Validate Browser Mode: IE9 Document Mode: IE9 standards							
HTML	CSS	Console	Script	Profiler	Network		
URL	Method	Result	Type	Received	Taken	Initiator	
http://localhost:16869/index.htm	GET	304	text/html	335 B	15 ms		
/Scripts/jquery-1.6.4.min.js	GET	200	application/x-javascript	89.95 KB	172 ms		
/Scripts/jquery.signalR-0.5.1.min.js	GET	200	application/x-javascript	13.35 KB	63 ms		
/SimpleChat/negotiate?_=1340032079182	GET	200	application/json	0.54 KB	< 1 ms	JS Library XMLHttpRequest	
/SimpleChat/connect?transport=foreverFrame&connectionId=...	GET	(Pending...)	(Pending...)	0 B	(Pending...)		

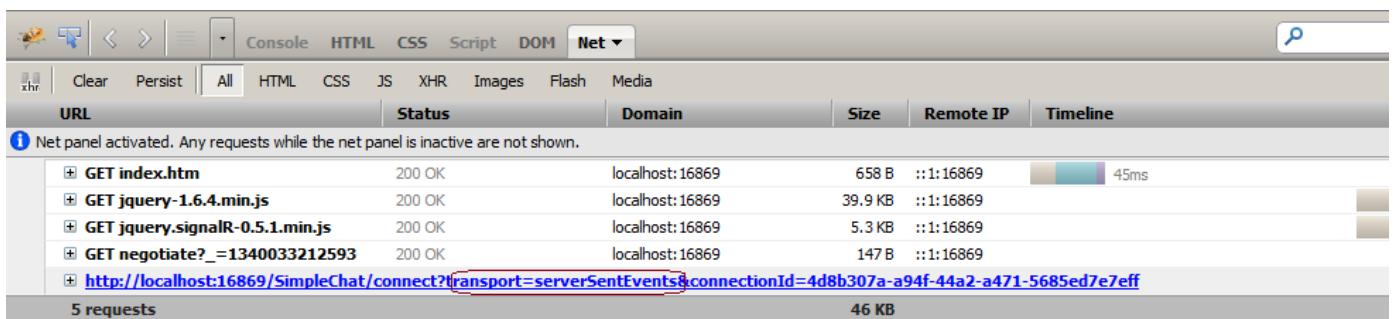
با توجه به تصویر بالا در شرایط موجود بهترین روش برای برقراری ارتباط با سرور رو `forever frame` تشخیص داده و مشاهده میشود که این ارتباط دائمی و فعلاً نتیجه‌ای از سمت سرور دریافت نکرده و ارتباط کاملاً زنده است. البته اگر در این ابزار درخواستهای ارسالی به سرور بیشتر جستجو بکنید اطلاعات بیشتری نصبیتون میشود که آوردنش اینجا بحث رو طولانی میکند.

حالا برنامه رو در یه مرورگر دیگه که از html5 پشتیبانی میکند اجرا کنید. مثلاً نتیجه در گوگل کروم و ابزار توسعه اون به شکل زیره:

	index.htm	GET	200 OK	text/html	Other	1.10KB 1.02KB	5ms 5ms	
	jquery-1.6.4.min.js /Scripts	GET	200 OK	application/x-javascript	index.htm:5 Parser	40.40KB 89.52KB	166ms 166ms	
	jquery.signalR-0.5.1.min.js /Scripts	GET	200 OK	application/x-javascript	index.htm:5 Parser	5.84KB 12.92KB	81ms 81ms	
	negotiate /SimpleChat	GET	200 OK	application/json	jquery-1.6.4.min.js:4 Script	557B 147B	12ms 2ms	
	connect /SimpleChat	GET	200 OK	text/event-stream	jquery.signalR-0.5.1.min.js Script	477B 29B	33.62s 13ms	
<pre>http://localhost:16869/SimpleChat/ connect?transport=serverSentEvents&connectionId=921a39 46-b983-4b3a-8d23-96aae5e7f53e</pre>								

همونطور که میبینیم در اینجا روش استفاده شده Server Sent Events هست.

در فایرفاکس هم با استفاده از ابزار محبوب firebug نتیجه مشابه کروم بدست میاد:



البته اگر علاقه زیادی به کندوکاو در جزئیات این درخواستها دارین (مثل خود من) چیزی بهتر از [fiddler2](#) بیدا نمیشه. میتوనیں پس از ارسال یک متن دوباره این درخواستها رو مورد بررسی قرار بدین و ببینیں که چیجوری کانالهای ارتباط پس از ارسال و دریافت دیتا قطع و برقرار میشه.

این نکته رو هم باید یادآور بشم که هرچند که این کتابخونه بهترین روش رو میتونه انتخاب کنه اما به برنامه نویس امکان تعیین صریح روش ارتباط رو هم میده. اگر به راهنمای این کتابخونه سر برزنین میبینین که امکانات زیادی بهش اضافه شده و امکانات زیادی هم در آینده به اون اضافه میشه. امکاناتی از قبیل ارسال دادهها به یک کلاینت خاص و یا به گروهی خاص از کلاینتها، خصوصی‌سازی آدرس سرور و همچنین پشتیبانی از Cross Domain در آخرین نسخه، امکان استفاده از [Reactive Extension](#) (بلایگ)، [Self Hosting](#) که امکان خیلی جالبیه و میتوونه خیلی جاها به عنوان یک راه حل سبک و سریع به کار بیاد، قابلیت فوق العاده در باینینگ دادهها در سمت سرور و مخصوصاً کلاینت، امکان تشخیص برقراری یا قطع ارتباط کلاینتها در سمت سرور، استفاده از امکانات این کتابخونه برای برقراری ارتباط با کلاینتها در خارج از فضای کلاس‌های مشتق شده از دو کلاس پایه ([Hub](#) و [PersistentConnection](#)) و چند مورد دیگه تا نسخه جاری اضافه شدند.

در حال حاضر دارم روی یه برنامه چت با امکانات بیشتر کار میکنم که پس از آماده شدن ارائه میدمیش. یکی از پروژه‌های میتوانی من بازی که با استفاده از این کتابخونه توسعه داده شده [jabbr.net](#) است. یه اتاق گفتگوی کامل با امکانات جالبه که میتوانیم به اون هم یه سری برزنین.

در آخر هم یه لینک جالب برای مطالعه معرفی میکنم: [Highest voted SignalR Questions - stackoverflow](#)

در ASP.NET MVC به کمک یک سری فیلتر می‌توان مشخص کرد که یک اکشن متدها به درخواست‌هایی از نوع `Get` پاسخ دهد، دیگری به درخواست‌هایی از نوع `Post` و الی آخر. عادت متد اول من هم برای نمایش معمولی صفحات، استفاده از حالت `HttpGet` است که در شبکه‌های داخلی بدون مشکل کار می‌کند چون `Bot` ای در این شبکه‌ها وجود ندارد و اگر باشد احتمالاً یک ویروس است!

```
[HttpGet]  
public ActionResult Index()  
{  
    return View();  
}
```

اما روی اینترنت وضع فرق می‌کند. برای مثال یک کنترلر `Feed` ممکن است درخواست‌هایی از نوع `Head` نیز داشته باشد و یا بعضی از `Bot`‌ها بجای `Get` جهت دریافت اطلاعات فید، از `Post` هم استفاده می‌کنند! (مانند فیدبزرن گوگل مطابق لاغ‌های برنامه!) و در این حالت اگر متدهای `Get` را به `HttpPost` تنظیم کرده باشیم، مدام با خطای `A public action method xyz was not found on controller abc` در لاغ‌های خطای برنامه مواجه خواهیم شد.
بنابراین بهتر است که در سایتهاي عمومي، جهت ارائه اطلاعات و صفحات عمومي، از هیچ نوع فیلتری برای محدود کردن یک اکشن متدها کارش صرفاً رندر کردن اطلاعات است، استفاده نکنیم. به این ترتیب اگر کسی توسط درخواستی از نوع `Head` فقط قصد بررسی تغییرات صورت گرفته را داشت و نه دریافت کل فید، بی‌جهت آن را محدود نکرده باشیم.

در **قسمت قبل** درباره روش‌های برقراری ارتباط با سرور در کتابخونه SignalR کمی بحث شد. برای ادامه بهتره که به برنامه چت ساده‌ای که تو این مدت کمی تکمیلش کردم به نگاهی بندازین: [SimpleChat.rar](#)

لطف کنین این برنامه رو دانلود و اجرا کنین تا کمی با جزئیات این کتابخونه بیشتر آشنا بشین. این برنامه قدم به قدم نوشته شده و حاوی نسخه‌های مختلفی از برنامه چت هست که هر کدام تو یه فایل html استفاده شده. نسخه آخر شامل عملیات لاین، چت گروهی، چت خصوصی و امکان تعییر گروه است. در ضمن این برنامه کمی با عجله نوشته شده پس اگه باگ یا موردی مشاهده کردین و یا پیشنهادی دارین اشاره کنین تا بقیه هم استفاده کنن.

حالا به یه نکته در مورد آغاز برقراری ارتباط کلاینت با سرور اشاره میکنم. قبل از برقراری این ارتباط (که در قسمت قبل توضیحاتی در این مورد داده شده) برنامه کلاینت یک درخواست به سمت سرور ارسال میکنه. به تصویر زیر دقت کنین:

Key	Value
Request	GET /signalr/negotiate?_=1340358748157 HTTP/1.1
X-Requested-With	XMLHttpRequest
Accept	application/json, text/javascript, */*; q=0.01
Referer	http://localhost:16869/Htmls/index7.htm
Accept-Language	en-us
Accept-Encoding	gzip, deflate
User-Agent	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; MAM3)
Host	localhost:16869
Connection	Keep-Alive
Cookie	DXCurrentThemeASPxGridView=RedWine; .ASPXANONYMOUS=fWlYyT9_zQEkAAAAAYTAyMzIyZmUtNTczMS00MTJjLTg5Y2ItNWU4ODRjMjjhNzUy0

به این درخواست اولیه در کتابخونه SignalR همونطور که مشاهده میشه مذاکره (negotiate) گفته میشه. برقراری ارتباط این درخواست اولیه هم توسط XHR انجام میشه. نتیجه این مذاکره در تب Response Body قابل مشاهده است:

```

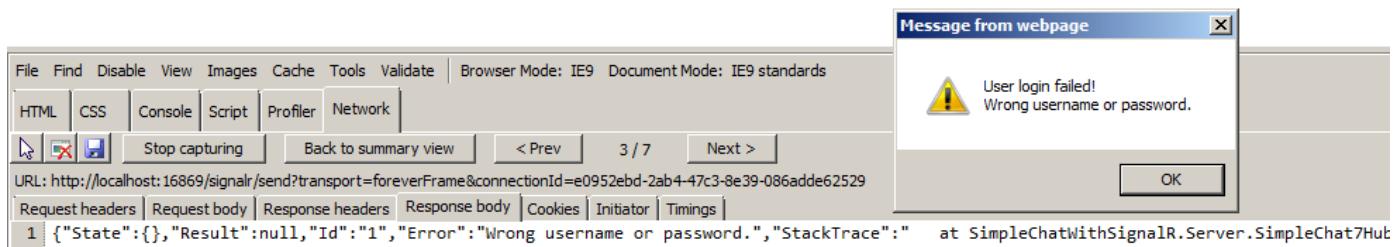
1 {"Url":"/signalr","ConnectionId":"633f036b-b52c-4445-8ac6-2219a962b78c","TryWebSockets":false,"WebSocketServerUrl":null,"ProtocolVersion":"1.0"}

```

میبینین که علاوه بر آی دی ارتباط که یک guid است (تلفظ مرسومش «گوئید» هستش) امکان برقراری ارتباط از طریق روش WebSocket رو هم از طرف سرور مشخص میکنه که با توجه به استفاده من از ویندوز 7 امکانش وجود نداره. یعنی اگر مثلاً شما از ویندوز 8 و 8 IIS استفاده کنین مقدار true برابر TryWebSocket پارامتر همچنین پارامتر WebSocketServerUrl نال نخواهد بود. البته این پارامتر تنها مربوط به سروره و برنامه کلاینت مورد استفاده (در اینجا یک مرورگر) هم باید توانایی استفاده از این روش را داشته باشه. پس از اتمام این مذاکره ارتباط اصلی برقرار میشه.

یکی از قابلیت‌های خوب این کتابخونه ارسال خطاهای رخ داده در سمت سرور به کلاینت هست. در تصویر زیر بدنه یک نمونه از

پاسخهای سرور که نمایش دهنده خطای رخ داده در سمت سرور هست رو نشون داده شده:



برای راحتی دوستان در استفاده از راهنمای این کتابخونه، یه مقدار کار رو شون انجام دادم و با خلاصه کردن محتوای اونا (و کاهش حجم 95 درصدی!) برای استفاده آفلاین آماده کردم:

[SignalR github docs.rar](#)

اگه فرصتی پیش بیاد و دوستان هم علاقه داشته باشن در قسمت بعدی برنامه چت رو بیشتر با هم بررسی میکنیم.

در ادامه قصد دارم تا روی بازدهی و کارایی این کتابخونه تو بار زیاد یه بررسی هایی انجام بدم (البته اگه وقت کنم چون راه اندازی یه محیط تست برای این جور کتابخونه ها چندان آسون نیست) و مطلبی هم در مورد نحوه راه اندازی تست بار ارائه بدم.

نظرات خوانندگان

نویسنده: علی
تاریخ: ۱۳۹۱/۰۴/۰۲ ۱۷:۱۰

مرسی آقای یوسف نژاد، امیدوارم این بحثتون ادامه پیدا کنه

نویسنده: باربد
تاریخ: ۱۳۹۱/۰۴/۰۳ ۸:۱۱

خیلی عالی بود ...
راستش من هم تو فاصله این دو تا مقاله باهش کار کردم.
چیز خیلی فوق العاده ایه . البته اگر لطف کنید و دو تا ساختار اصلی این مبحث را که Hubs و Persistent Connection را به تفکیک و با جزئیات توضیح بدید ، خیلی خیلی عالی خواهد بود. (فقط یه سوال؟ تو نسخه بعد دات نت ، جزء ساختار اصلی فریم ورک خواهد بود؟)
باز هم متشرکم و موفق باشید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۳ ۹:۱۶

ASP.NET 4.5 در WebSockets Protocol پشتیبانی می شود: (^)

نویسنده: باربد
تاریخ: ۱۳۹۱/۰۴/۰۳ ۱۰:۳۵

سپاس (:)

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳۹۱/۰۴/۰۳ ۱۵:۵۹

سلام ... عالی بود ... خیلی استفاده بردم ... برنامه ای که نوشتینم خیلی مفیده ;) ... به نظرم اگه در مورد برنامه ای که نوشتین تو قسمت های بعدی سرمایه گذاری کنین خیلی خوب میشه

نویسنده: بنیامین
تاریخ: ۱۳۹۱/۱۰/۱۱ ۱۸:۸

با سلام و خسته نباشید
واقعاً ممنون از مطالعه که گذاشتین. خیلی از مشکلات من رو حل کرد. خواهش می کنم بحث رو ادامه بدید.
با تشکر

نویسنده: MSN
تاریخ: ۱۳۹۱/۱۰/۲۲ ۱۷:۲۸

خیلی خوب و کاربردی ، واقعاً خسته نباشید.

سئوال اینجاست ، زمانی که پیامی رو ارسال میکنیم دستور زیر رو اجرا میکنیم که این پیام واسه بقیه اعضاء هم ارسال بشه

```
public void SendMessage(string message, string id)
{
    Clients.receiveMessage(message, Users[id]);
}
```

زمانی که مثلاً توی یک اتاق 10 نفر در یک زمان پیامی رو ارسال میکنند، واسه همه اعضا 10 تا Response جدا میاد؟ اگه اینطور هست چطور میشه این مورد رو بهینه کرد؟

نویسنده: میثم ق
تاریخ: ۱۳۹۲/۰۲/۱۵ ۱:۱۱

سلام

خیلی خوب بود. واقعاً دستت درد نکنه

امیدوارم این بحث و ادامه بدین

خیلی ممنون

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۱۵ ۹:۵۸

جهت اطلاع؛ دوره [SignalR سایت](#) این مباحثت رو بر اساس نگارش جدیدترش تکمیل کردد.

نویسنده: یوسف نژاد
تاریخ: ۱۳۹۲/۰۲/۱۵ ۱۶:۵۷

با توجه به ریلیز شدن نسخه نهایی این کتابخونه با عنوان ASP.NET SignalR و تغییرات بوجود اومده، در حال تهیه مطلبی جدید و نیز بروزرسانی برنامه چت ارائه شده و تهیه چند برنامه جدید مفید دیگه هستم که امیدوارم به زودی آماده بشن.

چند وقتی میشه که دنبال روش‌های [OpenID](#) هستم که بینم چطوری کار می‌کنند، خودم هم تازه شروع کردم خوب قبل از هر چیزی اول بینم مفهوم [OpenID](#) چی هست؟ و کم کم جلو میریم و مثال‌هایی معرفی می‌کنیم.

به شما اجازه می‌دهد با استفاده از اکانت (نام کاربری) که در یک سایت دارید بتوانید به سایت‌های متفاوتی وارد شوید (لاگین کنید) بدون این که نیاز به ثبت نام دوباره در آن سایت‌ها داشته باشید.

نمونه بارز آن می‌توان به سایت‌های مانند [StackOverflow](#) اشاره کرد.

به شما اجازه می‌دهد شما در یک نام کاربری که برای خود ایجاد کرده اید اطلاعاتی را که دارید با دیگر وبسایت‌ها به اشتراک بگذارید.

با [OpenID](#) کلمه عبور شما فقط توسط سرویس دهنده گرفته می‌شود و سرویس دهنده هویت شما را برای بازید از سایت دیگر تایید می‌کند. از طرف دیگر سرویس دهنده شما تا شما اجازه ندهید هیچ وب سایتی کلمه عبور شما را به هیچ وب سایتی نمی‌بینند. بنابراین نیازی نیست در مورد کلمه عبور خود نگرانی به خود راه دهید.

به سرعت در حال گسترش بروی وب استف در حال حاضر بیش از یک میلیارد نام کاربری (اکانت) وجود دارد و بیش از 50000 سایت [OpenID](#) را پذیرفته و با آن کار می‌کنند. چندین سازمان بزرگ موضوع [OpenID](#) را پذیرفته اند، سازمان‌های مانند Google, FaceBook, Yahoo!, Microsoft, AOL, MySpace, Sears, Universal Music Group, France Telecom, Novell, Sun, Telecom Italia و بسیاری از سازمان‌های دیگر.

در کل می‌توان گفت ما در یک سایت خاص مانند یاهو، گوگل، ... یک نام کاربری خواهیم داشت سپس برای ارتباط سایت مقصد (مثلاً همین سایت) با نام کاربری ما در گوگل به این صورت عمل می‌شود که ابتدا از طریق این سایت ما به سایت گوگل هدایت می‌شویم در آنجا از ما یک تاییدیه جهت استفاده از سرویس [OpenID](#) از کاربر می‌گیرد. در صورت تایید کاربر سایت گوگل از این لحظه جهت احراز هویت کاربر برای ورود به سایت مقصد استفاده می‌کند.

زمانی که کاربر می‌خواهد به این سایت لاگین کند سایت نام کاربری و کلمه عبور او را (در صورتی که قبلاً به گوگل لاگین کرده باشد نیازی نیست و وارد سایت می‌شود) به گوگل می‌فرستد و پس از تایید هویت در صورت صحیح بودن اجازه می‌دهد کاربر به آسانی وارد سایت مقصد شود. تصویر زیر نمایانگر این روش می‌باشد.

در پست‌های آینده مثال‌ها و کامپوننت‌های سورس بازی جهت این کار به شما معرفی خواهیم کرد. جهت مطالعات بیشتر می‌توانید به این لینک‌ها مراجعه کنید ([^](#) و [^](#) و [^](#)).).

نظرات خوانندگان

نویسنده: ashi mashی
تاریخ: ۱۳۹۱/۰۴/۱۸ ۱۱:۳۵

سلام،

این چیزی که شما میگید ، آیا مشابه single sign on هستش ؟

همون کار رو انجام میده ؟

آیا با Active directory قابل اتصال هست یا نه ؟

با تشکر....

نویسنده: صابر فتح الله
تاریخ: ۱۳۹۱/۰۴/۱۸ ۱۱:۳۸

فکر نمیکنم بشه

چون Active Directory با Domain تنظیم میشه

اما این حالت احراز هویت روی فرم تنظیم میشه

برای SSO مطلب توی همین سایت هست در واقع توی اون روش ما با یک اکانت که توی یک سایت داریم از سایت‌های دیگه روی

همون سرویس استفاده میکنیم و میشه گفت حالت اختصاصی داره

اما توی این حالت شما با هر سایتی که این قابلیت داشته باشه میتونی کار کنی

نویسنده: Hamid NCH
تاریخ: ۱۳۹۲/۰۹/۱۴ ۱۲:۳۱

نقشه شروع یادگیری من در برنامه نویسی وب، پروژه رایگان [شما](#) جناب آقای استاد صابر فتح الهی بود.

بعد از اون هم که با سایت آقای دلشاد آشنا شدم و اینک سایت فوق العاده پربار جاری به همت تمام عزیزان.

و باز هم الان دارم از شما آقای فتح الهی چیز یاد میگیرم.

در حین جستجوی مطلبی در فوروم‌های ASP.NET مطلبی رو از یکی از اعضای تیم ASP.NET [دیدم](#) که خلاصه‌اش به این شرح است:

اگر چندین درخواست با یک Session ID به موتور پردازشی ASP.NET وارد شوند و حداقل یکی از آن‌ها به صورت سشنی خواندنی+نوشتگی علامت گذاری شده باشد، اینگونه درخواست‌ها serialized می‌شود. به این معنا که این نوع درخواست‌ها در صف قرار داده شده و به ترتیب رسیدن، اجرا می‌شوند. این کار هم از این جهت لازم است که چندین درخواست یا به عبارتی چندین ترد همزمان، سبب انجام تغییرات ناخواسته‌ای در مقادیر سشن مورد استفاده نشوند.

و اگر چندین درخواست با یک Session ID به موتور پردازشی ASP.NET وارد شوند و تمام آن‌ها سشن «فقط خواندنی» باشند یا در آن‌ها سشن «غیرفعال» شده باشد، این درخواست‌ها به صورت موازی پردازش خواهند شد.

بنابراین اگر در برنامه خود از سشن استفاده نمی‌کنید، سطر زیر را به وب کانفیگ برنامه اضافه نمایید تا حداکثر استفاده را از پردازش موازی خودکار کرده باشید:

```
<httpModules>
  <remove name="Session"/>
</httpModules>
```

```
<sessionState mode="Off" />
```

و یا در ASP.NET MVC می‌توان کنترل ریزتری را به نحو زیر نیز اعمال کرد:

```
[SessionState(SessionStateBehavior.Disabled)]
public class AjaxTestController : Controller
{
    //...
}
```

در اینجا تنها حالت سشن مرتبط با اکشن متدهای یک کنترلر (و یا تمام اکشن‌های یک کنترلر) که برای درخواست‌های Ajax طراحی شده، خاموش گردیده است تا درخواست‌های غیرهمزمان Ajax ای بتوانند حداکثر استفاده را از پردازش موازی خودکار کنند. نکته دیگر اینکه اگر نیاز به استفاده از سشن در یک اکشن متدهای داشته باشد می‌توان از فیلتر سشن‌های فقط خواندنی استفاده کرد تا باز هم به همان پردازش موازی پشت صحنه برسیم:

```
[SessionState(SessionStateBehavior.ReadOnly)]
```

نظرات خوانندگان

نویسنده: بهاره قدمی
تاریخ: ۱۳۹۱/۰۴/۰۳ ۱۱:۵۰

با سلام و ممنون از سایت خیلی خوب شما سوالم اینه که آیا اگر بخوایم در یک برنامه asp.net از سشن استفاده کنیم و ان را در کل سایت readonly اعلام کنیم تا بتونیم پردازش موازی داشته باشیم راهی هست که در وب کانفیگ این تنظیم را انجام دهیم؟ یا حتی هنگام تعریف اولیه سشن پس از مقدار دهی آنرا readonly کنیم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۳ ۱۲:۳

در وب کانفیگ خیر. تعریف‌های انجام شده در وب کانفیگ سراسری است. به این معنا که اگر می‌شد سشن را در وب کانفیگ readonly کرد، بعد کجا باید در آن مقداری را درج می‌کردیم؟ بنابراین استفاده از سشن غیرممکن می‌شد؛ چون برنامه اجازه‌ی درج هیچ مقداری را در هیچ نوع سشنی نمی‌داد.

جهت تکمیل بحث:
در وب فرم‌ها هم امکان readonly کردن سشن مورد استفاده در یک صفحه وجود دارد :

```
pagesSection.EnableSessionState = PagesEnableSessionState.ReadOnly
```

```
<% @Page EnableSessionState="ReadOnly" %>
```

نویسنده: reza
تاریخ: ۱۳۹۱/۰۴/۰۳ ۱۵:۵۳

آقای نصیری آیا استفاده از سشن در زمانی که کاربران زیاد هستند بار زیاد روی سرور ایجاد نمی‌کنند. تا چه تعداد کاربر را می‌شه با سشن ساپورت کرد. مثلا همون اطلاعات یوزری که لاگین کرده؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۳ ۱۶:۱۴

سؤال شما خارج از بحث پردازش موازی است. سشن به صورت پیش فرض در حافظه سرور ذخیره می‌شود بنابراین حد و حدود آن مشخص است. البته سشن را در ASP.NET [می‌شود در SQL Server هم](#) ذخیره کرد. کمی کندتر از حافظه است اما مشکل مقیاس پذیری آنچنانی نداره.

نویسنده: omid
تاریخ: ۱۳۹۱/۰۸/۲۳ ۲۰:۴

سلام منظورتون از چندین درخواست یا یک sessionId یکسان به موتور asp.net برسند رو میشه توضیح بدید؟ آیا منظورتون از Session همون Session[""] هست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۲۳ ۲۲:۴۶

توضیحات بیشتر در مورد SessionId ([^](#)) :

نویسنده: علی
تاریخ: ۱۳۹۱/۱۰/۲۸ ۲۳:۵

سلام

من `UserId` کاربری که به سایت لاگین کرده رو در `session` نگه می دارم تا برای کارهای مختلف هر بار به دیتابیس کوئری نزنم. آیا روش دیگری هست که بجای سشن بشه استفاده کرد (خصوصاً در MVC)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۲۸ ۲۳:۲۰

بله. از [Forms authentication](#) استفاده کنید.

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

بعد `userId` رو به شکل فوق تنظیم کنید. از این به بعد با مراجعه به شیء `User` به صورت `User.Identity.Name` به مقدار `userId` خواهد رسید.

نویسنده: علی
تاریخ: ۱۳۹۱/۱۰/۲۸ ۲۳:۳۰

ممnon، آیا اطلاعات دیگری را هم می‌شود با این روش ذخیره کرد، مثل نام کاربر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۲۸ ۲۳:۳۸

کاری که در اینجا انجام می‌شود ذخیره اطلاعات به صورت رمزنگاری شده در یک کوکی است. رمزنگاری، ذخیره و مدیریت این کوکی (`AuthCookie`) خودکار است. در سایر موارد خودتون این مراحل مدیریت کوکی‌ها رو با کدنویسی شبیه سازی کنید.

نویسنده: علی
تاریخ: ۱۳۹۱/۱۰/۲۸ ۲۳:۳۹

ممnon بابت راهنمایی خوبتون

نویسنده: رضا گرمارودی
تاریخ: ۱۳۹۲/۱۱/۱۹ ۱۱:۱۰

سلام من واقعاً متوجه نحوه استفاده یا عدم استفاده از `Session`ها نشدم!
در Asp.net برای نگهداری یکسری از اطلاعات کاربر و یا هرگونه اطلاعات مربوط به یک کاربر خاص از `Session`ها استفاده می‌کردیم.
حال در Mvc برای نگهداری هر گونه اطلاعات کاربر `Login` شده از چی باید استفاده کرد. اگر جواب کوکی هاست این کوکی‌ها در سمت کلاینت ثبت می‌شود و نیاز است که اطلاعات در سمت سرور باشد.
کدم بهتره است: بخشی از حافظه برای اطلاعات کاربر جاری مصرف شود و در عوض در هر درخواستی از آن استفاده شود و یا خیر هر بار این اطلاعات واکشی شود (البته بدون درنظر گرفتن سطح دوم کش).
همچنین `Session`ها در Asp.net با Mvc می‌توانند استفاده کردند. این می‌شود این اطلاعات را بر طرف بفرمایید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۹ ۱۱:۳۰

- در سایت جاری ماژول سشن در وب کانفیگ کلا حذف شده. در هیچ جایی از آن از سشن استفاده نشده. (یک نمونه عملی)

- به کوکی سمت کلاینت در سمت سرور دسترسی وجود دارد. مانند سیستم [Forms Authentication](#) که اساس آن استفاده از کوکی‌های رمزگاری شده‌است؛ چه برای نگهداری اطلاعات کاربر لاگین شده به سیستم و چه برای نگهداری اطلاعات نقش‌های منتبه به او.
- تفاوتی ندارند. یک هسته‌ی اصلی ASP.NET وجود دارد که حداقل دو فریم‌ورک وب فرمز و MVC از آن استفاده می‌کنند.

نویسنده: همراه
تاریخ: ۱۳۹۳/۰۹/۱۱ ۱۱:۰

سلام؛ اگر بخواهیم از سشن استفاده نکنیم، برای نگهداری اطلاعات کسی که لاگین کرده چه راهی وجود دارد؟ مثلاً UserId یا LoginId که با اون بشه به UserId رسید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۱۱ ۱۱:۲۴

- در سایر نظرات مطلب جاری پیشتر بحث شده در این مورد.
- هیچکدام از روش‌های مدرن authentication در ASP.NET از سشن استفاده نمی‌کنند. برای مثال از [forms authentication](#) و یا [asp.net identity](#) استفاده کنید.

نویسنده: ناظم
تاریخ: ۱۳۹۴/۰۳/۲۰ ۱۴:۴۲

در 7 iis و بالاتر ظاهرا باید به این شکل عمل کرد:

```
<system.webServer>
  <modules>
    <remove name="Session"/>
  </modules>
</system.webServer>
```

یک سری نکته ریز را جهت بهبود ظاهر برنامه‌های وب می‌توان درنظر داشت؛ برای مثال: مجموعه‌ی Twitter Bootstrap که به عنوان یکی از فریم‌ورک‌های خوب CSS مطرح است، دارای تعدادی دکمه تهیه شده با است: (^)

برای نمایش یک چنین دکمه‌هایی فقط کافی است یک span را به صفحه اضافه کرده و آنرا مثلاً مساوی btn btn-info قرار دهیم تا دکمه‌ای آبی رنگ نمایش داده شود.



طراحی زیبایی دارد. با مرورگرهای جدید سازگار است و ... در اصل یک span بیشتر نیست و قابلیت post back به سرور را ندارد.

برای اضافه کردن چنین قابلیتی می‌توان از نکته زیر استفاده کرد:

```
<span style=" margin:7px;" onclick="#loginForm').submit()">ورود به سیستم</span>
```

jQuery که جزء پیش فرض برنامه‌های ASP.NET MVC است. متدهای submit را هم می‌توان در اینجا در صورت کلیک بر روی یک span فراخوانی کرد. نکته مهم آن ذکر ID فرم جاری است که مثلاً به این شکل قابل تعریف است:

```
@using (Html.BeginForm(actionName: "LogOn", controllerName: "Login",
                      method: FormMethod.Post, htmlAttributes: new { id = "loginForm" }))
```

نظرات خوانندگان

نویسنده: احمد
تاریخ: ۱۳۹۱/۰۴/۰۸ ۱۱:۱۶

بسیار عالی!

وقتی CSS رو به صفحه اضافه میکنیم روی تمام کنترلها اثر میداره. که مطلوب نیست. راه حل چیه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۸ ۱۱:۳۰

ضرورتی به استفاده از تمام پکیج آن نیست. مثلا می‌شود فقط از فایل CSS مربوط به دکمه‌های آن استفاده کرد تا بر روی سایر کنترل‌ها تاثیر گذار نباشد. مانند این فایل:

[buttons.css](#)

کار متد کمکی Url.Action یک Url بر اساس تعاریف مسیریابی برنامه است. البته متد کمکی مشابهی نیز مانند `Html.ActionLink` در MVC وجود دارد که کار آن تولید یک لینک قابل کلیک است؛ اما ممکن است در حالت خاص تنها نیاز به خود Url داشته باشیم و نه لینک قابل کلیک آن.

الف) اگر از `jQuery Ajax` استفاده می‌کنید، حتما باید استفاده از `Url.Action` را لحاظ کنید

برای نمونه اگر قسمتی از عملیات Ajax آیی برنامه شما به نحو زیر تعریف شده است :

```
$ajax({
    type: "POST",
    url: "/Home/EmployeeInfo",
    ...
})
```

... غلط است!

در این حالت برنامه شما تنها در زمانیکه در ریشه یک دومین قرار گرفته باشد کار خواهد کرد. اگر برنامه شما در مسیری مانند `http://www.site.com/myNewApp` نصب شود، کلیه فراخوانی‌های Ajax آیی آن دیگر کار نخواهند کرد چون مسیر url فوق به ریشه سایت اشاره می‌کند و نه مسیر جاری برنامه شما (در یک sub domain جدید).

به همین جهت در یک چنین حالتی حتما باید به کمک `Url.Action` مسیر یک اکشن متد را معرفی کرد تا به صورت خودکار بر اساس محل قرارگیری برنامه و تعاریف مسیریابی آن، url صحیحی تولید شود.

```
@Url.Action("EmployeeInfo", "Home")
```

ب) دریافت Url مطلق از یک `Url.Action`

Url‌های تولید شده توسط `Url.Action` به صورت پیش فرض نسبی هستند (نسبت به محل نصب و قرارگیری برنامه تعریف می‌شوند). اگر نیاز به دریافت یک مسیر مطلق که با `http` برای مثال شروع می‌شود دارید، باید به نحو زیر عمل کرد:

```
@Url.Action("About", "Home", null, "http")
```

پارامتر چهارم تعریف شده، سبب خواهد شد لینک تولیدی از حالت نسبی خارج شود.

ج) استفاده از `Url.Action` در یک کنترلر

فرض کنید قصد تولید یک فید RSS را در کنترلری دارید. یکی از آیتم‌هایی که باید ارائه دهید، لینک به مطلب مورد نظر است. این لینک باید مطلق باشد همچنین در یک View هم قرار نداریم که به کمک `@` بلافاصله به متد کمکی `Url.Action` دسترسی پیدا کنیم. در کنترلرهای وله جاری کلاس به شیء Url و متد Action آن به نحو زیر دسترسی دارد و خروجی نهایی آن یک رشته است:

```
var url = this.Url.Action(actionName: "Index",
                           controllerName: "Post",
                           protocol: "http",
                           routeValues: new { id = item.Id });
```

بنابراین در کنترلرهای زیر Url‌ها را دستی تولید نکنید و اجازه دهید بر اساس پارامترهای زیادی که در پشت صحنه `Url.Action`

لحاظ می‌شود، Url صحیحی برای شما تولید گردد.

د) استفاده از Url.Action در کلاس‌های کمکی برنامه خارج از یک کنترلر

فرض کنید قصد تهیه یک Html Helper سفارشی را به کمک کدنویسی در یک کلاس مجزا دارید. در اینجا نیز نباید Urlها را دستی تولید کرد. در UrlHelper سفارشی نیز می‌توان به کمک متدهای UrlHelper.GenerateUrl به همان امکانات دسترسی یافت:

```
public static class Extensions
{
    public static string MyLink(this HtmlHelper html, ...)
    {
        string url = UrlHelper.GenerateUrl(null, "actionName", "controllerName",
                                            null,
                                            html.RouteCollection,
                                            html.ViewContext.RequestContext,
                                            includeImplicitMvcValues: true);
    }
}
```

نظرات خوانندگان

نوبسند: محسن
تاریخ: ۱۳۹۱/۰۴/۱۰ ۱۵:۶

سلام و خسته نباشید خدمت شما آقای نصیری. و بلاگتون واقعاً مفید و عالیه
ممnon از زحماتتون بابت مطالبی که به اشتراک میگذارید
یه سوال از خدمتون داشتم

من روی یه پروژه MVC کار میکنم. به این صورت کار کردم که یک View کلی دارم که توی اون نمایش اطلاعات و "ایجاد" و "حذف"
و "ویرایش" همه یکجا انجام میشن. این View را کاملاً با Jquery Insert , Update , Delete انجام میشیم. اما توی یکی از View های دیگه که Strongly Type هم هستش، نمیخام به طور کامل از Jquery استفاده کنم. به این صورت
که من اطلاعات جدول مربوطه رو توسط Jquery از پایگاه داده میگیرم و در یک گرید نمایش میدم. توی این گرید برای هر ردیف
دوتا لینک "حذف" و "ویرایش" وجود داره. برای حذف هم با Jquery کارم و انجام میدم. اما برای "ایجاد" یک فرم از نوع فرم های
MVC دارم که داخل یه dialog از نوع Jquery قرارشون دادم. برای ایجاد هم مشکلی نیست. اما برای ویرایش، نمیدونم که چطوری
باید اطلاعات رو از پایگاه داده لود کنم که خودش مستقیماً داخل TextBox ها قرار بگیره. البته توسط Jquery او مدم تک تک
برنامه رو و استون بفرستم

```
function editmode(val) {
    $.ajax({ url: "/User>SelectUser", data: { Username: val }, type: "post", dataType: "json",
    success: function (data) {
        if (data != "timeout") {
            if (data.isRedirect) { window.location.href = data.redirectUrl; return; }
            try {
                $("#dvEdit").dialog({ modal: 'true', title: 'ویرایش', hide: 'clip' });
                //alert($("#dvEdit").html());

                $("#UsernameEdit").val(data.Username);
                $("#FirstNameEdit").val(data.FirstName);
                $("#LastNameEdit").val(data.LastName);
                $("#NationalCodeEdit").val(data.NationalCode);
                $("#EmailEdit").val(data.Email);
                $("#PhoneNoEdit").val(data.PhoneNo);
                $("#MobileNoEdit").val(data.MobileNo);
                $("#CreationDateEdit").val(data.CreationDate);
                $("#CreationDateEdit").prop('disabled', true);
                $("#LastActivityDateEdit").val(data.LastActivityDate);
                $("#LastActivityDateEdit").prop('disabled', true);
                $("#LastLoginDateEdit").val(data.LastLoginPage);
                $("#LastLoginDateEdit").prop('disabled', true);
                $("#IsLockedOutEdit").val(data.IsLockedOut);
                $("#AddressEdit").val(data.Address);

            }
            catch (err) {
                $("#dverr").show(); $("#lblErr").html(err);
            }
        }
        else
            AjaxTimeout();
    }
    error: function (req, textStatus, errorThrown) { AjaxError(req, textStatus, "#dverr",
    "#lblErr"); }
    complete: function (xhr, e) { AjaxComplete(xhr, "#dverr", "#lblErr"); }
  });
}
```

نوبسند: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۰ ۱۶:۵۹

فرم ادیت رو به صورت strongly typed partial view از نوع یک partial view مستقل درست کنید.
سیس در کنترلر مرتبط، یک اکشن متده را مخصوص رندر کردن این partial view در نظر بگیرید. کار آن دریافت اطلاعات مرتبط

با Model ارسالی به همین partial view است. سپس در آخر کار آن هم خواهیم داشت:

```
return PartialView("_MyPartialViewName", data);
```

حالا فراخوانی این اکشن متد توسط `jQuery Ajax` سبب پر شدن خودکار فیلدهای فرم `strongly typed` شما هم می‌شود. در حین درخواست، اطلاعات مدل از بانک اطلاعاتی دریافت شده و به `Partial view` ارسال می‌شود. چون `strongly typed` است، فیلدهای آن به صورت خودکار پر شده و نهایتاً کل `partial view` به صورت یک رشته آماده شده در اختیار شما خواهد بود. بنابراین، قسمت عمده‌ای از کدهای سمت کاربر فوق کاهش پیدا می‌کند. چون یک `view` کامل حاضر و آماده از سرور دریافت شده است که باید به صفحه توسط `jQuery` اضافه شود.

+

لطفاً اینجا رو تبدیل به یک انجمن عمومی نکنید. عنوان بحث ساخت `Url` بود ... بعد تغییر جهت پیدا کرد به یک عنوان دیگر.
با تشکر از توجه شما.

نویسنده: Afshin Akbari
تاریخ: ۲۰۱۳۹۱/۰۵/۱۸

سلام جناب نصیری ممنون از مطالب مفیدتون.
سوال:

همون طور که مرسوم هست معمولاً کدهای `jquery` رو در داخل فایل مجزا قرار میدن که در این صورت دیگه `mvc` وجود نداره که بخوایم از `@url` استفاده کنیم میخوام بدونم برای این مشکل راه حلی وجود دارد.

نویسنده: وحید نصیری
تاریخ: ۲۰۱۳۹۱/۰۵/۱۷

در View مورد نظر یک `section` تعریف کنید به نحو زیر (که در آن می‌شود متغیرها را تعریف کرد):

```
@Section scriptVariables {
    <script type="text/javascript">
        var variableA = '@MyVarA';
    </script>
}
```

سپس در فایل `layout` سایت، پیش از الحاق اسکریپت اصلی آن را فراخوانی کنید:

```
<head>
    <meta charset="utf-8" />
    @RenderSection("scriptVariables", false)

    <script src="@Url.Content("~/Scripts/custom.js")" type="text/javascript"></script>
</head>
```

نویسنده: محسن
تاریخ: ۲۶۰۸/۱۳۹۲/۱۳:۵۱

سلام؛ من توی پروژم 2 تا Area دارم. یک هلپر برای نمایش گالری هم دارم که توی اون هلپر از `Url.Action` استفاده می‌کنم. جالب اینجاست که این هلپر توی یک Area مسیر اکشن رو درست بر می‌گردونه ولی برای Area دومی که همون هلپر رو استفاده می‌کنم اسم اکشن (که `Index` هست) رو بر نمی‌گردونه. اینه:

```
Url.Action("Index", "ShowGallery", null, Request.Url.Scheme)
```

نویسنده: وحید نصیری
تاریخ: ۱۴:۲۹ ۱۳۹۲/۰۸/۲۶

نام Area را باید ذکر کنید

```
@Url.Action("Action", "MyController", new { area = "OtherArea" })
```

و یا بهتر است از [T4MVC](#) استفاده کنید تا این مسایل را به صورت خودکار لحاظ کند.

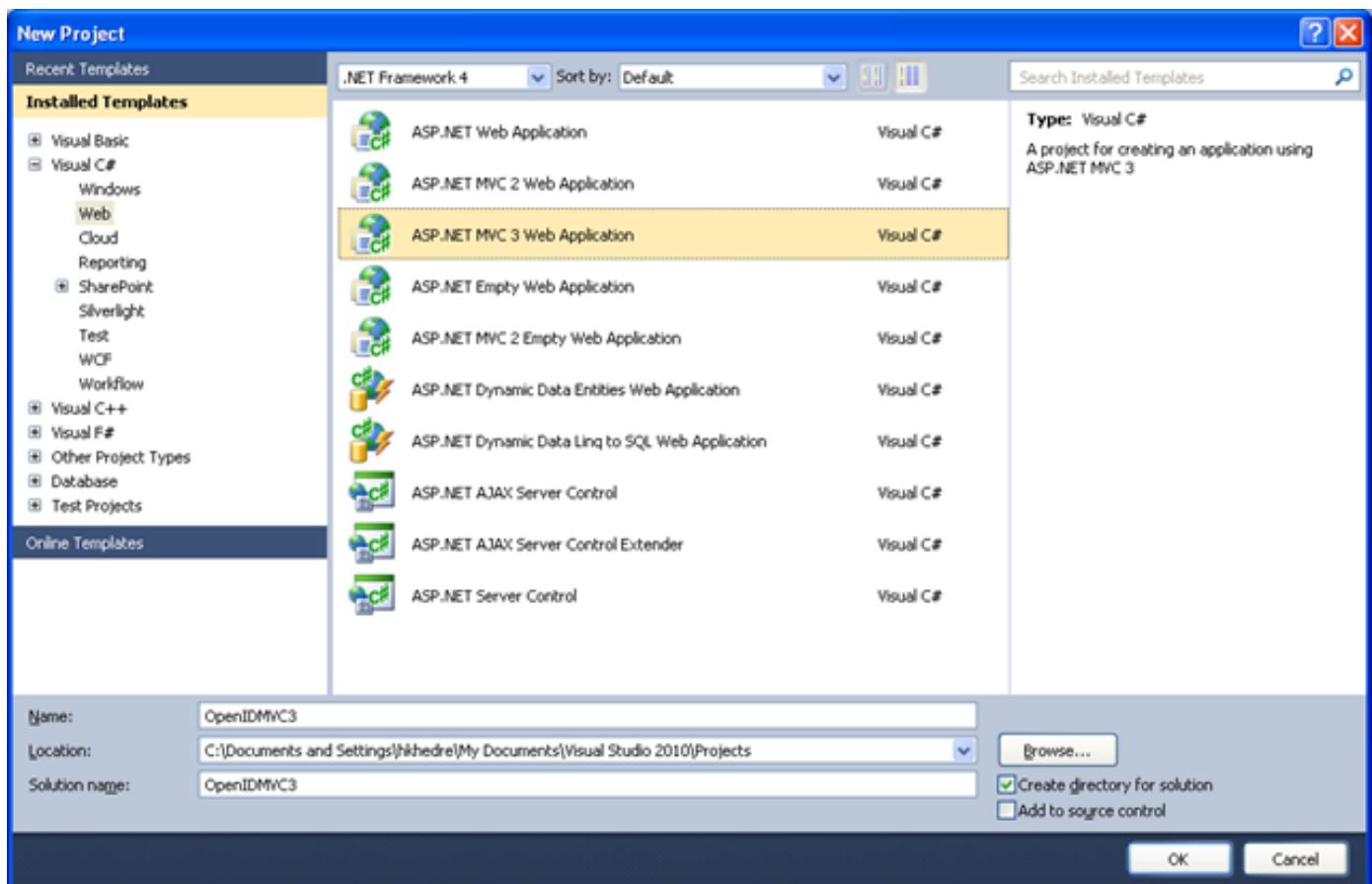
نویسنده: وحید نصیری
تاریخ: ۲۰:۱ ۱۳۹۴/۰۳/۰۲

روشی دیگر برای تولید Url در کدهای کنترلر برنامه

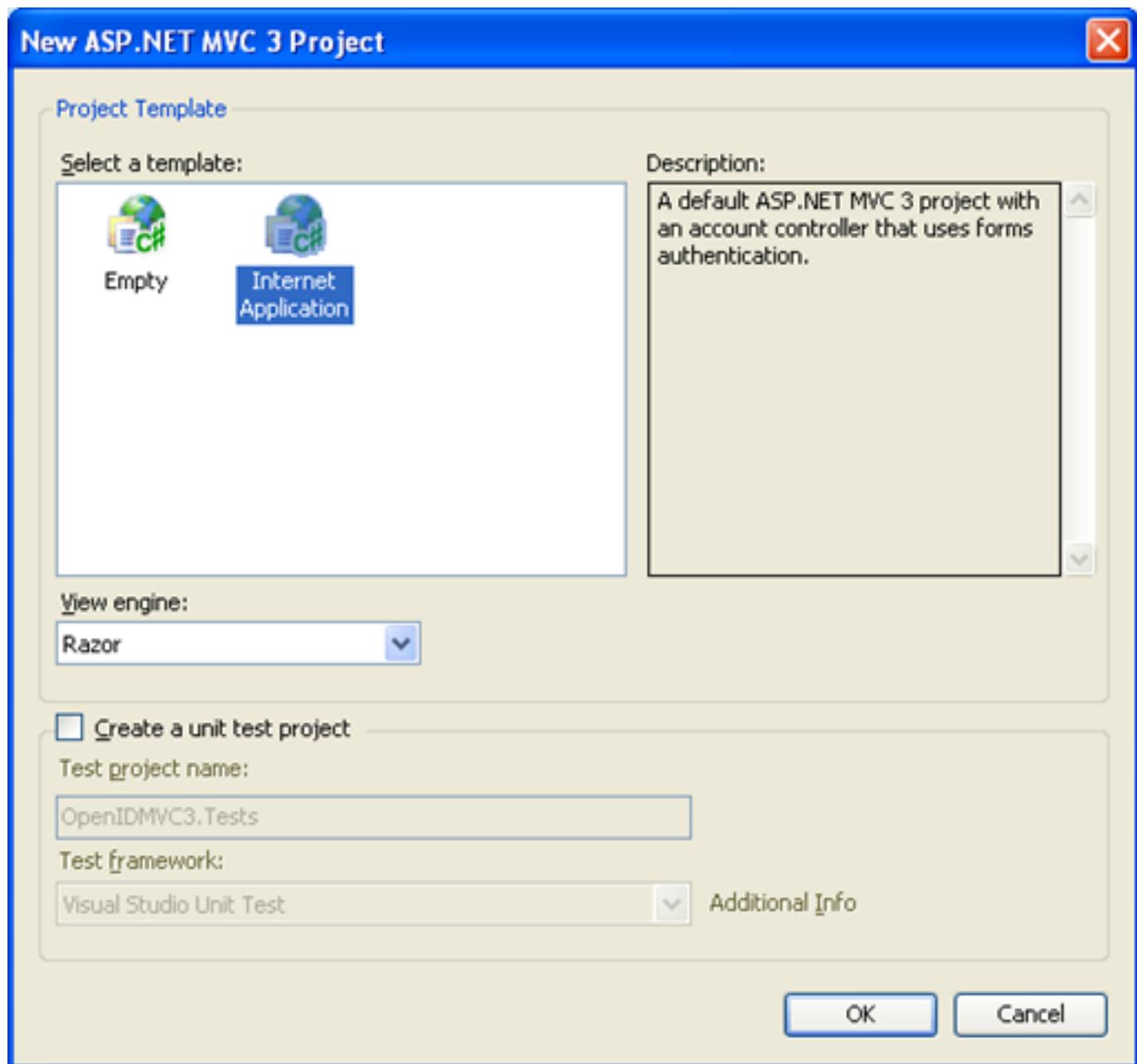
```
new UrlHelper(this.Request.RequestContext).Action("Index", "Home", new { area = "ArticlesArea" })
```

```
// for images  
new UrlHelper(this.Request.RequestContext).Content("~/content/images/file.png")
```

قبل از شرح مختصری در زمینه OpenID در [اینجا](#) گفته شد.
 حال می‌خواهیم این امکان را در پروژه خود بکار ببریم، جهت این کار باید ابتدا یک پروژه ایجاد کرده و از کتابخانه‌های سورس باز موجود استفاده کرد.
 ۱- ابتدا در ویژوال استودیو یا هر نرم افزار دیگر یک پروژه MVC ایجاد نمایید.



۲- نوع Internet Application و برای View Engine سایت Razor را انتخاب نمایید.



3- کتابخانه DotNetOpenId سورس باز را می‌توانید مستقیماً از این [آدرس](#) دانلود نموده یا از طریق [Package Manager Console](#) و با نوشتن Install-Package DotNetOpenAuth به صورت آنلاین این کتابخانه را نصب نمایید.

4- مدل‌های برنامه را مانند زیر ایجاد نمایید

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Globalization;
using System.Security.Cryptography;
using System.Text;
using System.Web.Mvc;
using System.Web.Security;

namespace OpenIDExample.Models
{
    #region Models

    public class ChangePasswordModel
    {
        [Required]
```

```

[DataType(DataType.Password)]
[Display(Name = "Current password")]
public string OldPassword { get; set; }

[Required]
[ValidatePasswordLength]
[DataType(DataType.Password)]
[Display(Name = "New password")]
public string NewPassword { get; set; }

[DataType(DataType.Password)]
[Display(Name = "Confirm new password")]
[Compare("NewPassword", ErrorMessage = "The new password and confirmation password do not
match.")]
public string ConfirmPassword { get; set; }
}

public class LogOnModel
{
    [Display(Name = "OpenID")]
    public string OpenID { get; set; }

    [Required]
    [Display(Name = "User name")]
    public string UserName { get; set; }

    [Required]
    [DataType(DataType.Password)]
    [Display(Name = "Password")]
    public string Password { get; set; }

    [Display(Name = "Remember me?")]
    public bool RememberMe { get; set; }
}

public class RegisterModel
{
    [Display(Name = "OpenID")]
    public string OpenID { get; set; }

    [Required]
    [Display(Name = "User name")]
    public string UserName { get; set; }

    [Required]
    [DataType(DataType.EmailAddress)]
    [Display(Name = "Email address")]
    public string Email { get; set; }

    [Required]
    [ValidatePasswordLength]
    [DataType(DataType.Password)]
    [Display(Name = "Password")]
    public string Password { get; set; }

    [DataType(DataType.Password)]
    [Display(Name = "Confirm password")]
    [Compare("Password", ErrorMessage = "The password and confirmation password do not match.")]
    public string ConfirmPassword { get; set; }
}

#endregion Models

#region Services

// The FormsAuthentication type is sealed and contains static members, so it is difficult to
// unit test code that calls its members. The interface and helper class below demonstrate
// how to create an abstract wrapper around such a type in order to make the AccountController
// code unit testable.

public interface IMembershipService
{
    int MinPasswordLength { get; }

    bool ValidateUser(string userName, string password);

    MembershipCreateStatus CreateUser(string userName, string password, string email, string
OpenID);

    bool ChangePassword(string userName, string oldPassword, string newPassword);
}

```

```

        MembershipUser GetUser(string OpenID);
    }

    public class AccountMembershipService : IMembershipService
    {
        private readonly MembershipProvider _provider;

        public AccountMembershipService()
            : this(null)
        {
        }

        public AccountMembershipService(MembershipProvider provider)
        {
            _provider = provider ?? Membership.Provider;
        }

        public int MinPasswordLength
        {
            get
            {
                return _provider.MinRequiredPasswordLength;
            }
        }

        public bool ValidateUser(string userName, string password)
        {
            if (String.IsNullOrEmpty(userName)) throw new ArgumentException("Value cannot be null or
empty.", "userName");
            if (String.IsNullOrEmpty(password)) throw new ArgumentException("Value cannot be null or
empty.", "password");

            return _provider.ValidateUser(userName, password);
        }

        public Guid StringToGUID(string value)
        {
            // Create a new instance of the MD5CryptoServiceProvider object.
            MD5 md5Hasher = MD5.Create();
            // Convert the input string to a byte array and compute the hash.
            byte[] data = md5Hasher.ComputeHash(Encoding.Default.GetBytes(value));
            return new Guid(data);
        }

        public MembershipCreateStatus CreateUser(string userName, string password, string email, string
OpenID)
        {
            if (String.IsNullOrEmpty(userName)) throw new ArgumentException("Value cannot be null or
empty.", "userName");
            if (String.IsNullOrEmpty(password)) throw new ArgumentException("Value cannot be null or
empty.", "password");
            if (String.IsNullOrEmpty(email)) throw new ArgumentException("Value cannot be null or
empty.", "email");

            MembershipCreateStatus status;
            _provider.CreateUser(userName, password, email, null, null, true, StringToGUID(OpenID), out
status);
            return status;
        }

        public MembershipUser GetUser(string OpenID)
        {
            return _provider.GetUser(StringToGUID(OpenID), true);
        }

        public bool ChangePassword(string userName, string oldPassword, string newPassword)
        {
            if (String.IsNullOrEmpty(userName)) throw new ArgumentException("Value cannot be null or
empty.", "userName");
            if (String.IsNullOrEmpty(oldPassword)) throw new ArgumentException("Value cannot be null or
empty.", "oldPassword");
            if (String.IsNullOrEmpty(newPassword)) throw new ArgumentException("Value cannot be null or
empty.", "newPassword");

            // The underlying ChangePassword() will throw an exception rather
            // than return false in certain failure scenarios.
            try
            {
                MembershipUser currentUser = _provider.GetUser(userName, true /* userIsOnline */);
                return currentUser.ChangePassword(oldPassword, newPassword);
            }
        }
    }
}

```

```

        catch (ArgumentException)
        {
            return false;
        }
        catch (MembershipPasswordException)
        {
            return false;
        }
    }

    public MembershipCreateStatus CreateUser(string userName, string password, string email)
    {
        throw new NotImplementedException();
    }
}

public interface IFormsAuthenticationService
{
    void SignIn(string userName, bool createPersistentCookie);
    void SignOut();
}

public class FormsAuthenticationService : IFormsAuthenticationService
{
    public void SignIn(string userName, bool createPersistentCookie)
    {
        if (String.IsNullOrEmpty(userName)) throw new ArgumentException("Value cannot be null or
empty.", "userName");

        FormsAuthentication.SetAuthCookie(userName, createPersistentCookie);
    }

    public void SignOut()
    {
        FormsAuthentication.SignOut();
    }
}

#endregion Services

#region Validation

public static class AccountValidation
{
    public static string ErrorCodeToString(MembershipCreateStatus createStatus)
    {
        // See http://go.microsoft.com/fwlink/?LinkId=177550 for
        // a full list of status codes.
        switch (createStatus)
        {
            case MembershipCreateStatus.DuplicateUserName:
                return "Username already exists. Please enter a different user name.";

            case MembershipCreateStatus.DuplicateEmail:
                return "A username for that e-mail address already exists. Please enter a different
e-mail address.";

            case MembershipCreateStatus.InvalidPassword:
                return "The password provided is invalid. Please enter a valid password value.";

            case MembershipCreateStatus.InvalidEmail:
                return "The e-mail address provided is invalid. Please check the value and try
again.';

            case MembershipCreateStatus.InvalidAnswer:
                return "The password retrieval answer provided is invalid. Please check the value
and try again.';

            case MembershipCreateStatus.InvalidQuestion:
                return "The password retrieval question provided is invalid. Please check the value
and try again.';

            case MembershipCreateStatus.InvalidUserName:
                return "The user name provided is invalid. Please check the value and try again.';

            case MembershipCreateStatus.ProviderError:
                return "The authentication provider returned an error. Please verify your entry and
try again. If the problem persists, please contact your system administrator.';

            case MembershipCreateStatus.UserRejected:

```

```

        return "The user creation request has been canceled. Please verify your entry and
try again. If the problem persists, please contact your system administrator.";

    default:
        return "An unknown error occurred. Please verify your entry and try again. If the
problem persists, please contact your system administrator.";
    }
}

[AttributeUsage(AttributeTargets.Field | AttributeTargets.Property, AllowMultiple = false,
Inherited = true)]
public sealed class ValidatePasswordLengthAttribute : ValidationAttribute, IClientValidatable
{
    private const string _defaultErrorMessage = "'{0}' must be at least {1} characters long.";
    private readonly int _minCharacters = Membership.Provider.MinRequiredPasswordLength;

    public ValidatePasswordLengthAttribute()
        : base(_defaultErrorMessage)
    {
    }

    public override string FormatErrorMessage(string name)
    {
        return String.Format(CultureInfo.CurrentCulture, ErrorMessageString,
            name, _minCharacters);
    }

    public override bool IsValid(object value)
    {
        string valueAsString = value as string;
        return (valueAsString != null && valueAsString.Length >= _minCharacters);
    }

    public IEnumerable<ModelClientValidationRule> GetClientValidationRules(ModelMetadata metadata,
ControllerContext context)
    {
        return new[]{
            new
ModelClientValidationStringLengthRule(FormatErrorMessage(metadata.GetDisplayName()), _minCharacters,
int.MaxValue)
        };
    }
}

#endregion Validation
}

```

5- در پروژه مربوطه یک Controller به نام AccountController ایجاد نمایید. و کدهای زیر را برای آنها وارد نمایید.

```

using System.Web.Mvc;
using System.Web.Routing;
using System.Web.Security;
using DotNetOpenAuth.Messaging;
using DotNetOpenAuth.OpenId;
using DotNetOpenAuth.OpenId.RelyingParty;
using OpenIDExample.Models;

namespace OpenIDExample.Controllers
{
    public class AccountController : Controller
    {
        private static OpenIdRelyingParty openid = new OpenIdRelyingParty();

        public IFormsAuthenticationService FormsService { get; set; }

        public IMembershipService MembershipService { get; set; }

        protected override void Initialize(RequestContext requestContext)
        {
            if (FormsService == null) { FormsService = new FormsAuthenticationService(); }
            if (MembershipService == null) { MembershipService = new AccountMembershipService(); }

            base.Initialize(requestContext);
        }

        // ****
        // URL: /Account/LogOn
    }
}

```

```

// ****
public ActionResult LogOn()
{
    return View();
}

[HttpPost]
public ActionResult LogOn(LogOnModel model, string returnUrl)
{
    if (ModelState.IsValid)
    {
        if (MembershipService.ValidateUser(model.UserName, model.Password))
        {
            FormsService.SignIn(model.UserName, model.RememberMe);
            if (Url.IsLocalUrl(returnUrl))
            {
                return Redirect(returnUrl);
            }
            else
            {
                return RedirectToAction("Index", "Home");
            }
        }
        else
        {
            ModelState.AddModelError("", "The user name or password provided is incorrect.");
        }
    }

    // If we got this far, something failed, redisplay form
    return View(model);
}

// ****
// URL: /Account/LogOff
// ****

public ActionResult LogOff()
{
    FormsService.SignOut();

    return RedirectToAction("Index", "Home");
}

// ****
// URL: /Account/Register
// ****

public ActionResult Register(string OpenID)
{
    ViewBag.PasswordLength = MembershipService.MinPasswordLength;
    ViewBag.OpenID = OpenID;
    return View();
}

[HttpPost]
public ActionResult Register(RegisterModel model)
{
    if (ModelState.IsValid)
    {
        // Attempt to register the user
        MembershipCreateStatus createStatus = MembershipService.CreateUser(model.UserName,
model.Password, model.Email, model.OpenID);

        if (createStatus == MembershipCreateStatus.Success)
        {
            FormsService.SignIn(model.UserName, false /* createPersistentCookie */);
            return RedirectToAction("Index", "Home");
        }
        else
        {
            ModelState.AddModelError("", AccountValidation.ErrorCodeToString(createStatus));
        }
    }

    // If we got this far, something failed, redisplay form
    ViewBag.PasswordLength = MembershipService.MinPasswordLength;
    return View(model);
}

```

```

// ****
// URL: /Account/ChangePassword
// ****

[Authorize]
public ActionResult ChangePassword()
{
    ViewBag.PasswordLength = MembershipService.MinPasswordLength;
    return View();
}

[Authorize]
[HttpPost]
public ActionResult ChangePassword(ChangePasswordModel model)
{
    if (ModelState.IsValid)
    {
        if (MembershipService.ChangePassword(User.Identity.Name, model.OldPassword,
model.NewPassword))
        {
            return RedirectToAction("ChangePasswordSuccess");
        }
        else
        {
            ModelState.AddModelError("", "The current password is incorrect or the new password
is invalid.");
        }
    }

    // If we got this far, something failed, redisplay form
    ViewBag.PasswordLength = MembershipService.MinPasswordLength;
    return View(model);
}

// ****
// URL: /Account/ChangePasswordSuccess
// ****

public ActionResult ChangePasswordSuccess()
{
    return View();
}

[ValidateInput(false)]
public ActionResult Authenticate(string returnUrl)
{
    var response = openid.GetResponse();
    if (response == null)
    {
        //Let us submit the request to OpenID provider
        Identifier id;
        if (Identifier.TryParse(Request.Form["openid_identifier"], out id))
        {
            try
            {
                var request = openid.CreateRequest(Request.Form["openid_identifier"]);
                return request.RedirectingResponse.AsActionResult();
            }
            catch (ProtocolException ex)
            {
                ViewBag.Message = ex.Message;
                return View("LogOn");
            }
        }
        ViewBag.Message = "Invalid identifier";
        return View("LogOn");
    }

    //Let us check the response
    switch (response.Status)
    {
        case AuthenticationStatus.Authenticated:
            LogOnModel lm = new LogOnModel();
            lm.OpenID = response.ClaimedIdentifier;
            //check if user exist
            MembershipUser user = MembershipService.GetUser(lm.OpenID);
            if (user != null)
            {
                lm.UserName = user.UserName;
                FormsService.SignIn(user.UserName, false);
            }
    }
}

```

```
        }

        return View("LogOn", lm);

    case AuthenticationStatus.Canceled:
        ViewBag.Message = "Canceled at provider";
        return View("LogOn");
    case AuthenticationStatus.Failed:
        ViewBag.Message = response.Exception.Message;
        return View("LogOn");
    }

    return new EmptyResult();
}

}
```

6- سپس برای Action به نام LogOn یک View می‌سازیم، برای Authenticate نیازی به ایجاد View ندارد چون قرار است درخواست کاربر را به آدرس دیگری Redirect کند. سپس کدهای زیر را برای View ایجاد شده وارد می‌کنیم.

```

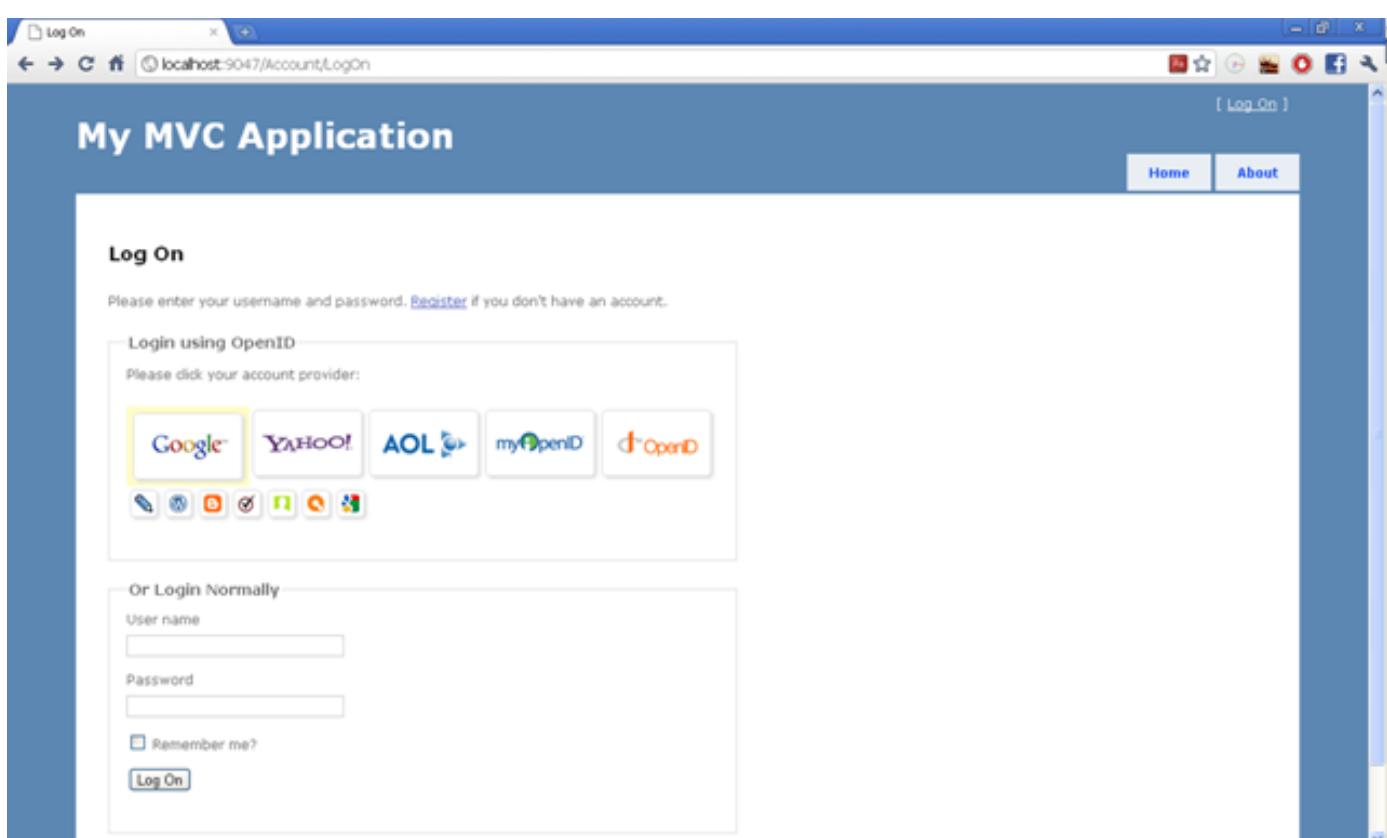
@model OpenIDExample.Models.LogOnModel
 @{
     ViewBag.Title = "Log On";
 }
<h2>
    Log On</h2>
<p>
    Please enter your username and password. @Html.ActionLink("Register", "Register")
    if you don't have an account.
</p>
<script src="@Url.Content("~/Scripts/jquery.validate.min.js)" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")"
type="text/javascript"></script>
<form action="Authenticate?ReturnUrl=@HttpUtility.UrlEncode(Request.QueryString["ReturnUrl"])"
method="post" id="openid_form">
<input type="hidden" name="action" value="verify" />
<div>
    <fieldset>
        <legend>Login using OpenID</legend>
        <div class="openid_choice">
            <p>
                Please click your account provider:</p>
            <div id="openid_btns">
            </div>
        </div>
        <div id="openid_input_area">
            @Html.TextBox("openid_identifier")
            <input type="submit" value="Log On" />
        </div>
        <noscript>
            <p>
                OpenID is service that allows you to log-on to many different websites using a single
                identity. Find out <a href="http://openid.net/what/">more about OpenID</a> and
                <a href="http://openid.net/get/">how to get an OpenID enabled account</a>.</p>
        </noscript>
    <div>
        @if (Model != null)
        {
            if (String.IsNullOrEmpty(Model.UserName))
            {
                <div class="editor-label">
                    @Html.LabelFor(model => model.OpenID)
                </div>
                <div class="editor-field">
                    @Html.DisplayFor(model => model.OpenID)
                </div>
                <p class="button">
                    @Html.ActionLink("New User", "Register", "Register", new { OpenID = Model.OpenID })
                </p>
            }
            else
            {
                //user exist
                <p class="buttonGreen">
                    <a href="@Url.Action("Index", "Home")">Welcome , @Model.UserName, Continue...</a>
                </p>
            }
        }
    </div>
</form>

```

استفاده از OpenID در وب سایت جهت احراز هویت کاربران

```
        }
    </div>
</fieldset>
</div>
</form>
@Html.ValidationSummary(true, "Login was unsuccessful. Please correct the errors and try again.")
@using (Html.BeginForm())
{
    <div>
        <fieldset>
            <legend>Or Login Normally</legend>
            <div class="editor-label">
                @Html.LabelFor(m => m.UserName)
            </div>
            <div class="editor-field">
                @Html.TextBoxFor(m => m.UserName)
                @Html.ValidationMessageFor(m => m.UserName)
            </div>
            <div class="editor-label">
                @Html.LabelFor(m => m.Password)
            </div>
            <div class="editor-field">
                @Html.PasswordFor(m => m.Password)
                @Html.ValidationMessageFor(m => m.Password)
            </div>
            <div class="editor-label">
                @Html.CheckBoxFor(m => m.RememberMe)
                @Html.LabelFor(m => m.RememberMe)
            </div>
            <p>
                <input type="submit" value="Log On" />
            </p>
        </fieldset>
    </div>
}
```

پس از اجرای پروژه صفحه ای شبیه به پایین مشاهده کرده و سرویس دهنده OpenID خاص خود را می‌توانید انتخاب نمایید.



7- برای فعال سازی عملیات احراز هویت توسط FormsAuthentication در سایت باید تنظیمات زیر را در فایل web.config انجام دهید.

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

خوب تا اینجا کار تمام است و کاربر در صورتی که در سایت OpenID نام کاربری داشته باشد می‌تواند در سایت شما Login کند. جهت مطالعات بیشتر و دانلود نمونه کدهای آماده می‌توانید به لینک‌های ([^](#) و [^](#) و [^](#) و [^](#) و [^](#)) مراجعه کنید. کد کامل پروژه را می‌توانید از [اینجا](#) دانلود نمایید.

[منبع](#)

نظرات خوانندگان

نویسنده: ahmadalli
تاریخ: ۲۳:۱۲ ۱۳۹۱/۰۴/۰۸

خوب این کدهای شما برای نسخه‌های قدیمی MVC هست. من نصفش رو درست متوجه نشدم. اگر میشه برای نسخه ۴ یا ۳ هم مثال بزارید.

نویسنده: امیرحسین جلوداری
تاریخ: ۱:۳۵ ۱۳۹۱/۰۴/۰۹

اگه میشه کد برنامه را ضمیمه کنید...

نویسنده: صابر فتح الله
تاریخ: ۲:۴ ۱۳۹۱/۰۴/۰۹

کدی که ابتدا گذاشته بودم ظاهرا خیلی قدیمی بود اون با کدهای جدید تعویض کردم و لینک دانلود کد هم برای دوستان قرار دادم

نویسنده: صابر فتح الله
تاریخ: ۲:۲۱ ۱۳۹۱/۰۴/۰۹

اصلاح شد

نویسنده: saleh
تاریخ: ۰:۳۳ ۱۳۹۱/۰۴/۱۷

فرضاً اگه شخصی که با OID لوگین کرده بخواهد در فروم سایت من فعالیت کنه چه جوری به اطلاعات پستها و اعمالش پی ببرم؟
چون این شخص در سایت ثبت نام نکرده و طبیعتاً اکانتی در سایت نداره!

در قسمت اول مقاله به این موضوعات اشاره نکردید.

نویسنده: وحید نصیری
تاریخ: ۱۱:۴۴ ۱۳۹۱/۰۴/۱۷

بحث فوروم نوشته شده شما با طراحی یک سیستم جدید (که در اینجا مد نظر است) متفاوت است. سورس فوق را مطالعه کنید.
تا با نحوه یکپارچگی آن با سیستم membership دات نت آشنا شوید.

نویسنده: احمدعلی شفیعی
تاریخ: ۱۲:۳۸ ۱۳۹۱/۰۴/۱۸

ممnonm

نویسنده: کامران
تاریخ: ۲۳:۲۱ ۱۳۹۲/۰۲/۱۲

سلام دوست عزیز.

قربان برای ASP.NET Webforms هم میتوانید مقاله ای رو آماده کنید؟

استفاده از OpenID در وب سایت جهت احراز هویت کاربران

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۱۳ ۰:۴

یک پروژه دیگر هم در سایت در مورد پروتکل OAuth هست.

نویسنده: کامران
تاریخ: ۱۳۹۲/۰۲/۱۳ ۰:۹

ممnon از جوابتون، دوست عزیز این هم سمبیل و کدهاش با MVC هست، مقالاتی برای Webforms پیدا کردم به زبان لاتین اما درست توضیح داده نشده متأسفانه

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۲/۱۳ ۰:۲۵

سلام

دوست گلم یه مثال هست گذاشتم توی این آدرس

[DotNetOpenAuth با استفاده از ASP.NET در Google OpenID Authentication](#)

در مواردی مانند همین وبلاگ احتیاج به داشتن محل نوشتن، در محیط ادیتور مانندی می‌باشد که برای این مورد می‌توان از ادیتور استفاده کرد. بعد از دانلود محتویات مربوطه (که در اینجا از `d11` مربوط به استفاده نشده بلکه فایل‌های ادیتور [CKEditor](#) مستقیماً مورد استفاده قرار گرفته است)، کافیست لینک‌های مربوط به جاوا اسکریپت و استایل را به هدر اضافه کرد:

```
<link href="admin/editor/contents.css" rel="stylesheet" type="text/css" />
<script src="admin/editor/ckeditor.js" type="text/javascript"></script>
```

و در کد صفحه، بسته به زبان مورد نیاز، یک اسکریپت را به صفحه اضافه نمود؛ در اینجا فارسی استفاده شده:

```
var clientId = "txtPostBody";
var strEditor = strBodyFa = "CKEDITOR.replace( '" + clientId + "', {skin : 'office2003' , language : 'fa' , contentsLangDirection : 'rtl'} );";
ClientScript.RegisterStartupScript(GetType(), clientId + "_Editor", strBodyFa, true);
```

در مثال بالا برای باکس متنی به بنام `txtPostBody`، کد مربوط ایجاد و به صفحه افزوده شده است.
نکته: پوشه مربوط به ادیتور در محلی که در دسترس صفحه باشد باید قرار گیرد که در اکثر موارد قسمت ادمین وب یا اپلیکشن است

نظرات خوانندگان

نویسنده: امیر
تاریخ: ۱۳۹۱/۰۴/۱۱ ۱۱:۴۵

دوست عزیز سلام
مطلوب مفیدی داری که مشتاقانه او نه را پیگیری می کنم
راجع به این ادیتور یه سوال داشتم
ایا استفاده از سورس بدون لاینس امکانپذیره؟
یا حتماً لاینس می خواهد؟
من از کجا می تونم سورسش رو داشته باشم؟

بسیار بسیار متشکرم
گودرزی

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۴/۱۱ ۱۶:۵۶

سلام، میتوانید برای اطلاعات دقیق‌تر بخود سایت ckeditor.com/license رجوع کنید ولی تا اونجایی که من اطلاع دارم تا کد رو تغییر ندید استفاده ازش تو پروژه‌ها رایگان است ولی مورد گالری اون رو باید بخرید. ممنونم

نویسنده: امیر
تاریخ: ۱۳۹۱/۰۴/۱۸ ۱۸:۲۲

با تشکر از پاسخ شما.
 مهمترین چیزی که من از ادیتور می خواهم، قابلیت آپلود عکسه!
 شما در همین باکس ارسال نظر این قابلیت رو دارید در حالی که ckeditor چنین قابلیتی رو نداره.

لطفاً منو راهنمایی کنید.
سپاسگزارم

نویسنده: صابر فتح الله
تاریخ: ۱۳۹۱/۰۴/۱۸ ۱۹:۳

سلام
CKEditor مستقیماً این امکان نداره اما می‌توینی از CKFinder که روی CKEditor نصب میشه و برای این کار طراحی شده استفاده کنید
موفق و موید باشید

نویسنده: بهزاد شیرانی
تاریخ: ۱۳۹۲/۰۵/۲۶ ۱۴:۴۳

آیا از لحاظ امنیتی استفاده از ckEditor مشکلی نداره؟
به خاطر این که تا اونجایی که می‌دونم اگر بخوایم به روشه که شما گفتید از این کنترل استفاده کنیم باید تنظیمات امنیتی صفحه (ValidateRequest) رو غیر فعال کنیم و در این صورت کاربر هر نوع کد HTML که بخواهد می‌توانه توی اون وارد کنه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۲۶ ۱۴:۵۲

- ValidateRequest به صورت پیش فرض به هر نوع تگی عکس العمل نشون می دهد؛ نه فقط به یک سری تگ خاص.
- در MVC می تونید فقط به ازای یک خاصیت، AllowHtml رو فعال کنید و نه به اجبار به ازای کل صفحه یا کل سایت (مثل وب فرم ها).
- بعد از اینکه AllowHtml رو فعال کردید، نیاز خواهید داشت ورودی کاربر را تمیز کنید.

نویسنده: بهزاد شیرانی
تاریخ: ۱۳۹۲/۰۵/۲۶ ۱۵:۳۹

ممnon از پاسختون
ولی من از WebForm استفاده می کنم
فکر می کنم همون تمیز کردن کدها راه خوبی باشه

نویسنده: حسین
تاریخ: ۱۳۹۳/۰۷/۰۹ ۱۲:۲۲

با سلام؛ من ویراšگر متني که در قسمت تماس با استفاده کرده اید را نیاز دارم. در صورت امکان لینک آن را به ایمیل من ارسال کنید. با تشکر فراوران

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۰۹ ۱۲:۳۶

هست. البته بهتر است از Froala استفاده کنید؛ چون پشتیبانی بهتری دارد.

در بیشتر مواردی که در قسمت ادمین سایت می‌گذرد، مثل ریاضیات که شامل چهار عمل اصلی است، افزودن، ویرایش، حذف، و انتخاب نیز، همین وظیفه را در عملیات‌های کار با داده، انجام می‌دهند. البته در این بین اعتبارسنجی را نباید فراموش کرد:

0. Validation
1. Insert
2. Edit
3. Delete
4. Select

برای مثال می‌توان از یک کلاس مشترک برای این کار که از `Page` ارث‌بری کرده است، استفاده و بصورت یک کلاس `abstract` وظیفه پیاده سازی را به صفحه مورد نظر محول کرد. برای نمونه کلاس ساده زیر را در نظر بگیرید:

```
public abstract class PageStateMachine : Page
{
    ...
    protected abstract ... Save();
    protected abstract ... Select();
    protected abstract ... Delete();
    protected abstract ... Validation();
    ...
}
```

و برای روال گردانی هم مطمئناً این متدها باید به ترتیب یکسان در همه صفحات مورد استفاده قرار گیرد، پس لازم است حالت‌های صفحه به این کلاس پاس شود تا بر اساس حالت صفحه، متدها بترتیب صدا زده شوند:

```
public enum PageState
{
    Save = 0,
    Select = 2,
    Delete = 8,
    Cancel = 16
}
```

که در همین کلاس می‌توان متدهای برای این کار در نظر گرفت:

```
public void Go(PageState pageState)
{
    switch (pageState)
    {
        case PageState.Save:
        {
            var error = Validation();
            Message = error.HasError ? error.Message : Save().Message;
            if (!error.HasError)
            {
                ClearControls();
                RefreshList();
            }
            break;
        }
        case PageState.Select:
        {
            Message = Select().Message;
            break;
        }
        break;
        case PageState.Delete:
        {
            Message = Delete().Message;
            ClearControls();
            RefreshList();
            break;
        }
    }
}
```

```
        }
    case PageState.Cancel:
    {
        Message = string.Empty;
        ClearControls();
        break;
    }
}
```

ادامه دارد

نظرات خوانندگان

نویسنده: اردلان شاه قلی
تاریخ: ۱۳۹۱/۰۸/۲۵ ۱۱:۲۷

سلام.

پس از سپاس فراوان می خواستم بدانم ایا ممکنه این موضوع را با مبحث Command Patern بیامیزید. فکر کنم چیز خوبی ازش دربیاید.

خیلی خیلی سپاس

یک فریم ورک سبک برای ساخت سرویس‌های مبتنی بر HTTP بر روی Mono و در واقع پیاده سازی در Ruby برای net است. با استفاده از این کتابخانه شما می‌توانید به سادگی درخواست‌های مختلف کاربران را از طریق وب پاسخ دهید. از ویژگی‌های این کتابخانه امکان اجرای آن بدون نیاز به وجود وب سرور و به صورت Standalone می‌باشد. بهتر است وارد عمل شویم و ببینیم این سیستم چگونه عمل می‌کند. برای شروع ما خود net. Asp را به عنوان میزبان در نظر می‌گیریم.

- یک پروژه خالی Asp.net ایجاد کنید.
- وارد خط فرمان Package Manager شوید و عبارت زیر را وارد کنید:

```
Install-Package Nancy.Hosting.Aspnet
```

- پروژه‌های Nancy از تعدادی ماژول تشکیل شده اند که می‌توانیم برای هر ماژول آدرس وب جداگانه ای در نظر بگیریم. یک کلاس به پروژه اضافه کنید و نام آن را TestModule بگذارید. در متن کلاس عبارات زیر را تایپ کنید:

```
public class TestModule : NancyModule
{
  public TestModule()
  {
    Get["/] = x => "Welcome to my site!";
    Get["/Hello/] = x=> "Hello Nancy!";
    Get["/Bye/{name}"] = x=> "Good bye " + x.name;
  }
}
```

حال کلید F5 را زده و برنامه را اجرا کنید.

حالا در مرورگر خودتان عبارت <http://localhost:12345>Hello> را تایپ کنید. توجه کنید که به جای 12345 باید شماره پورتی که وب سرور دات نت اجراست تایپ کنید. همانطور که متوجه شدید ما در خطوط بالا تعیین کردیم که برای درخواست‌های از نوع Get که مسیر ریشه سایت را درخواست می‌کنند عبارت **Welcome to my site**! ارسال شود. همچنین برای درخواست‌هایی که مسیر **Hello** را درخواست می‌کنند عبارت **Hello Nancy** نمایان می‌شود.

نکته جالب برای درخواست‌های Bye/ است. در اینجا ما یک پارامتر به نام name تعریف کرده ایم و گفته ایم که درخواست‌های Bye که در ادامه آنها عبارتی وجود دارد به صورت Good bye به همراه عبارت بازگردانده شوند.

همین عملیات برای درخواست‌های Put و Head و سایر انواع درخواست وجود دارد.

برای اینکه درخواست‌ها از مسیر خاصی فراخوانی شوند کافی است در هنگام اعلان سازنده کلاس ماژول، مسیر را تعیین کنید.

```
public class TestModule : NancyModule
{
  public TestModule() : base("/test")
  {
    Get["/user"] = x=> "It is test for user";
  }
}
```

این درخواست‌ها از مسیری شبیه این مسیر فراخوانی خواهند شد:

```
http://localhost:12345/test/user
```

همانطور که برای پردازش درخواست از عبارات کوتاه استفاده کردیم می‌توانیم از توابع هم استفاده کنیم.

```
Get["/hello/{username}"] = x=> {
    // some code
    // ...
    // ...
    return "Hello, " + username;
};
```

همچنین امکان بازگرداندن کدهای وضعیت به طور مستقیم وجود دارد.

```
Get["/user"] = x=> return 200;
```

و یا

```
Get["/user"] = x=> return HttpStatusCode.OK;
```

همانطور که گفته شد امکان میزبانی پروژه‌های Nancy از داخل برنامه‌های دات نتی هم وجود دارد. در مقاله بعدی به این موضوع خواهیم پرداخت.

نظرات خوانندگان

نویسنده: بهمن ۸:۴۴ ۱۳۹۱/۰۴/۱۱
تاریخ:

سلام ممنون از اینکه این مطلب را ارسال کردید فقط یه سوالی داشتم اونم اینکه آیا در بیشتر سایتها که آدرس url آنها بدین شکل هست چه asp و چه php از این روش برای خلاصه کردن آدرس سایت استفاده میکنند ...؟

نویسنده: وحید نصیری ۱۰:۴۹ ۱۳۹۱/۰۴/۱۱
تاریخ:

این تکنیک اصطلاحا [Url Rewriting](#) نام دارد.

نویسنده: سلمان عرب عامری ۱۲:۲۴ ۱۳۹۱/۰۴/۱۱
تاریخ:

همانطور که جناب نصیری گفتند، به این روش که آدرس‌ها را کاربر پسند می‌کنیم اصطلاحا Url rewriting آدرس دهی باعت خوانا شدن، و همچنین بهتر ایندکس شدن در موتورهای جستجو می‌شود.
در دات نت پروژه MVC آدرس‌ها را طبق این اصول پیاده می‌کند و کاربر می‌تواند قاعده آدرس دهی خود را به صورت یک Route تعریف کند.

در Asp.net سنتی هم کتابخانه‌های Url Rewriting وجود دارند که برای شما این کار را انجام می‌دهند. ضمن اینکه خود شما هم با نوشتن HttpModule می‌توانید این کار را انجام دهید.

تعريف ActionLink زیر را درنظر بگیرید:

```
@Html.ActionLink("text", "Index", "Home")
```

پارامترهای دوم و سوم آن که به نام‌های یک اکشن متده و کنترلر آن اشاره می‌کنند، توسط رشته‌ها تعریف شده‌اند. مشکلاتی هم که با رشته‌ها در حالت کلی وجود دارند به شرح زیر است:

- (الف) می‌توان نام کنترلر یا نام متده را در برنامه تغییر داد. به این ترتیب تمام ActionLink هایی که در برنامه به این کنترلر اشاره می‌کردند از کار می‌افتد (تکرار رشته‌ها به علاوه refactoring friendly نبودن آن‌ها).
- (ب) برای نوشتن رشته‌ها intellisense کارآیی ندارد.
- (ج) امکان بروز اشتباهات تاییس در این بین بسیار زیاد است.

راه حل متدائلی که برای حل این نوع مشکلات وجود دارد، تعریف یک کلاس عمومی و معرفی رشته‌ها به صورت فیلدی‌ای ثابت در آن‌ها می‌باشد و سپس استفاده از این فیلدی‌ها بجای استفاده مستقیم از رشته‌ها.

و ... چقدر خوب می‌شد اگر ابزاری وجود می‌داشت که کلاس‌های کنترلرهای ما را آنالیز می‌کرد و خودش این ثوابت رشته‌ای را از آن‌ها استخراج و کلاس‌های عمومی یاد شده را تشکیل می‌داد!

خوبشخانه نیازی به اختراع مجدد چرخ نیست و اینکار توسط پروژه‌ی سورس بازی به نام T4MVC انجام شده است. برای دریافت آن به سایت زیر مراجعه نمائید:

<https://t4mvc.codeplex.com>

این پروژه توسط [David Ebbo](#) از اعضای تیم [ASP.NET MVC](#) تهیه شده است.

پس از دریافت پروژه، تنها به دو فایل زیر از آن نیاز داریم:

T4MVC.tt.settings.t4 و T4MVC.tt

دو فایل فوق را به درون پوشه‌ای از پروژه جاری MVC خود کپی کنید (مثلا یک پوشه T4MVC را ایجاد و این دو فایل را به آن اضافه کنید). بلاfacسله این فایلهای t4 وارد عمل شده و کلاس‌های کنترلرهای Viewها، تصاویر و غیره را آنالیز و ... ثوابت رشته‌ای معادل آن‌ها را تولید می‌کنند.

اکنون برای استفاده از این کلاس‌های تولید شده می‌توان به صورت زیر عمل کرد:

اصلاح ActionLink‌ها و حذف رشته‌های موجود در آن‌ها

اینبار بجای اینکه بنویسیم:

```
@Html.ActionLink("text", "Index", "Home")
```

می‌توان نوشت:

```
@Html.ActionLink("text", result: MVC.Home.Index())
```

و یا:

```
@Html.ActionLink("text", MVC.Home.ActionNames.Index, MVC.Home.Name)
```

برای دسترسی به امکانات آن با نام کلاس MVC شروع می‌کنیم و سپس برای مثال به نام کنترلر Home رسیده و توسط ActionNames آن به تمام اکشن متد‌های موجود در آن می‌توان دسترسی داشت.

البته این پروژه بسیار فراتر از تولید فیلد‌های strongly typed معمول رشته‌ها است. همانطور که ملاحظه می‌کنید، یک سری overload را هم به متد‌های پیش‌فرض ASP.NET MVC اضافه کرده است و حتی بجای معرفی رشته معمول اکشن متد Index، خود این اکشن متد را می‌توان معرفی کرد (آرگومانی از نوع ActionResult را هم اضافه کرده است). نمونه دیگر آن به نحو زیر می‌تواند باشد:

```
@Url.Action(result: MVC.Article.Delete())
```

که بجای مورد متد اول ذیل قابل استفاده است:

```
@Url.Action("Delete", "Article")
```

امکان معرفی بهتر نام Partial View‌ها

برای مثال اگر پیشتر یک Partial View را به این شکل تعریف می‌کردید:

```
@{ Html.RenderPartial("_ViewPage1"); }
```

اکنون امکان معرفی آن به نحو زیر فراهم شده است:

```
@{ Html.RenderPartial(MVC.Home.Views._ViewPage1); }
```

همچنین تمام این کلاس‌ها در کنترلرها نیز قابل دسترسی هستند:

```
return PartialView(Views._ViewPage1);
```

اینجا دیگر نیازی به ذکر کلاس MVC نبوده و می‌توان کار را با کلاس جدید Views شروع کرد. یا اگر نیازی به ذکر نام اکشن متدی در کنترلر جاری بود می‌توان از ActionNames مستقیماً استفاده کرد.

و یا بجای:

```
return RedirectToAction(actionName: "Index", controllerName: "Menu");
```

می‌توان نوشت:

```
return RedirectToAction(actionName: MVC.Menu.ActionNames.Index, controllerName: MVC.Menu.Name);
```

و یا حتی به صورت خلاصه‌تر به نحو زیر:

```
return RedirectToAction(result: MVC.Menu.Index());
```

این overload جدیدی است که با T4MVC به پروژه اضافه می‌شود و یک Action Result را می‌تواند پذیرد. به این ترتیب به صورت خودکار نام کنترلر و متد Index آن تنها در یک پارامتر لحاظ می‌شوند. اگر نیاز به ارسال پارامتری هم به متد Index وجود داشت، همینجا می‌توان اینکار را انجام داد (بجای استفاده از anonymously typed objects متد اول).

و یا بجای مسیر دهی به شکل زیر:

```
return PartialView("~/Views/CommentsArchive/_LatestCommentsInfo.cshtml", data);
```

می‌توان نوشت:

```
return PartialView(MVC.CommentsArchive.Views._LatestCommentsInfo, data);
```

همچنین برای مسیر دهی view‌های قرار گرفته در پوشش shared.Views از می‌توان شروع کرد.

امکان معرفی بهتر عناصر استاتیک سایت

این مورد نیز بسیار جالب توجه است. توسط کلاس Links آن می‌توان به محتویات استاتیک (تصاویر، فایل‌های css و غیره) پوشش‌های Content و Scripts هم دسترسی یافت و حتی این موارد را نیز refactor کرد:

```

<script src="@Links.Scripts.jquery_1_5_1_min_js" type="text/javascript"></script>
```

امکان تعریف بهتر پارامترها و مقادیر route

بجای اینکه routeValues را همانند سابق با anonymously typed objects مقدار دهی کنیم:

```
Html.ActionLink(linkText: "عنوان",
                actionName: "Index",
                controllerName: "Comments",
                routeValues: new
                {
                    userName = @Model.FriendlyName
                },
                htmlAttributes: null))
```

اینبار می‌توان نوشت:

```
Html.ActionLink(linkText: "عنوان",
                result: MVC.Comments.Index(userName: @Model.FriendlyName)
                htmlAttributes: null))
```

در overload جدیدی که ملاحظه می‌کنید، هم دسترسی به متدهای Index یک کنترلر از نوع strongly typed است و هم امکان تعریف پارامترهای آن به نحو منطقی‌تری فراهم شده است (متدهای Comments.Index واقعاً وجود خارجی داشته و پارامتری از نوع userName را می‌پذیرد).

چند نکته جانبی

- این ابزار بر اساس Reflection کار می‌کند (البته فقط در حین تشکیل خودکار کلاس‌های مورد نیاز؛ و گرنه ثوابتی را که ایجاد می‌کند کامپایل شده و در زمان اجرا سرباری را به برنامه اضافه نمی‌کند). بنابراین اگر کلاسی به پروژه اضافه شده است، کامپایل کردن آن را فراموش نکنید.

- اگر تغییری در فایل‌های View، در تعداد و نام آن‌ها صورت گرفت، روی فایل T4MVC.tt کلیک راست کرده و گزینه‌ی اجرای آن را انتخاب کنید. پس از این کار، مجدداً کامپایل پروژه را فراموش نکنید.

- در فایل T4MVC.tt.settings.t4 یک سری تنظیمات پیش فرض قرار دارند. برای مثال اگر علاقمندید که به این فایل‌های تولید شده خودکار، فضای نام سفارشی خاصی را اضافه کنید می‌شود آرایه ReferencedNamespaces آن را مقدار دهی کرد.

- overloads های جدید ActionResult دار آن نسبت به نمونه‌های استاندارد موجود، بسیار منطقی‌تر به نظر می‌رسند.
- توضیحات کامل امکانات T4MVC را در [مستندات رسمی](#) آن می‌توانید مطالعه کنید.

و ... اگر یک مدت با آن کار کنید خواهید گفت: «من قبلاً چطور با ASP.NET MVC کار می‌کردم؟!»

نظرات خوانندگان

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳۹۱/۰۴/۱۱ ۱۱:۱۶

خیلی ممنون ... واجب شد که ازش استفاده کنیم دیگه : دی

نویسنده: یوسف نژاد
تاریخ: ۱۳۹۱/۰۴/۱۱ ۱۲:۵

با تشکر. عالی بود. مخصوصا بخش «من قبلا چطور با ASP.NET MVC کار می‌کردم؟!» واقعا.

نویسنده: افشار محبی
تاریخ: ۱۳۹۱/۰۴/۱۱ ۱۶:۱

عاشق روش‌های Refactoring Friendly هستم! در یک پروژه در دست انجام MVC بلافصله شروع به استفاده از این ابزار کردیم... ابزار خوبی به نظر می‌رسد.

نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۱۱ ۱۹:۳۳

اگه در routeValues بخواهیم چند پارامتر بدم ج جوری باید عمل کرد من به صورت زیر میدم
`@Html.ActionLink(linkText: item.Name, result: MVC.Word.Details(id : item.ID, name : item.Name.ToSeoUrl()))`

نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۱۱ ۱۹:۳۷

پدا کردم مرسی

```
Html.ActionLink("Delete Dinner", MVC.Dinners.Delete(Model.DinnerID)
    .AddRouteValue("foo", 17)
    .AddRouteValue("bar", "abc"))
```

نویسنده: ali
تاریخ: ۱۳۹۱/۰۴/۱۱ ۲۱:۱۴

سلام یک سوال داشتم
من از طریق jquery یک iupdatio رو صفحه انجام دادم.

```
<script type="text/javascript">
    function LoadWordInfo(id) {
        showProgress();
        $.ajax({
            type: "Post",
            url: "test/Info",
            data: JSON.stringify({ ID: id }),
            contentType: "application/json; charset=utf-8",
            dataType: "json",
            complete: function (xhr, status) {
                var data = xhr.responseText;
                if (status === 'error' || !data) {

                } else {
                    var dialog = $("#dialog");
                    dialog.html(data);
                    dialog.dialog("open");
                }
            hideProgress();
        });
    }
}</script>
```

```

        return false;
    }
}

function showProgress() {
    $('#Progress').css("display", "block");
}

function hideProgress() {
    $('#Progress').css("display", "none");
}
$(function () {
    $("#dialog").dialog({
        autoOpen: false,
        show: "fade",
        hide: "fade",
        width: 550,
        title: "WordInfo",
        resizable: false
    });
});
</script>

```

این باید یه دیالوگ پر کنه نمایش بد. ولی از وقتی T4MVC استفاده کردم تو کروم درست نشون میده ولی فایرفاکس error زیرو میده "NetworkError: 404 Not Found - http://localhost:6012/test/test/Info" به آدرس دقت کنید دوبار test اورده

نویسنده: وحید نصیری
تاریخ: ۲۱:۲۴ ۱۳۹۱/۰۴/۱۱

در کد فوق که از T4MVC استفاده نشده ولی نحوه تعریف Url شما اشتباه است. قبل ام هم در مورد نحوه تعریف صحیح Url در ASP.NET MVC مطلب نوشتم. حتما باید از Url.Action استفاده کنید. برای اینکه بتونید از Url.Action در اسکریپت‌های خودتون استفاده کنید پایین View جاری یک JavaScript به نام مثلا layout درست کنید. در اینجا می‌شود داخل کدهای جاوا اسکریپتی هم از Razor استفاده کرد. سپس این section را در layout در قسمت include صفحه header کنید (در قسمت 14 سری MVC این سایت در مورد این تکنیک کاملاً توضیح دادم).

نویسنده: رضا.ب
تاریخ: ۲:۴۱ ۱۳۹۱/۰۴/۱۲

نمونه‌های شبیه این باز هم بودند که مسئله‌ای که سوالهاست دات نت با Reflection قابل حل کرده. ولی بازم تو جدیدترین تکنولوژی‌ها از مقادیر رشته‌ای استفاده می‌کنند. آیا هدف خاصی پشت این منطق هست؟ یا صرفاً بی‌توجهی و قصور تیم توسعه هست؟

نویسنده: وحید نصیری
تاریخ: ۸:۳۶ ۱۳۹۱/۰۴/۱۲

Reflection اندکی کند است. در حین توسعه برنامه اگر ویژوال استودیو مرتباً بخواهد کوئری‌های مبتنی بر reflection از اسمبلی‌های شما بگیرد، سرعت کار شما کم خواهد شد و نهایتاً ممکن است سبب ناپایداری آن هم بشود (اگر پروژه جاری بزرگ باشد).

نویسنده: افشار محبی
تاریخ: ۲۳:۷ ۱۳۹۱/۰۴/۱۲

در پروژه ما کنترلرها در یک class library جداگانه نگهداری می‌شوند. در این حالت T4MVC دچار مشکل می‌شود...

<http://forums.asp.net/p/1683368/5049673.aspx>

من T4MVC را به خود class library محل نگهداری Controllerها منتقل کردم و مشکلاتم حل شد.

نویسنده: حسین توسلی
تاریخ: ۱۷:۱۴ ۱۳۹۱/۰۶/۱۰

با سلام و خسته نباشید خدمت دوستان
آقا من نتونستم این فایل‌ها رو دانلود کنم، خواهش می‌کنم يه راهی بگین تا منم بتونم دانلودشون کنم.
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۸:۴۷ ۱۳۹۱/۰۶/۱۰

از [اینجا](#) دریافت کنید.

نویسنده: ایلیا اکبری فرد
تاریخ: ۸:۵۹ ۱۳۹۱/۰۸/۲۰

مهندس سلام.
من موقع اضافه کردن T4mvc از طریق نوگت تحت T4MVC.tt.settings.t4 فایلی که mvc4 نام داشته باشد را نمی‌بینم. کجای کارم ایراد دارد؟ با تشکر.

نویسنده: وحید نصیری
تاریخ: ۹:۱۰ ۱۳۹۱/۰۸/۲۰

اگر علاقمند هستید که اخبار آنرا دنبال کنید نیاز است سورس کنترل آنرا مرور کنید:

<http://t4mvc.codeplex.com/SourceControl/list/changesets>

24cc121c697f, by Michael Swain, Oct 22 8:39 PM

XmlSettings Patch

Patch to modify T4MVC to use an XML settings file instead of a static include. This allows for future compatibility as long as setting types do not change.

نام جدید آن T4MVC.tt.settings.xml شد.
+ این فایل به همراه بسته NuGet آن نیست و باید از سورس کنترل دریافت شود (داخل پوشه T4MVCHostMvcApp\T4MVC Files).

نویسنده: ایلیا اکبری فرد
تاریخ: ۹:۱۹ ۱۳۹۱/۰۸/۲۰

سپاس مهندس.

نویسنده: SMF
تاریخ: ۱۰:۳۲ ۱۳۹۱/۱۲/۱۰

با سلام

من تو 2012 V.S. تعدادی کنترل جدید به پروژه اضافه کردم ولی T4MVC لیست نمی‌کردشون، کامپایل مجدد کل سولوشن هم کمکی نکرد؛ بعد روی فایل tt راست کلیک و Run Custom Tool که زدم T4MVC خودشو به روز کرد؛ گفتم شاید به درد دوستانی که به مشکل مشابه بخوردن، بخوره ...

نویسنده: cooli

تاریخ: ۱۳۹۱/۱۲/۱۷ ۱۲:۵۰

دو سوال

- 1 - در ورزشی که بنده از NuGet نصب کردم (3.5.1) ، کار T4MVCExtensions هم نصب گردید ، کار T4MVC چیست ؟
- 2 - بعد از اجرا ، تمام کلاس‌های Controller های پروژه من Action های من partial شدند و تمام virtual های من شدند ، چرا ؟ آیا مشکلی در آینده برای Controller ها و Action ها پیش نخواهد آمد ؟

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۱۲/۱۷ ۱۲:۵۶

- یک سری متدهایی برای کار با T4MVC که به صورت پیش فرض در MVC وجود ندارد؛ خصوصاً برای کار با متدهایی که پارامتر Result دارند و متدهایی مانند MVC.Comments.Index را قبول می‌کنند.

- خیر. این روش کلی کار کتابخانه T4MVC است برای فراهم آوردن امکان overriding مسایل توکار خودش به همراه افزودن یک سری تعاریف قابل استفاده در کنترلر جاری. مثلاً زمانی که می‌نویسند View.MyPartialViewName، نیاز است کنترلر جاری partial تعريف شود تا به رشته یاد شده که توسط T4MVC اضافه می‌شود دسترسی یافتد.

در کل کار کردن با T4MVC علاوه بر دسترسی strongly typed به مسیرها و فایل‌های موجود پروژه، سرعت نهایی رندر شدن را هم بالا می‌برد؛ چون مسیرهایی را که ایجاد می‌کند، نیاز به کمترین حدس‌زنی و سعی و خطا را توسط موتور View جاری برای یافتن منبع مورد نظر دارد.

نویسنده: امیر

تاریخ: ۹:۱۶ ۱۳۹۱/۱۲/۲۰

فقط من پوشه رو اضافه کردم ولی زمانی که میخام مثل شما تو اکشن لینک بنویسم نمیشناسه مسیرها رو. خودش نمیاره مثل میخام به شاخه ویو دسترسی داشته باشم اصلاً نمیشناسه

نویسنده: وحید نصیری

تاریخ: ۹:۱۹ ۱۳۹۱/۱۲/۲۰

رو اجرا کردید؟ [run custom tool](#)

نویسنده: مولانا

تاریخ: ۱۱:۵۹ ۱۳۹۲/۰۱/۲۶

با سلام.

من T4MVC را بر روی پروژه اصلی قرار دادم.

فایل‌های resource من در یک پروژه جدا قرار دارند. مدل‌های پروژه من هم در یک پروژه جدا قرار دارند. حال برای تعريف خاصیت اعتبارسنجی روی یکی از خاصیت‌های ویومدل آیا می‌شود T4MVC را بکار برد؟ (بجای استفاده از magic string)

```
[StringLength(25, ErrorMessageResourceType = typeof(ValidationErrorsResource), ErrorMessageResourceName = "InvalidStringLength")]
public string FatherName { get; set; }
```

نوبتندۀ: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۲۶ ۱۲:۴۰

از استفاده کنید.

نوبتندۀ: مولانا
تاریخ: ۱۳۹۲/۰۱/۳۱ ۱۶:۱۸

با سلام.

من پوشه‌ای با عنوان SiteContent در root پروژه دارم. چگونه می‌توانم از طریق T4MVC به آن دسترسی داشته باشم. در حالت عادی وقتی Links را تایپ می‌کنم فقط منابع Scripts و Content و شناسایی می‌کند. باتشکر.

نوبتندۀ: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۳۱ ۱۷:۳۹

مراجعه کنید به [راهنمای مفصل آن](#). نقل قولی از آن:

Use StaticFilesFolders to tweak the list of folders from which links are generated

```
<!-- Folders containing static files for which links are generated (e.g. Links.Scripts.Map_js) -->
<StaticFilesFolders>
  <FileFolder>Scripts</FileFolder>
  <FileFolder>Content</FileFolder>
</StaticFilesFolders>
```

نوبتندۀ: علی
تاریخ: ۱۳۹۲/۰۴/۱۵ ۱۹:۴۶

با سلام بندۀ داخل پروژه ام از t4mvc استفاده کردم امروز صبح یک partialview اضافه کردم وقتی روی T4MVC.tt راست کلیک کردم و run custom tool را زدم با خطای :

```
Error1Running transformation: System.ArgumentException: The file 'T4MVC.tt' is not open.
  at System.Runtime.InteropServices.Marshal.ThrowExceptionForHRInternal(Int32 errorCode, IntPtr
errorInfo)
  at System.Runtime.InteropServices.Marshal.ThrowExceptionForHR(Int32 errorCode)
  at Microsoft.VisualStudio.NativeMethods.ThrowOnFailure(Int32 hr, Int32[] expectedHRFailure)
  at Microsoft.VisualStudio.Shell.RunningDocumentTable.FindDocument(String moniker, IVsHierarchy&
hierarchy, UInt32& itemid, UInt32& docCookie)
  at Microsoft.XmlEditor.XmlHelper.GetCanonicalNameWithoutView(IServiceProvider site, String fname)
  at Microsoft.XmlEditor.XmlFileInfoService.RenameOrRemoveCacheFileInfo(String fname)
  at Microsoft.XmlEditor.Package.GetOrCreateInfo(String fname, String text)
  at Microsoft.XmlEditor.Package.GetOrCreateInfo(String fname, IVsTextLines buffer)
  at Microsoft.XmlEditor.FactoryBase.GetDesignerFactory(String monikerTheFileOnDisk, String
monikerToUse, IVsTextLines existingBuffer, Guid logicalView)
  at Microsoft.XmlEditor.Chooser.ChooseEditorFactory(String pszMkDocument, IVsHierarchy pHier, UInt32
itemid, IntPtr punkDocDataExisting, Guid& rguidLogicalView, Guid& pguidEditorTypeActual, Guid&
pguidLogicalViewActual)
  at EnvDTE.ProjectItem.Open(String ViewKind)
  at Microsoft.VisualStudio.TextTemplating07E2522337018FCFC0AADC805726764A22758F53E05D3026845E2AA5E0226E57B7
7322743831B8A035E8EFE18D3CC1F808985A2958CCE46A2085D3520D2EC605.GeneratedTextTransformation.XmlSettings.
Init(ITextTemplatingEngineHost host) in e:\Project\GRP\GRP.Web\T4MVC.tt:line 2120
  at Microsoft.VisualStudio.TextTemplating07E2522337018FCFC0AADC805726764A22758F53E05D3026845E2AA5E0226E57B7
7322743831B8A035E8EFE18D3CC1F808985A2958CCE46A2085D3520D2EC605.GeneratedTextTransformation.XmlSettings.
Load[T](ITextTemplatingEngineHost host) in e:\Project\GRP\GRP.Web\T4MVC.tt:line 2093
  at Microsoft.VisualStudio.TextTemplating07E2522337018FCFC0AADC805726764A22758F53E05D3026845E2AA5E0226E57B7
7322743831B8A035E8EFE18D3CC1F808985A2958CCE46A2085D3520D2EC605.GeneratedTextTransformation.MvcSettings.
Load(ITextTemplatingEngineHost host) in e:\Project\GRP\GRP.Web\T4MVC.tt:line 1786
  at Microsoft.VisualStudio.TextTemplating07E2522337018FCFC0AADC805726764A22758F53E05D3026845E2AA5E0226E57B7
7322743831B8A035E8EFE18D3CC1F808985A2958CCE46A2085D3520D2EC605.GeneratedTextTransformation.TransformTex
t() in e:\Project\GRP\GRP.Web\T4MVC.tt:line 39e:\Project\GRP\GRP.Web\T4MVC.tt21201GRP.Web
```

مواجهه شدم t4mvc uninstall را کردم دوباره نصب کردم موقع generate دوباره همین پیغام را میدم. لطفاً یکی به من کمک کنند. ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۱۵ ۲۰:۱۳

مشکلات مرتبط رو [اینجا گزارش کنید](#).

نویسنده: محمد آزاد
تاریخ: ۱۳۹۲/۰۴/۱۶ ۱۴:۵۹

یکی از مشکلاتی که من با این ابزار خوب دارم اینه که بعد از هر تغییر در کنترلرهای اضافه کردن یک کنترلر جدید حتماً باید [run](#) اجرا کنم... می خواستم بدنم آیا راهی نیست که بشه این قضیه خودکار انجام بشه؟ (مثلاً بعد از Build پروژه) [custom tool](#)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۱۶ ۱۶:۲۷

[add-in](#) برای اینکار هست. ولی به شخصه ترجیح می‌دهم خودکار اجرا نشود؛ چون در یک پروژه با تعداد فایلهای بالا، پردازش آن مدتی زمان می‌برد و ممکن است در این بین، تغییرات انجام شده شامل تغییری در تعداد یا نام فایلها یا اکشن‌های متدها نباشند. تغییرات در سایر لایه‌ها باشند. در این حالت اصلاً ضرورتی ندارد که با هر Build یکبار اینکار تکرار شود.

نویسنده: کامی
تاریخ: ۱۳۹۲/۰۵/۲۶ ۰:۱۰

سلام؛ من از T4 استفاده کردم ولی وقتی پروژه رو [publish](#) می‌کنم و اون رو روی هاست upload می‌کنم t4 رو نمی‌شناسه و تمام مواردی رو که از t4 استفاده کردم رو [error](#) میده. آیا باید تنظیمات خاصی انجام بدم که این مشکل رو حل کنم؟ لطفاً راهنمایی کنید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۲۶ ۰:۲۶

کل کاری که T4MVC انجام می‌ده، اضافه کردن یک سری متدهای الحاقی جدید است به پروژه به علاوه تبدیل یک سری از مسیرها به معادل‌های آن و اندکی تغییر در متدهای کنترلرهای سازگار شدن با متدهای الحاقی جدید. بنابراین حاصل این عملیات به صورت کامپایل شده در پوشه bin پروژه شما موجود است. این پوشه را کلاً بر روی سرور ارسال کنید.

نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۰۸/۲۲ ۰:۴۸

«من قبلاً چطور با ASP.NET MVC کار می‌کردم؟!»

نویسنده: رویا حیدری
تاریخ: ۱۳۹۳/۰۳/۱۶ ۱۷:۱۶

من در پروژم Controller ها رو در یک لایه و View ها را در لایه دیگری دارم. از T4MVC هم در لایه Controller استفاده کردم. اما بعد از زدن Run Custom Tool مسیرهایی که منتهی به یک view می‌شوند ساخته نمی‌شوند و Error میده! (آنها یعنی که [RedirectToAction](#) هستند به درستی ساخته نمی‌شوند)

```
VDoc.Web.Controllers.TopicController.ViewsClass' does not contain a definition for '_Edit' and no
extension method '_Edit' accepting a first argument of type
'VDoc.Web.Controllers.TopicController.ViewsClass' could be found (are you missing a using directive or
an assembly reference?)
```

آیا برای استفاده از T4 در لایه‌های مختلف باید کار خاصی انجام بدم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۱۸:۴

- ViewsRootFolder داخل فایل تنظیمات آن، قابل تنظیم است.

- همچنین بحث هم در [اینجا](#) شده با [یک چنین راه حلی](#)

نویسنده: احمد ولی پور
تاریخ: ۱۳۹۳/۰۷/۱۵:۰۴:۴۲

من سه تا فایل T4MVC.tt و T4MVC.tt.settings.xml و حتی T4MVC.tt.hooks.t4 را تو به پوشه تو سطح اول میدازم ولی همش خطای زیر را میده:

```
Could not load file or assembly 'System.Web.Mvc, Version=3.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35' or one of its dependencies. The located assembly's manifest definition does
(not match the assembly reference. (Exception from HRESULT: 0x80131040
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۱۵:۱۸:۱۱

تنظیمات فایل‌های کانفیگ برنامه شما (bindingRedirect) صحیح نیست یا ناقص است. [اطلاعات بیشتر](#)

نویسنده: rezal10
تاریخ: ۱۳۹۳/۰۸/۲۸:۱۵:۴۳

با سلام

T4MVC یکسری تغییرات در برنامه ما ایجاد می‌کنه از جمله مثلاً یک کانسٹراکتور برای کنترلر Home اضافه می‌کنه و من چگونه باید کدی مثل کد زیر را داخل کنترلر ام بنویسیم که خطای تکراری نگیره؟

```
private readonly ICacheService _cacheService;
public HomeController() : this(MvcApplication.CacheService)
{
}
public HomeController(ICacheService cacheService)
{
    _cacheService = cacheService;
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۲۸:۱۸:۱۳

- روش « [poor man's dependency injection](#) » اساساً زیر سؤال هست. روش صحیح تزریق وابستگی‌ها در برنامه‌های ASP.NET MVC در [اینجا](#) بحث شده است.
- اگر نمی‌خواهید T4MVC سازنده‌ی پیش فرض ایجاد کند، فایل‌های T4 آن را [ویرایش کنید](#).

نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۴/۰۲/۲۶:۱۹:۲۰

گاهی اوقات روی خط زیر

```
SmObjectFactory.Container.GetInstance(controllerType) as Controller;
```

با خطای زیر روبرو میشم:

```
http://localhost:16926/T4MVC_System_Web_MvcActionResult
```

نویسنده: وحید نصیری
تاریخ: ۱۹:۴۲ ۱۳۹۴/۰۲/۲۶

بررسی کنید چه آدرسی درخواست شده و آیا موجود است یا خیر؟

```
protected override IController GetControllerInstance(HttpContext requestContext, Type controllerType)
{
    if (controllerType == null && requestContext.HttpContext.Request.Url != null)
        throw new InvalidOperationException(string.Format("Page not found: {0}",
requestContext.HttpContext.Request.Url.AbsoluteUri.ToString(CultureInfo.InvariantCulture)));
    return SmObjectFactory.Container.GetInstance(controllerType) as Controller;
}
```

نویسنده: علی یگانه مقدم
تاریخ: ۱۹:۵۸ ۱۳۹۴/۰۲/۲۶

در Urlها که همین آدرس بالا رو زده بود ولی از اونجا که شما گفتید آدرس درخواستی، قسمت‌های مختلف Layout را چک کردم و خطرا رو پیدا کردم و لینک رو صحیح ارسال نکرده بودم که اصلاح شد
متشرکم

نویسنده: علی یگانه مقدم
تاریخ: ۲۰:۹ ۱۳۹۴/۰۲/۲۶

یک مورد اینکه من پروژه رو در حال اجرا برای مدتی رها کردم
وقتی برگشتم با خطای زیر مواجه شدم

```
http://localhost:16926/cf05761417b442f293d1cd9f0bffd789/arterySignalR/ping?requestUrl=http%3A%2F%2Floca
lhost%3A16926%2F&browserName=Firefox&_=1431790628217
```

این خطا توسط کدی که گفتید گزارش شده
... page not found

نویسنده: سیروان عفیفی
تاریخ: ۸:۲۹ ۱۳۹۴/۰۲/۲۷

این خط رو به فایل RouteConfig.cs اضافه کنید:

```
routes.IgnoreRoute("{*browserlink}", new { browserlink = @".*/arterySignalR/ping" });
```

در ادامه [این بحث](#) مثال محصولات برای ثبت و ویرایش و حذف محصول با توجه به متدهایی که باید پیاده سازی شود:

```
public partial class product : PageStateMachine
{
    protected void btnSave_Click(object sender, EventArgs e)
    {
        Go(PageState.Save);
    }
    protected override ... Save()
    {
        try
        {
            if (Id > 0)
            {
                // get from bank and update
            }
            else
            {
                // create new one and add it to context
            }
        }
        catch (Exception ex)
        {
            // handle exception
        }
    }

    protected void btnSelect_Click(object sender, EventArgs e)
    {
        Go(PageState.Select);
    }
    protected override ... Select()
    {
        try
        {
            // get from bank and show in details
        }
        catch (Exception ex)
        {
            // handle exception
        }
    }

    protected void btnDelete_Click(object sender, EventArgs e)
    {
        Go(PageState.Delete);
    }
    protected override StateMachineMessage Delete()
    {
        try
        {
            // remove from bank
        }
        catch (Exception ex)
        {
            // handle exception
        }
    }
}
```

به این ترتیب روال ذخیره و بازیابی و ویرایش و حذف به صورتی توسط PageStateMachine مدیریت شده و کافیست متدها در سمت کد پشت صفحه بدرستی نوشته شود .

کتابخانه ای ساده و سبک برای نگاشت اطلاعات یک شی به شی دیگر به صورت خودکار هست و... اگه این پست رو مطالعه کرده باشید یه مشکل امنیتی بنام «Mass Assignment» مطرح شد. برای رفع این مشکل یک روش استفاده از **ViewModel** بود. فرض کنید ما

```
public class User
{
    public int Id { get; set; }

    public string FirstName { get; set; }

    public string LastName { get; set; }

    public string UserName { get; set; }

    public string Password { get; set; }

    public bool IsAdmin { get; set; }

    public virtual ICollection<BlogPost> BlogPosts { get; set; }
}
```

و **ViewModel** ما

```
public class UserViewModel
{
    public string FirstName { get; set; }

    public string LastName { get; set; }

    public string Password { get; set; }
}
```

باشه(توجه کنید در واقع من برای View ی مورد نظرم فقط به نام ، نام خانوادگی و پسورد نیاز دارم) برای استفاده UserViewModel در View مدل بعنوان UserViewModel باید شی User را با اطلاعات شی User مقدار دهی کنیم مثلًا با کدی مثل این در کنترلر.

```
public ActionResult Index(int id = 1)
{
    var user = _userService.GetById(id);
    var userViewModel = new UserViewModel
    {
        FirstName = user.FirstName,
        LastName = user.LastName,
        Password = user.Password
    };
    return View(userViewModel);
}
```

رهایی از نوشتن اینجور کدهای تکراری و خسته کننده باعث پیدایش AutoMapper شد... برای استفاده از AutoMapper از نوگت استفاده میکنیم.

PM> Install-Package AutoMapper

در شروع برنامه نگاشت‌ها را تعریف می‌کنیم. یک روش [ابداعی](#) تعریف نگاشت‌ها در یک کلاس استاتیک و فراخوانی اون تو متدر. Application_Start

```
public static class AutoMapperWebConfiguration
{
    public static void Configure()
    {
        ConfigureUserMapping();
    }

    private static void ConfigureUserMapping()
    {
        Mapper.CreateMap<User, UserViewModel>();
    }
}
```

اولین پارامتر نوع مبدا و دومین پارامتر نوع مقصد هست.
برای انجام نگاشت هم از متدر Map استفاده می‌کنیم.

```
public ActionResult Index(int id=1)
{
    var user = _userService.GetById(id);
    var userViewModel=new UserViewModel();
    AutoMapper.Mapper.Map(user, userViewModel);

    return View(userViewModel);
}
```

همنطور که می‌بینید با نوشتتن چندین خط کد عملیات نگاشت اطلاعات یک شی به شی دیگه انجام شد.
ادامه دارد...

نظرات خوانندگان

نویسنده: رضا
تاریخ: ۱۳۹۱/۰۴/۱۳ ۱۷:۳۲

دمت گرم مدته معطل همین مپینگ‌ها به صورت دستی هستم

نویسنده: سروش ترک زاده
تاریخ: ۱۳۹۱/۰۴/۱۳ ۱۷:۳۳

سلام
ممnon،
راه حل ساده واقعا لذت بخش هست...

نویسنده: daniel
تاریخ: ۱۳۹۱/۰۴/۱۳ ۲۱:۱۳

تو [valueinjecter](#) از Compact Framework باید استفاده کرد.

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳۹۱/۰۴/۱۳ ۲۳:۲

خیلی ممنونم ... میشه این سمپل رو ضمیمه کنید؟!

نویسنده: رضاب
تاریخ: ۱۳۹۱/۰۴/۱۳ ۲۳:۴۶

تشکر بابت معرفی.
کاربرد این Auto Mapper در اینجور نگاشتها معنی میده فقط؟ یعنی نگاشت یه شی جزء به شی کل؟ یا مثلا کاربردهایی مثل ORM هم میتواند داشته باشد؟

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۰۴/۱۴ ۹:۲۱

یکی از مهمترین کاربردها نگاشت از یک مدل بزرگ و پیچیده به یک مدل ساده هست که بهش Flattening میگن و کاربردهای دیگه مثل Projection و ... که همنظور که آخر پست گفتم تو پست‌های بعدی تکمیلش میکنم.
برای مطالعه بیشتر همه به این [ویکی](#) مراجعه کنید.

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۰۴/۱۴ ۱۲:۱۶

سمپل خاصی نیست کدها رو از یک پروژه کشیدم بیرون...
تنها قسمتی که تو این پست نداشتمن (و البته نیاز هم نبود) View ش بود.

```
@model DotnetDevBlog.Web.Models.UserViewModel  
{@  
    ViewBag.Title = "Index";  
  
<h2>Index</h2>  
@Html.DisplayForModel()
```

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۲/۰۴/۰۲ ۱۶:۱۹

با سلام
منون از مطلب مفیدتون
میخواستم بدونم اگه از چند model بخواهیم استفاده کنیم باید چکار کنیم؟

نویسنده: محمد صاحب
تاریخ: ۱۳۹۲/۰۴/۰۴ ۱۰:۵۸

از این کلاس کمک استفاده کن

hossein101211 نویسنده:
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۳:۵۸

با سلام میخواستم ببینم آیا امکان استفاده از automapper بدین صورت هم وجود دارد؟

```
public void Edit_news_ajax(News news)
{
    using (var Context = new ProCamContext())
    {
        var q=Context.News.Find(news.id);
        AutoMapper.Mapper.Map(news,q);
        //NewsRepository.EditNews(news.id, q);
        Context.SaveChanges();
    }//end news
}
```

با تشکر

hossein101211 نویسنده:
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۴:۳۰

اگه q از نوع news هست که نیازی به AutoMapper نداری!

```
AutoMapper.Mapper.Map(news,q);
```

این خط مشکلی نداره

hossein101211 نویسنده:
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۵:۵

من میخواهم محتویات news رو (که توسط model binding پر شده) رو در سطحی از جدولم بریزم (با اون آیدی که با find پیدا کردم جایگزین بشه) اینکار با AutoMapper امکان پذیر هست؟
برای اینکار نیاز به create.map هم دارم؟
با تشکر

hossein101211 نویسنده:
تاریخ: ۱۳۹۲/۰۴/۱۱ ۱۱:۲۷

اگه نوع هاشون متفاوته بله باید Mapper.CreateMap را تعریف کنی.

در ASP.NET MVC امکان استفاده از چند View Engine به صورت همزمان وجود دارد و همچنین هر بار که قرار است View ای رندر شود، از تمام این‌ها تا یافتن موتور مناسب نمایش View جاری کوئری می‌گیرد. بدیهی است هر چقدر تعداد موتورهای ثبت شده در اینجا بیشتر باشند، زمان بیشتری نیز برای یافتن موتور نمایشی مناسب صرف خواهد شد؛ خصوصاً اگر موتور مناسب در آخر لیست ثبت شده باشد.

در ASP.NET MVC 3 دو موتور نمایشی به صورت پیش فرض نصب هستند (WebForms and Razor). بنابراین اگر صرفاً از Razor استفاده می‌کنید، می‌توان موتور اول را کلا از سیستم پردازشی برنامه حذف کرد. برای اینکار تنها کافی است در فایل global.asax.cs برنامه بنویسیم:

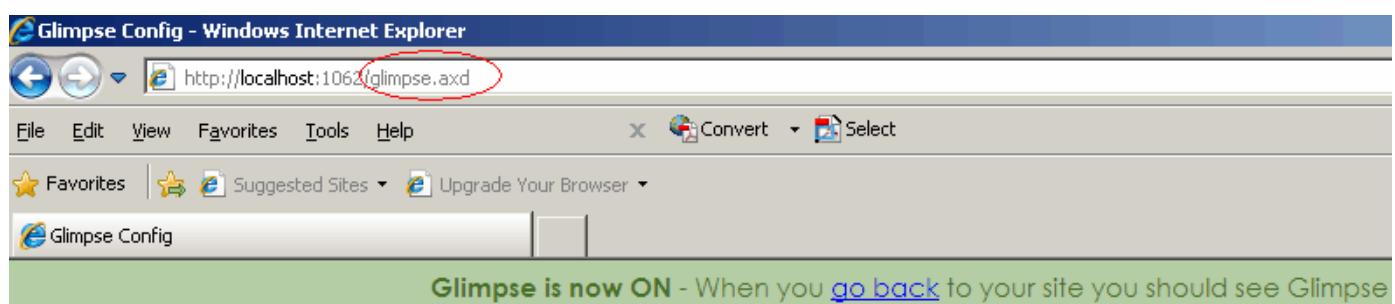
```
protected void Application_Start() {
    ViewEngines.Engines.Clear();
    ViewEngines.Engines.Add(new RazorViewEngine());
    ...
}
```

این موارد را توسط [Glimpse](#) بهتر می‌توان بررسی کرد. Glimpse یک پروفایلر سمت سرور ASP.NET است و دارای نسخه مخصوص ASP.NET MVC نیز می‌باشد. برای نصب آن باید از طریق NuGet اقدام کرد و حتماً دقت داشته باشید که نسخه MVC آن باید نصب شود تا برگه‌های Routing و View ظاهر شوند.

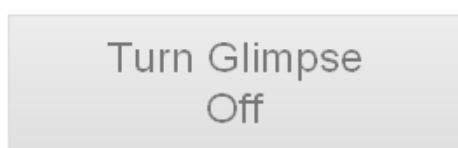
پس از نصب از طریق NuGet، به صورت خودکار اسمبلی‌های لازم به پروژه اضافه شده و همچنین فایل web.config برنامه نیز ویرایش می‌شود. در انتهای این فایل سطر ذیل مشخص می‌کند که Glimpse فعال باشد یا خیر.

```
<glimpse enabled="true" />
```

پس از نصب، برنامه را اجرا کرده و به آدرس <http://localhost/glimpse.axd> مراجعه کنید تا صفحه تنظیمات آن ظاهر شود. تنها کاری که باید در اینجا صورت گیرد کلیک بر روی دکمه Turn Glimpse On است.



What Firebug
the server... in
Glimpse into w



به این ترتیب یک کوکی به مرورگر اضافه شده و اکنون پس از بازگشت به صفحه اصلی برنامه و refresh کامل صفحه، در کنار سمت راست پایین صفحه، آیکن آن ظاهر خواهد شد.



بر روی این آیکن کلیک نمائید تا در برگه‌ی View آن، انواع View‌هایی که در گیر نمایش صفحه جاری بوده‌اند، مشخص شوند:

Ajax	Binding	Config	Environment	Execution	MetaData	Remote	Request	Routes	Server
Ordinal	Requested View	Master Override	Partial	View Engine	Check Cache	Found	Details		
1	Index		False	WebFormViewEngine	True	False	Not Found In WebFormViewEngine cache		
2	Index		False	RazorViewEngine	True	False	Not Found In RazorViewEngine cache		
3	Index		False	WebFormViewEngine	False	False	Not Found In ~/Views/Home/Index.aspx ~/Views/Home/Index.ascx ~/Views/Shared/Index.aspx ~/Views/Shared/Index.ascx		
4	Index		False	RazorViewEngine	False	True	Key Value ViewData -- Model -- TempData --		

همانطور که ملاحظه می‌کنید در اینجا دو موتور پیش فرض فعال بوده و پس از سعی و خطای صورت گرفته، در انتهای کار Razor انتخاب شده است. اکنون اگر نکته حذف موتورهای نمایشی اضافی را اعمال کنیم به تصویر زیر خواهیم رسید:

The screenshot shows a table titled 'View Engine' with columns: Ordinal, Requested View, Master Override, Partial, View Engine, Check Cache, Found, and Details. Row 1 (Ordinal 1) has 'Index' in 'Requested View' and 'RazorViewEngine' in 'View Engine'. The 'Found' column is circled in red and shows 'False'. The 'Details' section shows 'Not Found In' and 'RazorViewEngine cache'. Row 2 (Ordinal 2) also has 'Index' in 'Requested View' and 'RazorViewEngine' in 'View Engine'. The 'Found' column is circled in red and shows 'True'. The 'Details' section shows a table with columns 'Key' and 'Value' containing ' ViewData --', ' Model --', and ' TempData --'.

Ordinal	Requested View	Master Override	Partial	View Engine	Check Cache	Found	Details								
1	Index		False	RazorViewEngine	True	False	<table border="1"><tr><td>Not Found In</td></tr><tr><td>RazorViewEngine cache</td></tr></table>	Not Found In	RazorViewEngine cache						
Not Found In															
RazorViewEngine cache															
2	Index		False	RazorViewEngine	False	True	<table border="1"><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>ViewData</td><td>--</td></tr><tr><td>Model</td><td>--</td></tr><tr><td>TempData</td><td>--</td></tr></tbody></table>	Key	Value	ViewData	--	Model	--	TempData	--
Key	Value														
ViewData	--														
Model	--														
TempData	--														

هم تعداد سعی و خطاهای کمتر شده و هم تعداد فایل‌هایی که بررسی شده است به حداقل رسیده (برای مثال در حالتیکه موتور WebForms فعال باشد، چهار فایل با پسوندات مختلف در مکان‌های پیش فرض نیز حتما جستجو خواهد شد).

نظرات خوانندگان

نویسنده: شهریار جعفری
تاریخ: ۲۳:۵۸ ۱۳۹۱/۰۴/۱۳

یه سوال شاید نه چندان حرفه ای
شما گفتید که ا مکان استفاده از چند View Engine به صورت همزمان وجود دارد. آیا میشه تو یک ویو این کارو انجام داد؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۹۱/۰۴/۱۴

خیر. همزمانی در اینجا به معنای امکان استفاده همزمان از فایل‌های مستقل aspx و cshtml یا vbhtml است.

نویسنده: شهریار جعفری
تاریخ: ۰:۱۹ ۱۳۹۱/۰۴/۱۴

یه سوال نامرتبه Ninject چیه؟ کاره همون structuremap میکنه؟

نویسنده: وحید نصیری
تاریخ: ۰:۲۷ ۱۳۹۱/۰۴/۱۴

بله. از این نوع [کتابخانه‌ها](#) زیاد است. Ninject, StructureMap, AutoFac .Unity دارد.

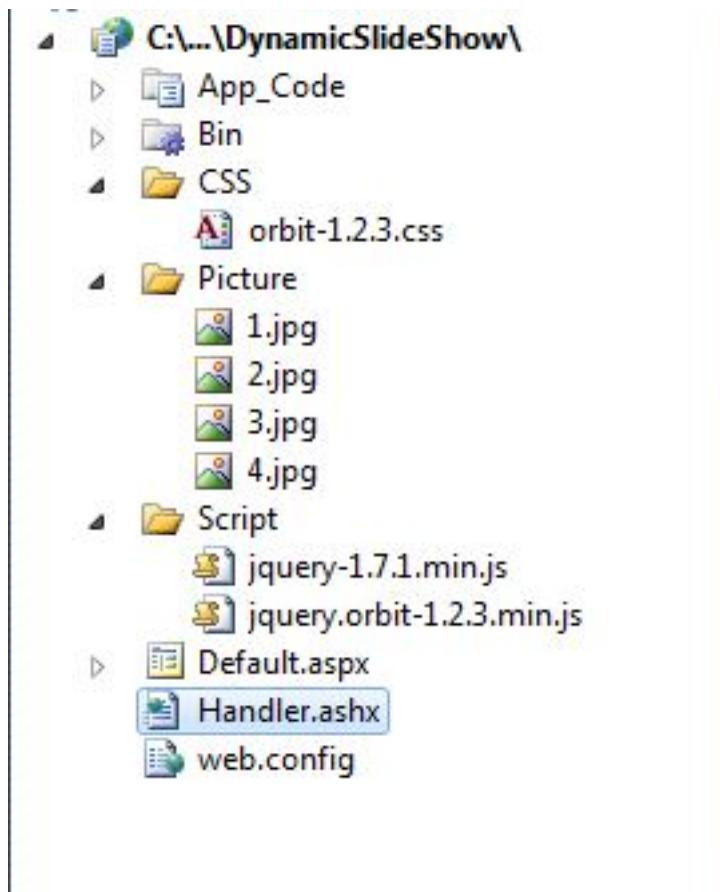
در این آموزش قصد دارم نحوه ایجاد یک Slideshow به صورت داینامیک را با ASP.NET طراحی کنم(منظور از ایجاد Slideshow صورت داینامیک این است که عکس‌ها را به صورت داینامیک از DB بخواند).

اولین گام این است که plugin مورد نظر را دریافت کنید که من از پلاگین [Orbit](#) استفاده کرده ام

ابتدا یک DataBase با نام DynamicSlideShow ایجاد و یک جدول با ساختار زیر با نام Pictures درون آن ایجاد می‌کنیم

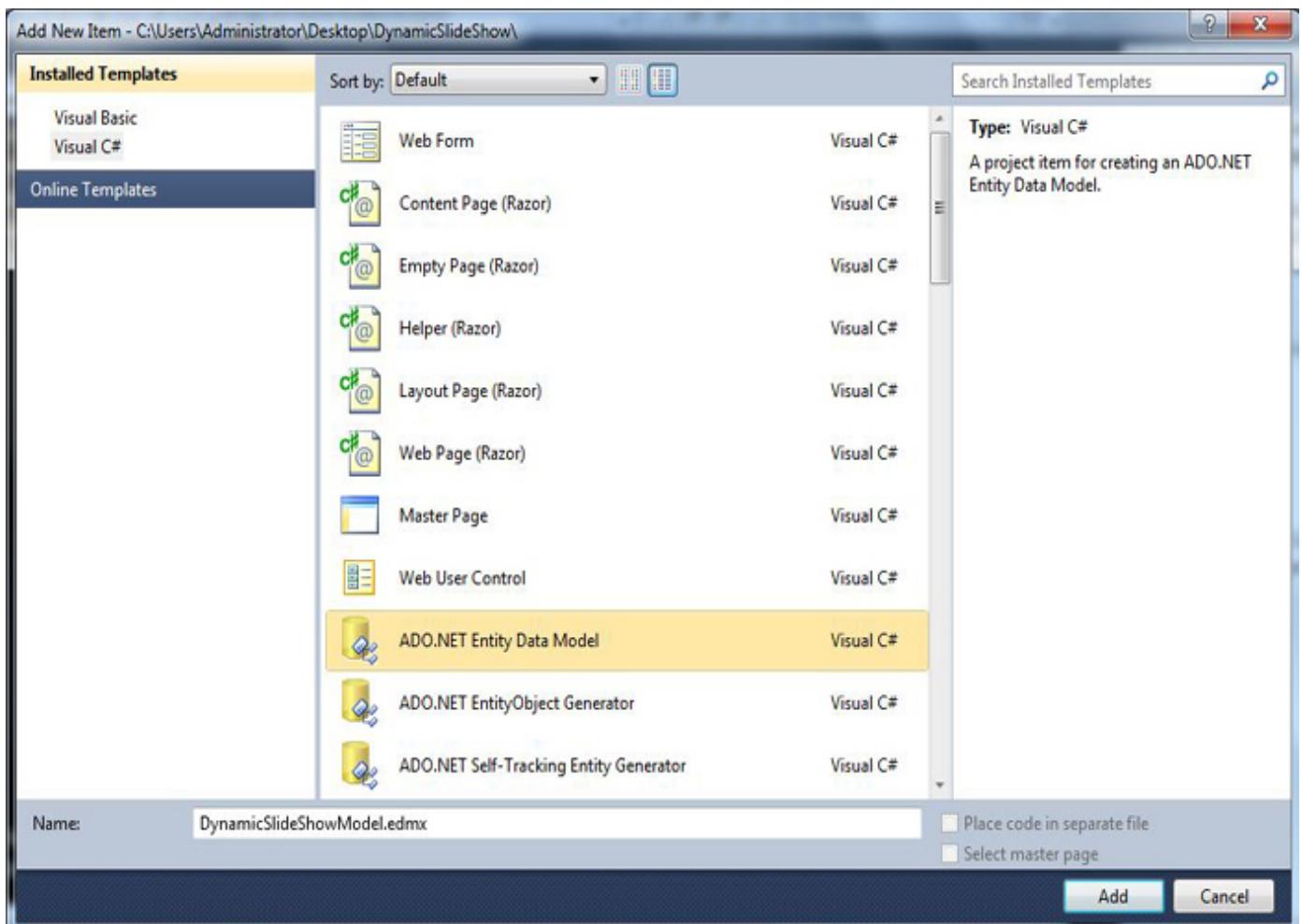
	Id	int	
	PicturePath	nvarchar(50)	
	PictureText	nvarchar(200)	

گام بعدی ایجاد یک پروژه Asp.Net با زبان C# و اضافه کردن فایل‌های پلاگین به پروژه و ایجاد یک Handler برای بازیابی داده‌ها از DB است. ساختار Solution ما باید به صورت زیر باشد

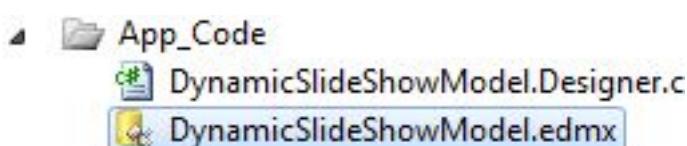


برای اینکه بتوانیم با DB ارتباط برقرار کنیم از EF استفاده می‌کنیم به همین منظور ابتدا یک Model به نام DynamicSlideShowModel ایجاد می‌کنیم

نحوه ایجاد یک اسلایدشو به صورت داینامیک



در گام بعد بر روی GenerateFromDatabase انتخاب کرده و بر روی دکمه Next کلیک می‌کنیم و در مراحل بعد ابتدا DB مربوط به مثال (DynamicSlideShow) و جدول آن را انتخاب می‌کنیم . حال یک Model به درون پوشه App_Code اضافه شده است



در ادامه برای واکشی رکوردهای موجود در جدول Pictures کدهای زیر را درون Handler می‌نویسیم

```
var ctx = new DynamicSlideShowEntities();
var list = ctx.Pictures.ToList();
string str = JsonConvert.SerializeObject(list);
context.Response.Write(str);
```

در این کدها تنها نکته‌ای که احتیاج به توضیح دارد این است که ابتدا یک لیست از رکوردها را از جدول Pictures واکشی می‌کنیم و برای پاس دادن به کلاینت ما آن‌ها را به فرمت Json تبدیل کرده ایم که برای تبدیل از کتابخانه آماده [Newtonsoft.Json.dll](#) استفاده کرده ایم

حال باید با استفاده از `jQuery` کدهای درون `Handler` را اجرا کنیم؛ برای همین منظور یک صفحه با نام `default.aspx` ایجاد کرده و کدهای زیر را درون آن می‌نویسیم

```
<head runat="server">
    <title>Dynamic SlideShow</title>
    <link href="CSS/orbit-1.2.3.css" rel="stylesheet" type="text/css" />
    <script src="Script/jquery-1.7.1.min.js" type="text/javascript"></script>
    <script src="Script/jquery.orbit-1.2.3.min.js" type="text/javascript"></script>
    <script type="text/javascript">
        $(function () {
            $.ajax({
                url: "Handler.ashx",
                contentType: "application/json; charset=utf-8",
                success: function (data) {
                    $.each(data, function (i, b) {
                        var str = '';
                        $("#featured").append(str);
                    });
                    $('#featured').orbit();
                },
                dataType: "json"
            });
        });
    </script>
</head>
<body>
    <form id="form1" runat="server">
        <div id="featured">
        </div>
    </form>
</body>
```

در اینجا ابتدا با استفاده از متدهای `jQuery` کتابخانه `Handler` را اجرا کرده و به ازای هر المان موجود در جدول یک تگ `img` به صفحه اضافه می‌کنیم.

نظرات خوانندگان

نویسنده: مسعود زیانی
تاریخ: ۱۳۹۱/۰۴/۱۷ ۱۰:۴۶

خیلی عالی بود ممنون

نویسنده: میثم
تاریخ: ۱۳۹۱/۰۴/۱۹ ۱۳:۳۱

مرسی خیلی عالی بود ، این عکس‌ها یکی یکی از سمت سرور اضافه می‌شوند یا همه با هم ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۹ ۱۳:۴۶

کاری که در اینجا انجام شده ساخت `div` زیر به صورت پویا است (بر اساس لیست تصاویر دریافت شده از سرور):

```
<div id="featured">
  
  
  
</div>
```

و سپس فراخوانی متده زیر روی آن:

```
$('#featured').orbit();
```

نویسنده: صادق
تاریخ: ۱۳۹۱/۰۶/۱۰ ۱۳:۱

با سلام

من کدهای بالا رو همه رو درست انجام دادم ولی در فایل `handler` متاسفانه `ToList()`; نمیشه. به فایل ضمیمه دقت کنید. با تشکر

نحوه ایجاد یک اسلایدشو به صورت داینامیک

نحوه ایجاد یک اسلایدشو به صورت داینامیک

دانلود کاربرد این پروژه برای آوردن پنج نمایه را

نام کاربری
رمز عبور
نام
نام خانوادگی
کدمل
شماره شناسنامه
رشنده تفصیلی
شماره تلفن
تلفن همراه
پست الکترونیک
استان
نشانی
فایل ضمیمه

شنبه تفصیلی

دوره های ارائه شده

درخواست های در حال انتظار

دوره های در حال برگزاری

دوره های ارائه شده

درخواست های ارائه شده

دوره های ارائه شده

عنوان درزوه آموزشی

ساعت درگذاری

روز درگذاری

ارسال

فایل ضمیمه

رسانید

نامه بندی

ردیف

عنوان جزو

مدرس

info@school.net

021-35719967

پست الکترونیک

موضوع پیام

ارسال پیام

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۱۰ ۱۳:۵۶

ذکر `System.Linq` را فراموش کردید.

نویسنده: صادق
تاریخ: ۱۳۹۱/۰۶/۱۰ ۱۳:۲۷

ممnonم از شما استاد عزیز

نویسنده: صابر فتح الله
تاریخ: ۱۰:۱۴ ۱۳۹۱/۰۶/۲۵

یه سوال واسه من پیش او مد مهندس
ممکنه ما ۱۰۰ تصویر داشته باشیم که بالطبع لود همه آنها یکباره درست نیست
آیا میشه کاری کرد که در هر درخواست که اسلایدشو تعویض میشه تصویر دینامیک لود کرد یا مثلًا هر دفعه سه تصویر لود کرد و
به پلاگین اضافه کرد تا نمایش بد؟
چون اینجوری همه تصاویر به یکباره لود شده و سرعت خیلی پایین میباشد

نویسنده: وحید نصیری
تاریخ: ۱۲:۲۱ ۱۳۹۱/۰۶/۲۵

میتونید از ایدههای افزونههایی تحت عنوان «[jQuery loading images on demand](#)» استفاده کنید. مثلا: (^)

نویسنده: حمید خرازان
تاریخ: ۲۲:۵۷ ۱۳۹۲/۰۱/۰۳

سلام، مهندس جان من وقتی از این کد استفاده میکنم فقط رکورد اول رو واسم میاره و بقیه رو undefined میدونید دلیلش چیه؟

نویسنده: مهدی آزادی
تاریخ: ۱۰:۳ ۱۳۹۲/۰۱/۱۸

عالی بود.

پاینده باشین.

نویسنده: پویا امینی
تاریخ: ۱۶:۳۴ ۱۳۹۲/۰۱/۱۸

دوست عزیز باید کدون رو ببینم و نظر بدم.

نویسنده: Abolfazl
تاریخ: ۱۲:۲۸ ۱۳۹۲/۰۲/۰۳

با سلام ..

من همه چیز را همان طوری که شما گفتید انجام دادم ولی نمیدونم چرا اجرا نی کنه .. بعد از لود صفحه یک لحظه نشون میده بعد فقط فلش که برای حرکت تصویر است نشون میده و اون هم در گوشه بالای صفحه :
این هم کد :::

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Web;

public class GallerySite
{
    int iD;
    public int ID
```

```

{
    get { return iD; }
    set { iD = value; }
}
string imagePath;
public string ImagePath
{
    get { return imagePath; }
    set { imagePath = value; }
}
string imageText;
public string ImageText
{
    get { return imageText; }
    set { imageText = value; }
}
public List<GallerySite> GetImage()
{
    List<GallerySite> _Gallery = new List<GallerySite>();
    SqlConnection cnn = new SqlConnection("Data Source=.;Initial Catalog=RoshanZamirDataBase;Integrated Security=True");
    cnn.Open();
    SqlCommand cmd = new SqlCommand("Select * From Gallery", cnn);
    SqlDataReader datareadfewr = cmd.ExecuteReader();
    if (datareadfewr.HasRows)
    {
        while (datareadfewr.Read())
        {
            _Gallery.Add(new GallerySite()
            {
                ID = Convert.ToInt32(datareadfewr["ID"]),
                ImagePath = (string)datareadfewr["ImagePath"],
                ImageText=(string)datareadfewr["ImageText"]
            });
        }
    }
    return _Gallery;
}
}
.....
.....

```

```

<link href="orbit-1.2.3.css" rel="stylesheet" />
<script src="jquery-1.8.3.min.js"></script>
<script src="jquery.orbit-1.2.3.min.js"></script>
<script type="text/javascript">
$(function () {
    $.ajax({
        url: "Handler.ashx",
        contentType: "application/json; charset=utf-8",
        success: function (data) {
            $.each(data, function (i, b) {
                var str = '';
                $("#featured").append(str);
            });
            $('#featured').orbit();
        },
        dataType: "json"
    });
});
</script>
.....
.....

```

Handler :

```

public class Handler : IHttpHandler {
    public void ProcessRequest (HttpContext context) {

```

نحوه ایجاد یک اسلایدشو به صورت داینامیک

```
var _Gallery = new GallerySite();
var List = _Gallery.GetImage();
string str = JsonConvert.SerializeObject(List);
context.Response.Write(str);
}
public bool IsReusable {
    get {
        return false;
    }
}
```

نویسنده: محسن خان
تاریخ: ۱۳:۱۴ ۱۳۹۲/۰۲/۰۳

با نحوه دیباگ اسکریپت‌های jQuery آشنا هستید؟ افزونه [فایرباگ](#) رو نصب کنید. بعد مراجعه کنید به قسمت مثل network آن. بعد مشاهده کنید، آیا چیزی رد و بدل شده؟ آیا در کنسول اسکریپت‌های مرورگر خطای لاغ شده. آیا مثل CSS‌های سایت روی این افزونه تاثیر داشته. مرورگر کروم هم در قسمت developer tools آن شبیه به فایرباگ رو دارد. دکمه F12 آن.

نویسنده: Abolfazl
تاریخ: ۱۴:۵۳ ۱۳۹۲/۰۲/۰۳

من نتونستم درستش کنم.. یه راه دیگه وجود نداره؟

نویسنده: وحید نصیری
تاریخ: ۱۷:۳۷ ۱۳۹۲/۰۲/۰۳

کد رو دیباگ کن:

```
var str = '';
```

بین اینجا نهایتا چه رشته‌ای تشکیل میشه.
آیا src درستی به مرورگر ارائه شده؟ مثلا آیا jpg مثلا \path\data\image\01.jpg است؟ یا اینکه به نحو صحیحی به صورت یک مسیر نسبی از ریشه سایت مانند siteimage/01.jpg مقدار دهی شده؟

نویسنده: مهرداد
تاریخ: ۱۸:۳۹ ۱۳۹۳/۰۷/۰۳

سلام؛ من از روش بالا در mvc استفاده کردم می‌خواستم ببینم آیا کار صحیحی انجام دادم یا خیر؟
در مورد Http Module و Http Handler هم اطلاعات کمی دارم و آیا استفاده از این‌ها در mvc کار صحیحی است یا خیر؟

نویسنده: وحید نصیری
تاریخ: ۱۹:۲ ۱۳۹۳/۰۷/۰۳

نیازی نیست. مطلب فوق در مورد وب فرم‌ها نوشته شده. برای انتقال آن به MVC بجای فایل ashx یک اکشن متده با خروجی Json خواهد داشت.

یک ادیتور آنلاین را تصور کنید که کاربران در قسمت ارسال تصویر آن قرار است فقط فایل‌های jpg, png و gif ارسال کنند و نه ASP.NET و موارد مشابه. در اینجا برای محدود کردن نوع فایل‌های آپلود شده می‌توان از فیلترهای سفارشی test.aspx کمک گرفت:

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Web.Mvc;

namespace SecurityModule
{
    public class AllowUploadSpecialFilesOnlyAttribute : ActionFilterAttribute
    {
        readonly List<string> _toFilter = new List<string>();
        readonly string _extensionsWhiteList;
        public AllowUploadSpecialFilesOnlyAttribute(string extensionsWhiteList)
        {
            if (string.IsNullOrWhiteSpace(extensionsWhiteList))
                throw new ArgumentNullException("extensionsWhiteList");

            _extensionsWhiteList = extensionsWhiteList;
            var extensions = extensionsWhiteList.Split(',');
            foreach (var ext in extensions.Where(ext => !string.IsNullOrWhiteSpace(ext)))
            {
                _toFilter.Add(ext.ToLowerInvariant().Trim());
            }
        }

        bool canUpload(string fileName)
        {
            if (string.IsNullOrWhiteSpace(fileName)) return false;

            var ext = Path.GetExtension(fileName.ToLowerInvariant());
            return _toFilter.Contains(ext);
        }

        public override void OnActionExecuting(ActionExecutingContext filterContext)
        {
            var files = filterContext.HttpContext.Request.Files;
            foreach (string file in files)
            {
                var postedFile = files[file];
                if (postedFile == null || postedFile.ContentLength == 0) continue;

                if (!canUpload(postedFile.FileName))
                    throw new InvalidOperationException(
                        string.Format("You are not allowed to upload {0} file. Please upload only these
files: {1}.",
                        Path.GetFileName(postedFile.FileName),
                        _extensionsWhiteList));
            }

            base.OnActionExecuting(filterContext);
        }
    }
}

```

توضیحات کدهای فوق:

برای تهیه فیلتر محدود سازی نوع فایل‌های قابل ارسال به سرور، با ارث بری از ActionFilterAttribute شروع خواهیم کرد. سپس با تحریف متد OnActionExecuting آن، توسط HttpContext.Request.Files می‌توان به کلیه فایل‌های در حال ارسال به سرور در طی درخواست جاری، دسترسی یافت. به این ترتیب از طریق مقدار خاصیت postedFile.FileName می‌توان به پسوند فایل در حال ارسال رسید و بر این اساس امکان

ارسال فایل‌های غیرمجاز را در نیمه راه با صدور یک استثناء سلب کرد.

برای استفاده از این فیلتر سفارشی تهیه شده نیز می‌توان به نحو زیر عمل کرد:

```
[AllowUploadSpecialFilesOnly(".jpg,.gif,.png")]
public ActionResult ImageUpload(HttpPostedFileBase file)
```

در اینجا پسوند فایل‌های مجاز قابل ارسال، توسط یک کاما از هم جدا خواهند شد.

یک نکته تکمیلی:

اگر کاربر قرار است تنها تصویر ارسال کند، بررسی پسوند فایل لازم است اما کافی نیست. برای این منظور می‌توان از کلاس `Image` واقع شده در فضای نام `System.Drawing` نیز کمک گرفت:

```
public static bool IsImageFile(HttpPostedFileBase photoFile)
{
    using (var img = Image.FromStream(photoFile.InputStream))
    {
        return img.Width > 0;
    }
}
```

در اینجا اگر فایل ارسالی تصویر نباشد، به صورت خودکار یک استثناء صادر خواهد شد.

نظرات خوانندگان

نویسنده: حسین مرادی نیا
تاریخ: ۱۴:۵ ۱۳۹۱/۰۴/۱۷

سلام

همانطور که گفتید چک کردن پسوند فایل الزامی است ولی کافی نیست. برای همین از کلاس `Image` استفاده کردید که اگر فایل ارسال شده یک فایل تصویری نبود استثنای صادر شود. حالا برای فایل‌ها با فرمتهای دیگه چیکار کنیم؟ (مثلًا ممکن است بخواهیم به کاربر اجازه ارسال فایل‌های `zip` و یا `rar` بدھیم و مواردی از این دست).

نویسنده: وحید نصیری
تاریخ: ۱۴:۱۶ ۱۳۹۱/۰۴/۱۷

برای بررسی محتوای خاص، نیاز به `parser` مخصوص این نوع فایل‌ها است تا بتواند بررسی کند محتوای دریافتی معتبر است یا نه و عموماً کسی محتوای این نوع فایل‌ها را بررسی نمی‌کند. بنابراین بررسی پسوند در اکثر موارد کافی است. مشکل این نیست که فایل `rar` واقعاً `rar` است یا نه. مشکل این است که این فایل ارسالی، قابلیت اجرا را از طریق فرآخوانی آدرس آن در مرورگر، نداشته باشد. بحث بررسی پسوند هم به همین دلیل است. فایل `aspx` را می‌شود از طریق فرآخوانی در مرورگر بر روی سرور اجرا کرد ولی فایلی با پسوند `rar` این قابلیت را ندارد.

نویسنده: شاهین کیاست
تاریخ: ۲:۱ ۱۳۹۱/۰۵/۰۹

ممnon ، استثنای که صادر می‌شود را چطور میشه ؟ در بدنی `Action` مربوطه مدیریت کرد ؟

نویسنده: وحید نصیری
تاریخ: ۸:۲۳ ۱۳۹۱/۰۵/۰۹

لازم نیست مدیریت کنید. هدف این بوده که نظم جاری را تغییر دهد. این استثناء توسط `ELmah` دریافت و ثبت خواهد شد. همچنین کاربر به یکی از صفحات پیش فرض خطای برنامه هدایت می‌شود و متوجه خواهد شد که خطایی رخداده است. ولی در کل می‌شود `IExceptionFilter` را نیز پیاده سازی و مدیریت کرد:

```
public class CustomFilter : FilterAttribute, IExceptionFilter
{
    public void OnException(ExceptionContext filterContext)
    {
```

نویسنده: رضوی
تاریخ: ۸:۵۳ ۱۳۹۱/۰۵/۲۸

سلام

چند نکته تکمیلی برای امنیت بیشتر

- ۱- بهتر است که فایل روی دیتابیس ذخیره شود، در غیر اینصورت نام فایل و پسوند توسط سیستم تعیین شود.
- ۲- نام و پسوند ارسالی در نظر گرفته نشود.
- ۳- با اندازه گیری سایز فایل `GIF` نمیتوان به معتبر بودن آن اطمینان داشت.

<http://www.pyxsoft.com/en/hacking-hackers.html>
<http://www.exploit-db.com/exploits/16181>

نوبنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۰۷ ۱۱:۱۶

یک نکته‌ی تکمیلی
نحوه‌ی فقط خواندنی کردنی دسترسی به فایل‌های یک پوشه (گرفتن دسترسی اجرا)

```
<location path="upload">
  <system.webServer>
    <handlers accessPolicy="Read" />
  </system.webServer>
</location>
```

پیشیازها:

- نحوه ارسال فایل به سرور توسط ASP.NET MVC
- نحوه اعتبار سنجی سمت سرور ارسال فایل‌ها

در C# برای آپلود فایل‌ها عموماً عنوان می‌شود که از تگ input به نحو زیر استفاده شود:

```
<input type="file" name="file" />
```

مشکلی که با این روش وجود دارد، عدم فعال شدن اعتبار سنجی سمت کاربر در حد مثلاً «لطفاً یک فایل را انتخاب کنید» است. برای فعال سازی آن می‌توان از همان روش unobtrusive معرفی شده در ASP.NET MVC 3 به نحو زیر به صورت دستی استفاده کرد:

```
<input type="file" data-val="true" data-val-required="لطفاً فایلی را انتخاب کنید" name="file" />  
@Html.ValidationMessage("file")
```

در اینجا ویژگی‌های data-val و data-val-required به صورت خودکار اعتبار سنجی سمت کاربر را فعال خواهند کرد (شبیه به required field validator عمل می‌کنند).

از این نکته خصوصاً در طراحی `html helper`‌های سفارشی نیز می‌توان استفاده کرد. برای مثال فرض کنید مشغول طراحی یک کنترل captcha هستید. قسمتی از آن مربوط به دریافت ورودی از کاربر است. به علاوه، پیغام خطایی هم که باید نمایش داده شود نیز باید توسط کاربر قابلیت سفارشی شدن را داشته باشد (و نمی‌توان آن را توسط یک attribute به سادگی به یک مدل خاص انتساب داد).

در این حالت تنظیم data-val و data-val-required به کمک `data-val-attribute` پارامتر `htmlAttributes` می‌سر نیست (چون بین حروف آن `dash` وجود دارد) و باید از `overload` و نوع `dictionary` دار آن به نحو زیر استفاده کرد:

```
htmlHelper.TextBox("CaptchaInputText", string.Empty,  
    new Dictionary<string, object>  
    {  
        { "data-val", "true"},  
        { "data-val-required", validationErrorMessage},  
    })
```

نظرات خوانندگان

نوبنده: مولانا
تاریخ: ۱۳۹۲/۰۲/۱۷ ۱۷:۳۰

با سلام.

آیا امکان آپلود فایل بوسیله `ajax` نیز وجود دارد. و یا این روش معایبی دارد؟ با تشکر.

نوبنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۱۷ ۱۸:۵۴

بله. مرورگرهایی که Level 2 XMLHttpRequest را پشتیبانی کنند ([از ۱۰ IE به بعد مثلًا](#)), امکان Ajax upload توکار به همراه گزارش درصد آپلود را بدون نیاز به فلاش یا سیلورلایت، دارند. [یک نمونه پیاده سازی آن](#)

باسلام:

هدف این مقاله بیشتر آشنایی با IHttpModule در قالب یک پروژه می‌باشد.
 قصد دارم در این مقاله یک روش برای مسدود کردن IP هایی که به هر روشه در سایت شما اقداماتی غیر عادی انجام دادن و شما قصد دارید سایت شما به اونها نمایش داده نشے بیان کنم.
 دقت کنید که شما میتوانید با تشخیص کاربر مختلف مثل کاربری که بیش از 5 بار اقدام به وارد کردن نام کاربری و رمز عبور کرده کنید و IP کاربر رو در جایی مانند DataBase ذخیره کنید و یک زمان برآش ثبت کنید و تا 15 دقیقه بعد از اون دسترسی به سایت را ازش بگیرید.

برای شروع من کلاس مربوطه رو با نام IPBlockModule در پوشه App_Code میسازم:

```

using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public class IPBlockModule : IHttpModule
{
    public IPBlockModule()
    {
        // TODO: Add constructor logic here
    }
    public void Dispose()
    {
        //Dispose
    }

    public void Init(HttpContext context)
    {
        context.BeginRequest += new EventHandler(Application_BeginRequest);
    }

    private void Application_BeginRequest(object source, EventArgs e)
    {
        HttpContext context = ((HttpApplication)source).Context;
        string ipAddress = context.Request.UserHostAddress;
        if (IsBlockedIpAddress(ipAddress))
        {
            context.Response.StatusCode = 403;
            context.Response.Write("Forbidden : The server understood the request, but It is refusing
to fulfill it.");
        }
    }

    private bool IsBlockedIpAddress(string ipAddress)
    {
        //Here I have stored Ip addresses in String[]. you can also Store in database.
        string[] IPs = {
            "117.196.35.121",
            "117.196.35.122",
            "117.196.35.123",
            "117.196.35.124",
            "127.0.0.1"
        };

        foreach(string IP in IPs)
        {
            if(IP == ipAddress)
                return true;
        }
        return false;
    }
}

```

و در فایل web.config این کلاس را اضافه میکنیم :

```
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.0"/>
    <httpModules>
      <add name="IPBlockModule" type="IPBlockModule"/>
    </httpModules>
  </system.web>
</configuration>
```

میتوانید متغیر آرایه ای IPs را از پایگاه داده بخونید و کوئری رو با شرط مدت زمان سپری شده که معمولاً 15 دقیقه هست بگیرید.

موفق باشید.

نظرات خوانندگان

نویسنده: علیرضا اسمرا م
تاریخ: ۲۲:۳۰ ۱۳۹۱/۰۴/۲۵

سلام. با تشکر از شما.
بسیار مطلب خوب و کاربردی بود.
در همین رابطه مطلب خوبی در [اینجا](#) خوانده بودم که برای مطالعه بیشتر پیشنهاد می‌کنم.
همچنین متدهای `IsBlockedIpAddress` را می‌توان با عبارت LINQ زیر جایگزین کرد:

```
private static bool IsBlockedIpAddress(string ipAddress)
{
    string[] ips = {
        "117.196.35.121",
        "117.196.35.122",
        "117.196.35.123",
        "117.196.35.124",
        "127.0.0.1"
    };

    return ips.Any(ip => ip == ipAddress);
}
```

نویسنده: Nima
تاریخ: ۲۲:۳۵ ۱۳۹۱/۰۴/۲۵

سلام دوست عزیز

آیا بلاک کردن ip کار دقیقی است؟ از این نظر که دهها نفر در اینترنت ممکن است با یک ip به سایت ما وصل بشن و این کار باعث میشه همه اونا بلاک بشن

نویسنده: علیرضا اسمرا م
تاریخ: ۲۲:۴۳ ۱۳۹۱/۰۴/۲۵

حدودیت زمانی 15 دقیقه را برای همین منظور اعمال می‌کنند.

نویسنده: مجید مقصودی پور
تاریخ: ۱۲:۱۶ ۱۳۹۱/۰۴/۲۶

باسلام:
میشه گفت یک راه برای جلوگیری از هکرهای میباشد هکرهای میتونه هم انسان باشه و یا هم یک برنامه کاربردی (робوت) که یک عملیات رو در یک بازه زمانی زیاد ترست میکنه.
که این بحث در این مورد در [اینجا](#) نمیگنجه و بیشتر من در این مقاله هدفم استفاده از HTTP Module بود.
موفق باشید.

نویسنده: مجید مقصودی پور
تاریخ: ۱۲:۱۸ ۱۳۹۱/۰۴/۲۶

باسلام:
باتشکر از شما دوست گرامی.
یک نکته رو توجه داشته باشید که راهها و روش‌های زیادی برای پیاده سازی وجود داره من خودم سعی میکنم ساده‌ترین روش رو برای پیاده سازی متدها انتخاب کنم و هر برنامه نویسی میتونه اونو بسته به معماری که ازش استفاده میکنه تغییر بده.
راههای رسیدن به خدا زیاده.

نوبنده: وحید نصیری
تاریخ: ۱۶:۵۹ ۱۳۹۱/۰۴/۲۶

از این نمونه‌ها من در این سایت زیاد دارم. کسانی که مرتباً روزی چند بار سایت رو زیر حمله می‌گیرند.

Banned IP: 46.165.199.150 Details...	07/16/2012	09:11	ق.ظ
Banned IP: 46.165.199.150 Details...	07/16/2012	09:11	ق.ظ
Banned IP: 46.165.199.150 Details...	07/16/2012	09:11	ق.ظ
Banned IP: 46.165.199.150 Details...	07/16/2012	09:11	ق.ظ
Banned IP: 46.165.199.150 Details...	07/16/2012	09:11	ق.ظ
Banned IP: 46.165.199.150 Details...	07/16/2012	09:11	ق.ظ
Banned IP: 46.165.199.150 Details...	07/16/2012	09:11	ق.ظ
Banned IP: 46.165.199.150 Details...	07/16/2012	09:11	ق.ظ
Banned IP: 46.165.199.150 Details...	07/16/2012	09:11	ق.ظ
Banned IP: 46.165.199.150 Details...	07/16/2012	09:11	ق.ظ
Banned IP: 46.165.199.150 Details...	07/16/2012	09:11	ق.ظ
Banned IP: 46.165.199.150 Details...	07/16/2012	09:11	ق.ظ

Error	User	Date	Time
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب
Banned IP: 31.29.56.1 Details...		07/16/2012	02:46 ب

نویسنده: مجید مقصودی پور
تاریخ: ۱۳۹۱/۰۴/۲۶ ۱۷:۲

شما برای جلوگیری از این IP‌ها کاری میکنید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۲۶ ۱۷:۵

بله. یک فیلتر آنتی داس درست کردم که شبانه روز دسترسی این نوع مزاحمها را حذف میکنه. تا الان هر روز به طور میانگین حداقل 2 بار در زمان‌های مختلف به این سایت حمله شده.

نویسنده: مجید مقصودی پور
تاریخ: ۱۳۹۱/۰۴/۲۶ ۱۷:۷

اگر در قالب یک مقاله ارائه بدید خیلی مفید واقع میشه.
باتشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۲۷ ۱۰:۲۲

تقریبا [بر همین اساس](#) پیاده سازی شده. اطلاعات در کش سیستم ذخیره میشون و به این ترتیب IP‌های مهاجم را میشه تشخیص داد.

نویسنده: امیر
تاریخ: ۱۳۹۱/۰۴/۲۷ ۱۴:۲۶

با درود و عرض ادب

استاد نصیری

سایت ما هر چند وقت یک بار با IP‌های مختلفی هک میشود

همه جور محدودیت و injection هم اعمال کردم.

به نظرتون این مشکل رو چی کار باید کرد؟

ممnon از راهنماییتون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۲۷ ۱۴:۲۹

نمیشه همینطوری ندیده اظهار نظر کرد. نیاز به [code review](#) داره.

نویسنده: صابر فتح اللهی
تاریخ: ۱۳۹۱/۰۴/۲۷ ۱۵:۱۹

سلام

البته توی این [آدرس](#) به فارسی توضیح داده شده همون روش

نوبنده: صابر فتح اللهی
تاریخ: ۱۳۹۱/۰۵/۱۶ ۱۰:۲۳

مهند نصیری من IP هایی که به نحوی دارن بازدید اضافی انجام میدن و مشکوک هستن طبق همین روش اونارو توی لیست سیاه میدارم و دسترسی اونهارو به سایت قطع می‌کنم آیا روش درستیه؟
یعنی دیگه نمی‌توان بیان توی سایت اما این امکان گذاشتم که بعداً اونهارو از لیست سیاه در بیارم

نوبنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۶ ۱۳:۵۷

نه. IP‌های کاربران در ایران عموماً ثابت نیست. نصف روز بسته شود کافی است.

بر مبنای پیاده سازی متداولی که در n هزار سایت اینترنتی می‌توان یافت، نحوه کار با جستجوگر لوسین حدوداً به این شکل است:

```
var directory = FSDirectory.Open(new DirectoryInfo(Environment.CurrentDirectory + "\\LuceneIndex"));
using (var searcher = new IndexSearcher(directory, readOnly: true))
{
    //do something ...
    searcher.Close();
    directory.Close();
}
```

و ... اینکار به این شکل غلط است!

مطابق [مستندات رسمی](#) لوسین، این کتابخانه thread-safe است. به این معنا که در آن واحد چندین و چند کاربر می‌توانند از یک وله از شیوه‌های Reader و Searcher استفاده کنند و نباید به ازای هر جستجو، یکبار این اشیاء را ایجاد و تخریب کرد. البته در اینجا تنها یک Writer در آن واحد می‌تواند مشغول به کار باشد.

مشکلاتی که به همراه باز و بسته کردن بیش از حد IndexSearcher وجود دارد، مصرف بالای حافظه است (به ازای هر کاربر مراجعه کننده، یکبار باید ایندکس‌ها در حافظه بارگذاری شوند) و همچنین تاخیر اولیه این بارگذاری و کندی آنرا نیز باید مدنظر داشت.

نتیجه گیری:

برای کار با جستجوگر لوسین نیاز است از الگوی [Singleton](#) استفاده شود و تنها یک وله از این اشیاء بین تردهای مختلف به اشتراک گذاشته شود.

نظرات خوانندگان

نوبنده: حسین غلامی
تاریخ: ۱۳۹۱/۱۰/۱۱ ۱۲:۲۹

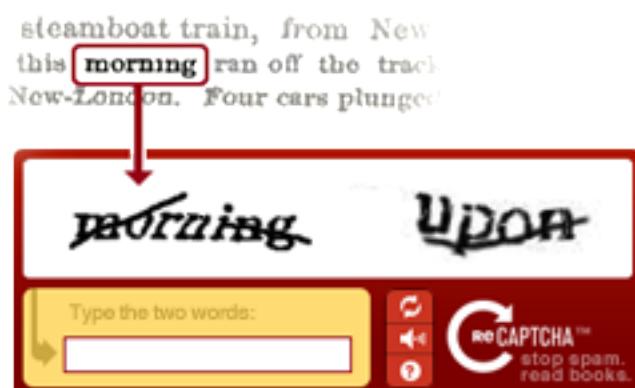
آقای نصیری میشه نمونه ای رو با استفاده از این الگو ، مثال بزنید؟

نوبنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۱۱ ۱۰:۹

مراجعه کنید به مثال [Auto Complete](#).

CAPTCHA جهت تشخیص انسان در وب استفاده می‌شود. در حقیقت می‌توان CAPTCHA را مانع مایین منابع یک سایت و روبات‌های محرب دانست. گوگل یک CAPTCHA رایگان جهت استفاده توسعه دهنده‌گان 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 CAPTCHA‌ها به سادگی قابل شکستن هستند!



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

1. 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.
2. دانلود [reCAPTCHA ASP.NET Library](#). این کتابخانه شامل کنترل reCAPTCHA است. پس از دریافت، **Recaptcha.dll** را توسط **Add Reference** به پروژه خود اضافه کنید.
3. 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 متod `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

نوبتند: زهرا
تاریخ: ۹:۱۴ ۱۳۹۱/۰۴/۳۱

با سلام

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

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

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

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

اگر دسترسی به گوگل گد ندارید، می‌شود از طریق [NuGet](#) نیز آن را دریافت کرد.

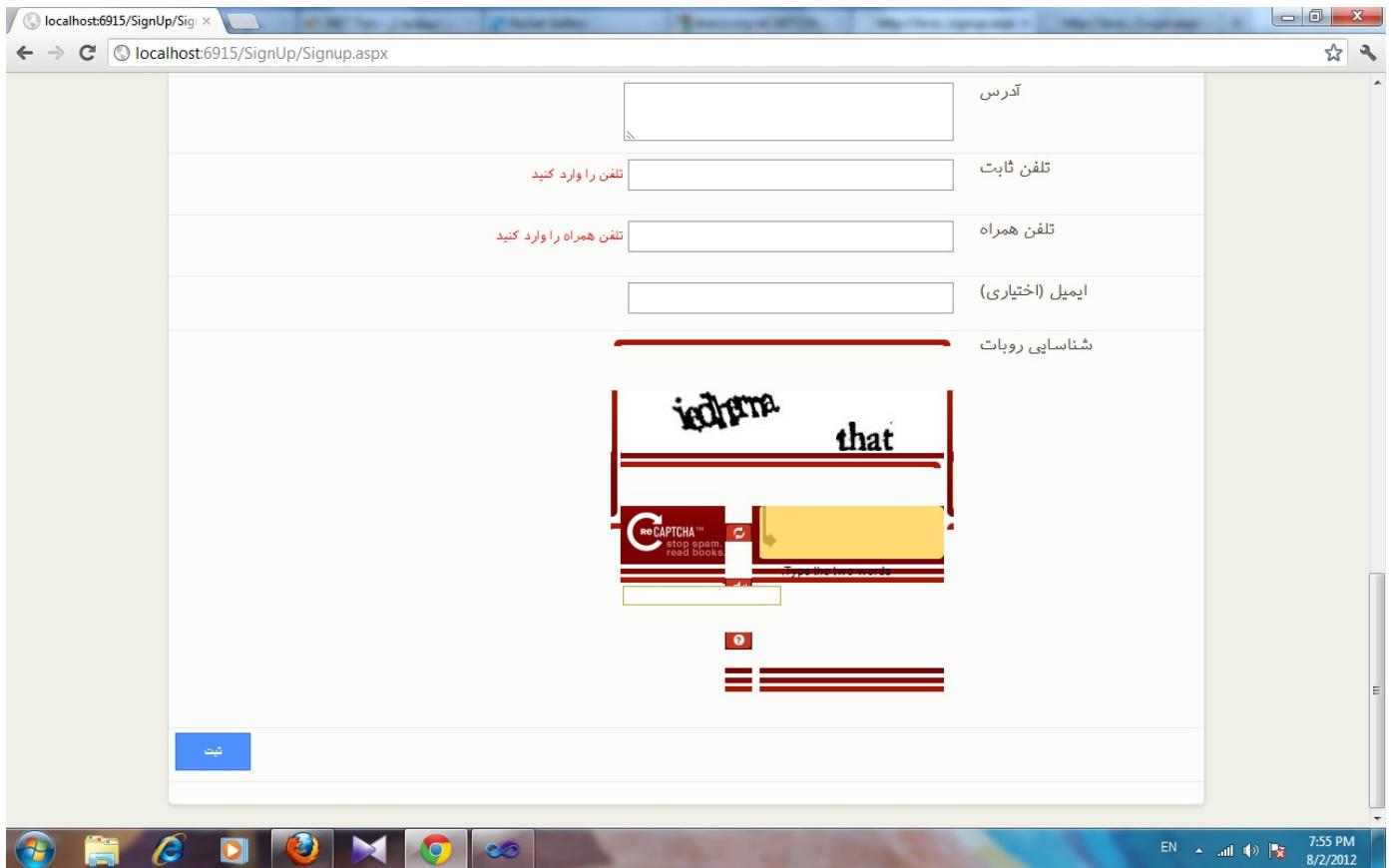
نوبتند: علیرضا اسمرا
تاریخ: ۹:۴۸ ۱۳۹۱/۰۴/۳۱

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 مقاطعه زمانی مختلف از این سرویس تا به حال بهره گرفته اند. در این لیست می‌توان به سایتهاز بزرگی چون Twitter، eBay و Facebook اشاره کرد. جای نگرانی نیست.

نوبتند: 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 ؟

استفاده از reCAPTCHA در ASP.NET



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

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 ای 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: ۱۳:۴۳ ۱۳۹۱/۰۵/۱۳

ممnon از راهنماییتون. مشکلم حل شد مینویسم که اگه دوستام 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 مود نباید در مورد جایی که این کنترل قرار داره اعمال بشه. ممنون استاد نصیری.

نویسنده: حسین
تاریخ: ۲۲:۳۱ ۱۳۹۱/۰۵/۱۷

سلام
دوستان من نمیتونم فایل دل ال الشو دانلود کنم، تو رو خدا فایلشو بذارین منم بردارم. با اینکه تو لاگین جی میلم هستم ازم پسورد میخاد که هر چی میزنم میگه اشتباhe ... ! با اینکه من داخل جی میلم 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

بالاتر 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 بودید و 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(' خطا ! ');", true);
    }
    ScriptManager.RegisterStartupScript(this, GetType(), "showalert", "alert(' آمدید ! ');", 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 باشد، قسمت نمایش خطا اجرا میشنه.

یکی از مسائلی که در هنگام کار با کنترلهای داده‌ای نظیر `ListView`, `GridView` و ... با آن روبرو هستیم مسئله صفحه بندی می‌باشد و در بسیاری از موارد، کل اطلاعات در هر درخواست، بارگذاری می‌شود. در حالیکه روش بهینه به این صورت است که با توجه به `Index` و `PageSize` رکورد، می‌توان تعداد رکورد مورد نظر در همان صفحه را بارگذاری کرد، نه کل رکوردها را. در این مثال که از `Ef Code First` و الگوی `Unit Of Work` استفاده کرده ام، قصد نمایش اطلاعات در یک `ListView` و صفحه بندی آن را بصورت `chunk chunk` دارم. برای آشنایی بیشتر با الگوی `Unit Of Work` می‌توانید به مقاله [UOW](#) در همین سایت مراجعه کنید.

ابتدا یک کنترل `ListView` روی صفحه ایجاد می‌کنیم. برای آشنایی بیشتر با این کنترل و بررسی قابلیتهای آن می‌توان از این مقاله [معرفی کنترل‌های DataPager و ListView](#) استفاده کنید.

سپس با توجه به الگوی `Unit Of Work` از یک مدل ساده استفاده می‌کنیم. همچنین برای بارگذاری اطلاعات به صورت صفحه به صفحه، نیاز به داشتن `Index` رکورد و `PageSize`، جهت محاسبه تعداد رکورد مورد نیاز داریم.

```
public class User
{
    public Int64 UserId { get; set; }
    public String UserName { get; set; }
}
```

```
public interface IUserService
{
    int GetCustomerCount();
    List<User> GetCustomers(int StartIndex, int PageSize);
}
```

```
public class ImplUserService : IUserService
{
    IUnitOfWork _uow;
    IDbSet<User> _user;

    public ImplUserService(IUnitOfWork uow)
    {
        _uow = uow;
        _user = uow.Set<User>();
    }

    public int GetCustomerCount()
    {
        int totalCount = _user.ToList().Count;
        return totalCount;
    }

    public List<User> GetCustomers(int StartIndex, int PageSize)
    {
        return _user.OrderBy(i => i.UserId).Skip(StartIndex).Take(PageSize).ToList();
    }
}
```

در کدهای بالا متدهای `GetCustomerCount` تعداد کل رکوردهایی که باید واکشی شوند را مشخص می‌کند. همچنین می‌توان این تعداد را در `cache` یا `ViewState` ذخیره کرد تا در دفعات بعدی در صورت خالی نبودن مقدار `cache` با `ViewState`، نیاز به محاسبه مجدد `count` نداشته باشیم.

متدهای اصلی را انجام میدهد از دو پارامتر استفاده می‌کند: `PageIndex` نقطه شروع و `PageSize` تعداد رکورد مورد نظر. در اینجا از دستورات `Linq` استفاده شده و دستور `Skip` مشخص می‌کند از کدام شماره رکورد به بعد شروع به واکشی

واکشی اطلاعات به صورت chunk chunk (تکه تکه) و نمایش در ListView

نماید و دستور Take مشخص می‌کند که چه تعداد رکورد را واکشی نماید.

حالا به سراغ کدهای HTML می‌رویم. در آنجا علاوه بر ListView نیاز به DataPager جهت صفحه بندی و ObjectDataSource می‌باشد. کنترل بارگذاری اطلاعات به صورت chunk chunk داریم.

```
<asp:ListView ID="ListView1" runat="server" DataSourceID="ObjectDataSource1">
    <ItemTemplate>
        <tr style="background-color: #DCDCDC; color: #000000;">
            <td>
                <asp:Label ID="UserIdLabel" runat="server" Text='<%# Eval("UserId") %>' />
            </td>
            <td>
                <asp:Label ID="UserNameLabel" runat="server" Text='<%# Eval("UserName") %>' />
            </td>
        </tr>
    </ItemTemplate>

    <LayoutTemplate>
        <table runat="server">
            <tr runat="server">
                <td runat="server">
                    <table id="itemPlaceholderContainer" runat="server" border="1"
style="background-color: #FFFFFF; border-collapse: collapse; border-color: #999999; border-style: none;
border-width: 1px;">
                        <tr runat="server" style="background-color: #DCDCDC; color: #000000;">
                            <th runat="server">
                                UserId
                            </th>
                            <th runat="server">
                                UserName
                            </th>
                        </tr>
                        <tr id="itemPlaceholder" runat="server">
                            </tr>
                        </table>
                    </td>
                </tr>
                <tr runat="server">
                    <td runat="server" style="text-align: center; background-color: #CCCCCC;
color: #000000;">
                        <asp:DataPager ID="DataPager1" runat="server" PageSize="2">
                            <Fields>
                                <asp:NextPreviousPagerField ButtonType="Button"
ShowFirstPageButton="True" ShowNextPageButton="False"
                                    ShowPreviousPageButton="False" />
                                <asp:NumericPagerField />
                                <asp:NextPreviousPagerField ButtonType="Button"
ShowLastPageButton="True" ShowNextPageButton="False"
                                    ShowPreviousPageButton="False" />
                            </Fields>
                        </asp:DataPager>
                    </td>
                </tr>
            </table>
        </LayoutTemplate>
    </asp:ListView>
```

همانظور که مشاهده می‌کنید در DataPager مخصوص شده است. اما PageSize مقدار مشخص شده است.

```
<asp:ObjectDataSource ID="ObjectDataSource1" runat="server" EnablePaging="True"
SelectCountMethod="GetCustomerCount"
    SelectMethod="GetCustomers" TypeName="UserServiceImpl.UserService" EnableViewState="False"
    MaximumRowsParameterName="PageSize" StartRowIndexParameterName="StartIndex">
```

که فکر می‌کنم با توجه به متدهای تعریف شده در بالا، تعریف ObjectDataSource کاملاً گویا می‌باشد.

نکته مهم:

اگر الان برنامه را اجرا کنید با خطای No parameterless constructor defined for this object روبرو خواهد شد که جهت

حل این مشکل از کد زیر استفاده میکنیم.

```
protected void ObjectDataSource1_ObjectCreating(object sender, ObjectDataSourceEventArgs e)
{
    e.ObjectInstance = ObjectFactory.GetInstance<IUserService>();
}
```

در واقع نیاز است تا یک وله از کلاس مورد نظر را به `ObjectDataSource` معرفی کنیم.
حال با اجرای برنامه و `Trace` آن متوجه خواهید شد که با کلیک بر روی شماره صفحه، تنها به تعداد رکوردهای همان صفحه،
واکشی خواهیم داشت.

با تشکر از راهنمایی‌های آقای نصیری، امیدوارم این مقاله برای دوستان مفید باشد. منتظر نظرات دوستان هستم و اینکه چه جوری
بتوانیم اینکار رو با `jquery ajax` و به صورت سبکتر انجام بدیم.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۴:۲۵ ۱۳۹۱/۰۴/۳۰

یک نکته تکمیلی: در اینجا بهتر است جهت محاسبه تعداد رکوردها از متدها Count مستقیما بر روی dbSet استفاده کنید (نیازی بهToList ندارد).

نویسنده: ایمان اسلامی
تاریخ: ۱۴:۳۴ ۱۳۹۱/۰۴/۳۰

بله کاملاً درسته
ممnon مهندس.

نویسنده: علیرضا اسمراهم
تاریخ: ۱۵:۵۵ ۱۳۹۱/۰۴/۳۰

سلام و تشکر از شما بخاطر این مطلب. پیشنهاد می‌کنم جهت توسعه این ترفند (همانطور که اشاره کردید) نگاهی به [+](#) و [-](#) و [±](#) بیاندازید.
موفق باشید.

نویسنده: ایمان اسلامی
تاریخ: ۱۶:۳۰ ۱۳۹۱/۰۴/۳۰

با سلام و تشکر از شما بابت نظرتون
scrolling هم راه خوبیه برای اینکار که کاربردهای زیادی داره
ممnonم.

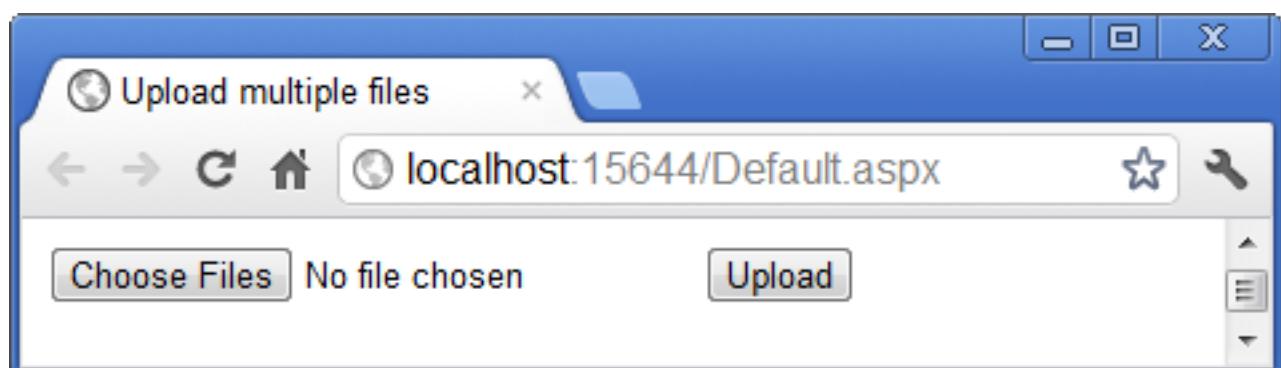
چندی پیش امکان بارگذاری چندین فایل بطور هم زمان روی سرور با استفاده از کنترل‌های Telerik یا DevExpress مهیا می‌شد. همچنین به کمک jQuery تکنیک‌های وجود داشت. اما در HTML5 می‌توان از تگ زیر استفاده کرد:

```
<input type="file" multiple="multiple" name="FileUpload1" id="FileUpload1" />
```

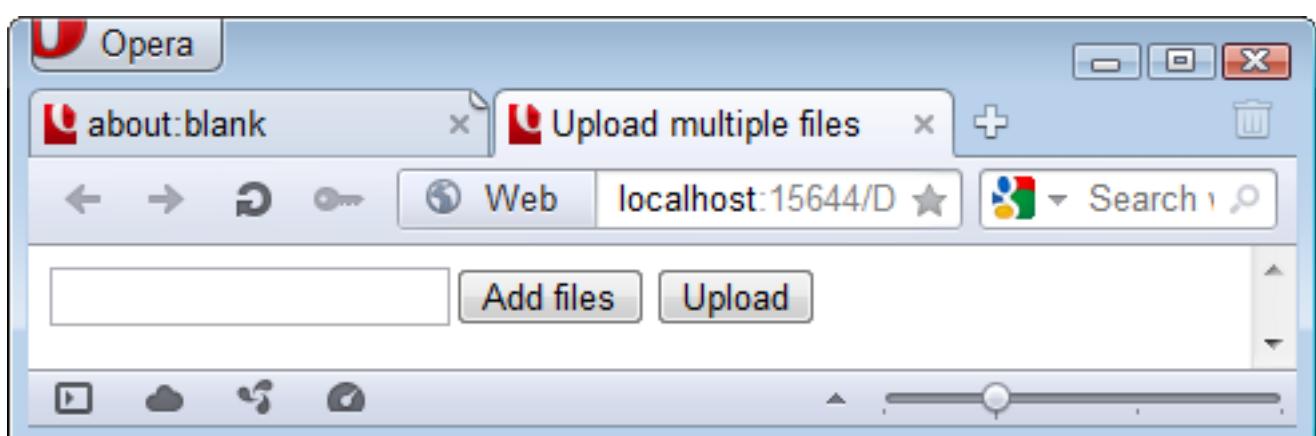
یکی از امکانات جدید ASP.NET4.5 سازگاری کنترل‌های سمت سرور با HTML5 است. از این رو به کنترل FileUpload خصوصیاتی اضافه شده است:

```
<asp:FileUpload ID="FileUpload1" runat="server" AllowMultiple="true" />
```

این کنترل در مرورگرهای متفاوت بصورت‌های مختلفی نمایش داده می‌شود. نمایش در مرورگر Chrome بصورت زیر خواهد بود:



: در Opera و



با این تفاسیر در سمت سرور، کار دشواری پیش روی ما نخواهد بود:

```
protected void Upload_Click(object sender, EventArgs e)
{
    if (FileUpload1.HasFiles)
    {
        string rootPath = Server.MapPath("~/App_Data/");
        foreach (HttpPostedFile file in FileUpload1.PostedFiles)
        {
            file.SaveAs(Path.Combine(rootPath, file.FileName));
            Label1.Text += String.Format("{0}<br />", file.FileName);
        }
    }
}
```

البته از آنجایی که هدف از این مطلب معرفی یکی از قابلیت‌های جدید HTML5 و ASP.NET4.5 است، کد بالا بسیار ساده نوشته شده است و فایل‌های ارسال شده به سرور را در پوشه App_Data و بدون در نظر گرفتن مسائل عرف درباره Upload فایل، ذخیره می‌کند.

رمزنگاری Connection String از طریق خط فرمان

عنوان:

سیروان عفیفی

نویسنده:

۱۹:۱۶ ۱۳۹۱/۰۵/۰۲

تاریخ:

www.dotnettips.info

آدرس:

ASP.Net, Security

گروهها:

برای رمز نگاری Connection String ابتدا Command Prompt را از مسیر زیر باز کنید

```
Start Menu\Programs\Microsoft Visual Studio 2010\Visual Studio Tools\Visual Studio Command Prompt
```

و سپس دستور زیر را برای رمز نگاری Connection String وارد کنید :

```
aspnet_regiis.exe -pef "connectionStrings" "sulotion path"
```

در اینجا پارامتر اول (pef) مشخص می کند که Application ما از نوع File System Website است، پارامتر دوم (connectionStrings) نام کلید موردنظری است که می خواهیم Encrypt شود، در اینجا می توانیم هر کلیدی را Encrypt کنیم به عنوان مثال فرض کنید تنظیمات SMTP را در Web.Config بدین صورت داریم :

```
<system.net>
  <mailSettings>
    <smtp deliveryMethod="Network">
      <network host="smtp.gmail.com" port="587" userName="username@gmail.com" password="password" />
    </smtp>
  </mailSettings>
</system.net>
```

برای رمز نگاری بدین صورت عمل می کنیم :

```
aspnet_regiis.exe -pef "system.net/mailSettings/smtp" "sulotion path"
```

و بعد از Encrypt کردن توسط دستور فوق بدین صورت در Web.Config نمایش داده می شود :

```
<system.net>
  <mailSettings>
    <smtp configProtectionProvider="RsaProtectedConfigurationProvider">
      <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element"
        xmlns="http://www.w3.org/2001/04/xmlenc#">
        <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc" />
        <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
          <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
            <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5" />
            <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
              <KeyName>Rsa Key</KeyName>
            </KeyInfo>
          <CipherData>
            <CipherValue>QuOFQrT6XwxDhQjFnM3EByleyWqYY61A1cGK1Dzli/hrDOYSj35ADk4MB3PeLOMVYh76kB8vch0/iZKAzaJlNUPKi/iZjEzE755B3sILKGLxfkH3j3qKHBox1WN65L6zBXgzufphCVaNRobQX015J3E0Df8VCF/bERZu741HLPs=</CipherValue>
          </CipherData>
        </EncryptedKey>
      </KeyInfo>
    <CipherData>
      <CipherValue>NdwBe0mWZ+Yg/DEzNuiDfx1GpicoH1ZMn54FTrLuVsY3rawS/k6KPID3bZv0WB/XYseTYFGhqs7FUEqIYMyWjJYYmDAzk6dd4iv9y6ch3ZcXWQ/R5TkQLWoLQPYgdwGI3uJNs22t28xUISm1wS0uDbizCM2Io+DzSQe8N4Ih2MP9mb2NCbZ4BZEBCPvCevpSpdEjGN9v7hk=</CipherValue>
      </CipherData>
    </EncryptedData>
  </smtp>
  </mailSettings>
</system.net>
```

در اینجا از نوع **RSAProtectedConfigurationProvider** انتخاب شده است که می‌توانیم آنرا به **DataProtectionConfigurationProvider** تغییر دهیم که مورد اول از روش رمز نگاری کلید عمومی RSA استفاده می‌کند. برای Decrypt کردن هم فقط کافی است پارامتر اول دستور **aspnet_regiis** را به pdf تغییر دهیم :

```
aspnet_regiis.exe -pdf "system.net/mailSettings/smtp" "solution path"
```

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۰۳ ۱۲:۱۳

البته خیلی‌ها دسترسی به سرور برای انجام این نوع کارها را ندارند. به همین جهت امکان انجام آن از طریق برنامه نویسی هم مهیا شده: ([^](#))

نویسنده: صمد بلاج
تاریخ: ۱۳۹۱/۱۰/۰۸ ۹:۱۵

سلام! من میخوام کانکشن استرینگ رو رمز نگاری کنم.
از روش EF استفاده کردم و تحت ویندوز هست آیا این روش مناسب هستش؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۰۸ ۹:۳۹

بله. [با انواع فایل‌های کانفیگ برنامه‌های دات نت سازگار است.](#)

نویسنده: مرتضی
تاریخ: ۱۳۹۲/۰۳/۲۱ ۱۸:۴۱

سلام؛ بنده سایتم رو رویه لوکال پاپلیش کردم بعد مطالبی رو که بالا نوشته شده رو اعمال کردم و سایت رو ریختم رویه هاست. خطای ران تایم ارور میده. باید ابتدا کانکشن استرینگ رو رمز بکنم بعد عمل پاپلیش رو انجام بدم یا اینکه باید عمل دیگه ای انجام بشه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۲۱ ۲۰:۳۸

حاصل این رمزنگاری بر اساس کلیدهای موجود در کامپیوترهای مختلف متفاوت خواهد بود. بنابراین روی سرور باید رمزنگاری شود. اگر دسترسی مدیریتی دارید، روش خط فرمان مناسب است؛ اگر خیر، این روش با برنامه نویسی هم [قابل اجرا است](#).

نویسنده: مرتضی مختاری
تاریخ: ۱۳۹۲/۰۳/۲۳ ۱۹:۲۶

سلام؛ بنده از طریق کد میخوام کانکشن استرینگ رو رمز بکنم ولی هنگام ذخیره Full ConfigurationSaveMode خطای میده به این عنوان System.Security.Cryptography.CryptographicException: Object already exists چطوری می‌تونم اصلاحش کنم. با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۲۳ ۲۰:۲۵

یک سری بررسی سطح دسترسی و همچنین محافظت نشده بودن ابتدا باید انجام شود:

```
var config = ConfigurationManager.OpenExeConfiguration(ConfigurationUserLevel.None);
var provider = "RsaProtectedConfigurationProvider";
var connStrings = config.ConnectionStrings;
if (connStrings != null)
{
    if (!connStrings.SectionInformation.IsProtected) // اگر محافظت شده نبود
    {
```

رمزگاری Connection String از طریق خط فرمان

```
if (!connStrings.ElementInformation.IsLocked)
{
    // Protect the section.
    connStrings.SectionInformation.ProtectSection(provider);

    connStrings.SectionInformation.ForceSave = true;
    config.Save(ConfigurationSaveMode.Full);
}
```

نویسنده: بهاره
تاریخ: ۱۳۹۲/۰۶/۲۲ ۱۰:۳۲

سلام

اگر پروژه از نوع sharepoint 2003 باشد، آیا این روش موثر است؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۶/۲۲ ۱۱:۱۶

شیرپوینت هم [یک برنامه ASP.NET](#) است.

نویسنده: بهاره
تاریخ: ۱۳۹۲/۱۱/۱۱ ۲۲:۱۸

اگر ویندوز اپلیکیشن بود چی؟ می‌شه هم روش کامندی و هم برنامه نویسیش رو بگید؟ ممنون

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۱/۱۱ ۲۲:۴۷

فرقی نمی‌کنه. روش کدنویسی آن که در نظرات مطرح شده با تمام فایل‌های کانفیگ استاندارد کار می‌کند.

نویسنده: مهرداد
تاریخ: ۱۳۹۳/۰۲/۲۲ ۲:۵۶

لینکی که معرفی کردین فقط رمزگاری رو انجام میده . برای decrypt کردن چه کاری باید انجام بدیم ؟ و اینکه بnde از EF استفاده میکنم . چه موقع و کجا باید این عمل database first encrypt و decrypt را انجام بدم ؟ ضمن اینکه از هاست اشتراکی استفاده میکنم .
متشرکم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۲۲ ۹:۷

لازم نیست [برای آن](#) کاری انجام دهید. خودکار است. نیاز به تنظیم خاصی در برنامه ندارد.

نویسنده: خوزستان
تاریخ: ۱۳۹۴/۰۱/۱۰ ۱۶:۲۱

برای رمزگاری باید کد بالا که فرمودین را بصورت یک متاد نوشت و آن را در Application_Start فراخوانی کرد ؟

نویسنده: محسن خان
تاریخ: ۱۳۹۴/۰۱/۱۱ ۲۰:۱

هر جایی که دوست داشتی فراخوانیش کن. هیچ فرقی نمی‌کنه. فقط مهم اینه که هر تغییری در فایل web.config سبب ری استارت برنامه میشے، بنابراین اینکار یکبار باید انجام بشه و نه بیشتر.

من فایل‌های سایت جاری رو در مسیر استاندارد app_data ذخیره سازی می‌کنم. علت هم این است که این پوشه، جزو پوشه‌های محافظت شده‌ی ASP.NET است و کسی نمی‌تواند فایلی را مستقیماً از آن دریافت و یا سبب اجرای آن با فرآخوانی مسیر مرتبط در مرورگر شود.

این مساله تا به اینجا یک مزیت مهم را به همراه دارد: اگر شخصی مثلًا فایل shell.aspx را در این پوشه ارسال کند، از طریق مرورگر قابل اجرا و دسترسی نخواهد بود و کسی نخواهد توانست به این طریق به سایت و سرور دسترسی پیدا کند. برای ارائه این نوع فایل‌ها به کاربر، معمولاً از روش خواندن محتوای آن‌ها و سپس flush این محتوا در مرورگر کاربر استفاده می‌شود. برای نمونه اگر به لینک‌های سایت دقت کرده باشید مثلًا لینک‌های تصاویر آن به این شکل است:

`http://site/file?name=image.png`

نام فایلی است در یکی از پوشه‌های قرار گرفته شده در مسیر Image.png.app_data هم در اینجا کنترلر فایل است که نام فایل را دریافت کرده و سپس به کمک File FilePathResult و return آن را به کاربر ارائه خواهد داد. تا اینجا همه چیز طبیعی به نظر می‌رسد. اما ... مورد ذیل چطور؟!

Bad file request: ~/a.s. [Details...](#)

Bad file request: ~/.. [Details...](#)

Bad file request: ~/... [Details...](#)

Bad file request: ~/ [Details...](#)

Bad file request: ~/web.config [Details...](#)

You are not allowed to upload script.sql file. Please upload **only** these files: .jpg,.gif,.png. [Details...](#)

Bad file request: ~/bin/myblog.zip [Details...](#)

Bad file request: ../../bin/myblog.zip [Details...](#)

Bad file request: ../../..../bin/myblog.zip [Details...](#)

Bad file request: ../../..../..../bin/myblog.zip [Details...](#)

Bad file request: ../../..../..../..../bin/myblog.zip [Details...](#)

Bad file request: ../../..../..../..../..../bin.zip [Details...](#)

Bad file request: ../../..../..../..../..../bin.zip [Details...](#)

Bad file request: ../../..../..../..../..../bin.zip [Details...](#)

Bad file request: ../../..../..../..../..../..../bin.zip [Details...](#)

Bad file request: ../../..../..../..../..../..../..../bin.zip [Details...](#)

Bad file request: ../../..../..../..../..../..../..../..../bin.zip [Details...](#)

Bad file request: ../../..../..../..../..../..../..../..../..../bin.zip [Details...](#)

Bad file request: ../../..../..../..../..../..../..../..../..../..../bin.zip [Details...](#)

Bad file request: ../../..../..../..../..../..../..../..../..../..../..../bin.zip [Details...](#)

Bad file request: ../../..../..../..../..../..../..../..../..../..../..../..../bin.zip [Details...](#)

Bad file request: ../../..../..../..../..../..../..../..../..../..../..../..../..../bin.zip [Details...](#)

Bad file request: ../../..../..../..../..../..../..../..../..../..../..../..../..../..../bin.zip [Details...](#)

Bad file request: ~/web.config? [Details...](#)

Bad file request: ~/web.config?.jpg [Details...](#)

Bad file request: ~/web.config [Details...](#)

لاغ خطاهای فوق مرتبط است به سعی و خطای شب گذشته یکی از دوستان جهت دریافت فایل web.config برنامه!
متدهای Server.MapPath یا متدهای return File بازخواست می‌دهند. به عبارتی اگر این بررسی امنیتی انجام نشده باشد که کاربر چه مسیری را درخواست می‌کند، محتوای کامل فایل web.config برنامه به سادگی قابل دریافت خواهد بود (به علاوه هر آنچه که در سرور موجود است).

چطور می‌شود با این نوع حملات مقابله کرد؟
دو کار الزامی ذیل حتما باید انجام شوند:

- الف) با استفاده از متدهای Path.GetFileName نام فایل را از کاربر دریافت کنید. به این ترتیب تمام زواید وارد شده حذف گردیده و فقط نام فایل به متدهای مرتبه ارسال می‌شود.
- ب) بررسی کنید مسیری که قرار است به کاربر ارائه شود به کجا ختم شده. آیا به c:\windows اشاره می‌کند یا مثلاً به c:\myapp\app_data
- اگر به لاغ فوق دقت کرده باشید تا چند سطح بالاتر از ریشه سایت هم جستجو شده.

نتیجه گیری:

اگر در برنامه‌های وب خود (فرقی نمی‌کند مرتبط به چه فناوری است)، نام فایل را از کاربر جهت ارائه محتوایی به او دریافت و از این نام فایل بدون هیچ نوع بررسی خاصی، مستقیماً در برنامه استفاده می‌کنید، برنامه شما به مشکل امنیتی [Directory Traversal](#) مبتلا است.

پ.ن.

- 1- این باغ امنیتی در سایت وجود داشت که توسط یکی از دوستان در روزهای اول آن گزارش شد؛ ضمن تشكر!
- 2- از این نوع اسکن‌ها در لاغ‌های خطاهای سایت جاری زیاد است. برای مثال به دنبال فایل‌هایی مانند DynamicStyle.aspx و theme.ashx یا css.ashx می‌گردند. حدس من این است که در یکی از پرتاب‌های معروف یا افزونه‌های این نوع پرتال‌ها فایل‌های یاد شده دارای باغ فوق هستند. فایل‌های ashx عموماً برای flush یک فایل یا محتوا به درون مرورگر کاربر در برنامه‌های ASP.NET مورد استفاده قرار می‌گیرند.

نظرات خوانندگان

نویسنده: فرید صالحی
تاریخ: ۱۳۹۱/۰۵/۰۳ ۱۱:۱۷

تا جایی که من متوجه شدم شما در کنترلر File، اسم فایل رو دریافت می‌کنید و اون رو به مسیری داخل App_Data نگاشت میدید و بعد فایل رو از اون مسیر به کاربر return می‌کنید. اگه به این صورته سوالی واسه من پیش اومند: همونطور که خودتون مثال زدید مهاجم ممکنه به جای اسم فایل یک مسیر مثلا ~/web.config رو به کنترلر بفرسته، خب اگه با این مسیر به صورت یک اسم برخورد بشه مثلا میشه:

```
~/App_Data/~/web.config
```

درسته؟ یعنی در این صورت هم باز از ریشه، فایل web.config برمیگردد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۰۳ ۱۱:۴۰

- این مورد چطور؟

```
var path = Server.MapPath("~/App_Data/..../web.config");
```

حتماً یکبار خروجی آن را دیباگ کنید؛ جالب است.
کاربر هم بجای مسیر یک تصویر یا فایل، مسیر زیر را وارد کرد:

```
..../web.config
```

+ عرض کردم در راه حل های عنوان شده.
اولین بررسی دریافتی از کاربر باید این مورد باشد:

```
var fileName = Path.GetFileName("~/web.config");
```

و نه استفاده مستقیم از نام دریافتی از وب.
خروجی متده فوق (web.config خالی) دیگر به ریشه سایت و یا هیچ مسیری اشاره نخواهد کرد.

نویسنده: فرید صالحی
تاریخ: ۱۳۹۱/۰۵/۰۳ ۱۳:۴۵

اتفاقاً خودم به استفاده از ".." توجه کردم، ولی چون تست نکردم این مورد رو، تصورم این بود که تو سی شارپ جواب نمیده

نویسنده: رحمت الله رضائي
تاریخ: ۱۳۹۱/۰۵/۰۳ ۲۱:۴۹

هدف تست دیشب مشکل دیگری بود که در مطلبی جدا به آن خواهم پرداخت. اما برای اینکه بدانید برای این مورد اخیر چقدر بی توجهی می‌شود کافیست در گوگل صفحاتی را جستجو کنید که آدرس‌هایی اینگونه دارند و سعی کنید با تغییر آدرس، فایل web.config را دانلود کنید:

```
.../download.aspx?file=...
.../download.ashx?file=...
.../get.aspx?file=...
.../get.ashx?file=...
.../download.aspx?path=...
```

```
.../download.ashx?path=...  
...
```

به این صورت جستجو کنید :

```
inurl:"/download.ashx?path="
```

سایتها فارسی زیادی هم میتوانند پیدا کنید که این مشکل را دارند.

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۵/۰۴ ۷:۵۸

من این مشکل رو قبلاً در BlogEngine.NET دیده بودم که برطرفش کردن.

نویسنده: فرهاد یزدان پنا
تاریخ: ۱۳۹۱/۰۵/۰۴ ۱۴:۳۰

آیا ذخیره محتواهای ارسالی (فایل، تصویر و ...) در پایگاه داده بهتر نیست؟
همچنین پشتیبان گیری را راحت می‌کند.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۰۴ ۱۴:۳۴

بله. همینطوره. به شرطی که امکانات سخت افزاری مهیا چنین اجازه‌ای رو به شما بدهد. مثلا سرور اختصاصی داشته باشد با RAM و CPU قابل قبول. به عبارتی در شبکه‌های خصوصی شرکت‌ها، نه سایت‌های عمومی با حداقل‌هایی که در اختیار دارند. یا پردازش ابری ... مثلا برای سازمان‌هایی که می‌توانند هزینه کنند یا دسترسی به این امکانات دارند.

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۰۵/۰۴ ۱۵:۱۵

ممnonم خیلی جالب بود...
یک روش هم میتوشه Encrypt کردن نام فایل باشه که البته سربار خودش رو داره...

نویسنده: فرهاد یزدان پنا
تاریخ: ۱۳۹۱/۰۵/۰۴ ۱۵:۵۳

بله در فضاهایی همچون Azure محدودیت‌هایی بر روی اندازه پایگاه داده و ... وجود دارد. ولی نه در هر نوع محیط ابری.
در ضمن الان شما برای جلوگیری از خیلی مسائل مجبور شدید یک Handler جهت پراکسی اطلاعات (چه ارسالی و چه دریافتی) ایجاد کنید که از خیلی لحاظ مشابه ذخیره فایل‌ها در پایگاه داده باشه.
در ضمن مگه اندازه فایل‌های ارسالی و ... چقدر است که نیازمند مقدار زیادی پردازش باشه. استفاده از مکانیزم‌های Cache موجود در asp.net هم می‌تونه کمک کنه.
حرف شما از خیلی لحاظ صحیحه و شکی در اون نیست ولی میشه به راه حل‌های دیگری هم فکر کرد.

نویسنده: رضوى
تاریخ: ۱۳۹۱/۰۵/۰۵ ۱۵:۸

سلام

باگهایی از این قبیل به باگهای **File Inclusion** معروفند که به دو دسته Local File Inclusion و Remote File Inclusion تقسیم می‌شوند و هکرها با استفاده از آنها می‌توانند به بسیاری از اطلاعات سرور دست پیدا کنند و یا بدون آپلود، شل بگیرند.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۵:۵۹

البته در گزارشات متداول منتشره، هر دو عبارت به جای هم بکار برده می‌شوند و استفاده از file inclusion بیشتر برای سایتها PHP و لینوکسی مرسوم است؛ چیزی مثل این:

`http://site.com/forcedownload.php?file=../../../../../../../../etc/passwd`

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۵/۱۳:۵۳

بله چندتایی از این سایتها رو منم تونستم ببینم، ولی نکته جالب اینه که سایتها غیر ایرانی این مسئله رو حل کردند

نویسنده: میثم جوادی
تاریخ: ۱۳۹۱/۰۵/۲۰:۳۵

یه روش دیگه واسه دانلود [Web.config](#)

در پست‌های قبلی [OpenID](#) چیست؟استفاده از [OpenID](#) در وب سایت جهت احراز هویت کاربران با مفاهیم OpenID تا حدودی آشنا شدیم

در این پست می‌خواهیم با یک مثال ساده نشان دهیم که سایت ما چگونه با استفاده از OpenID Authentication می‌تواند از اکانت گوگل استفاده کرده و کاربر در وب سایت ما شناسایی شود.

برای اینکار ابتدا

1- کتابخانه DotNetOpenAuth را از طریق [NuGet](#) به لیست رفرنس‌های پروژه وب خود اضافه نمایید

2- یک صفحه بنام Login.aspx یا هر نام دلخواهی را به پروژه خود اضافه نمایید. در نهایت کد HTML صفحه شما به ید به صورت زیر باشد.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Login.aspx.cs" Inherits="OAuthWebTest.Login" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div id="loginform">
            <div id="NotLoggedIn" runat="server">
                <asp:ImageButton ID="ButtonLoginWithGoogle" runat="server" ImageUrl="Google.JPG"
                    OnCommand="ButtonLoginWithGoogle_Click"
                    CommandArgument="https://www.google.com/accounts/o8/id" />
                <p>
                    <asp:Label runat="server" ID="LabelMessage" />
                </p>
            </div>
        </div>
    </form>
</body>
</html>
```

در کد HTML بالا به خصوصیت CommandArgument از کنترل ImageButton دقت نمایید که دارای مقدار "https://www.google.com/accounts/o8/id" می‌باشد. در واقع این آدرس باید برای اکانت گوگل جهت احراز هویت توسط OpenID استفاده شود. (آدرس API گوگل برای استفاده از این سرویس)

3- در قسمت کد نویسی صفحه کدهای زیر را وارد نمایید.

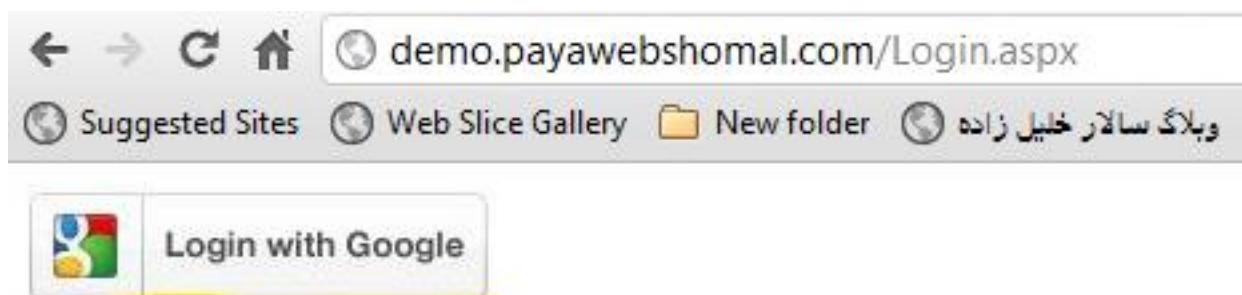
```
using System;
using System.Web.UI;
using System.Web.UI.WebControls;
using DotNetOpenAuth.OpenId.RelyingParty;

namespace OAuthWebTest
{
    public partial class Login : Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            OpenIdRelyingParty openIdRelyingParty = new OpenIdRelyingParty();
            var response = openIdRelyingParty.GetResponse();
            if (response == null) return;
            switch (response.Status)
            {
                case AuthenticationStatus.Authenticated:
                    NotLoggedIn.Visible = false;
                    Session["GoogleIdentifier"] = response.ClaimedIdentifier.ToString();
                    Response.Redirect("Default.aspx");
                    break;
                case AuthenticationStatus.Canceled:
                    LabelMessage.Text = "Cancelled.";
                    break;
            }
        }
    }
}
```

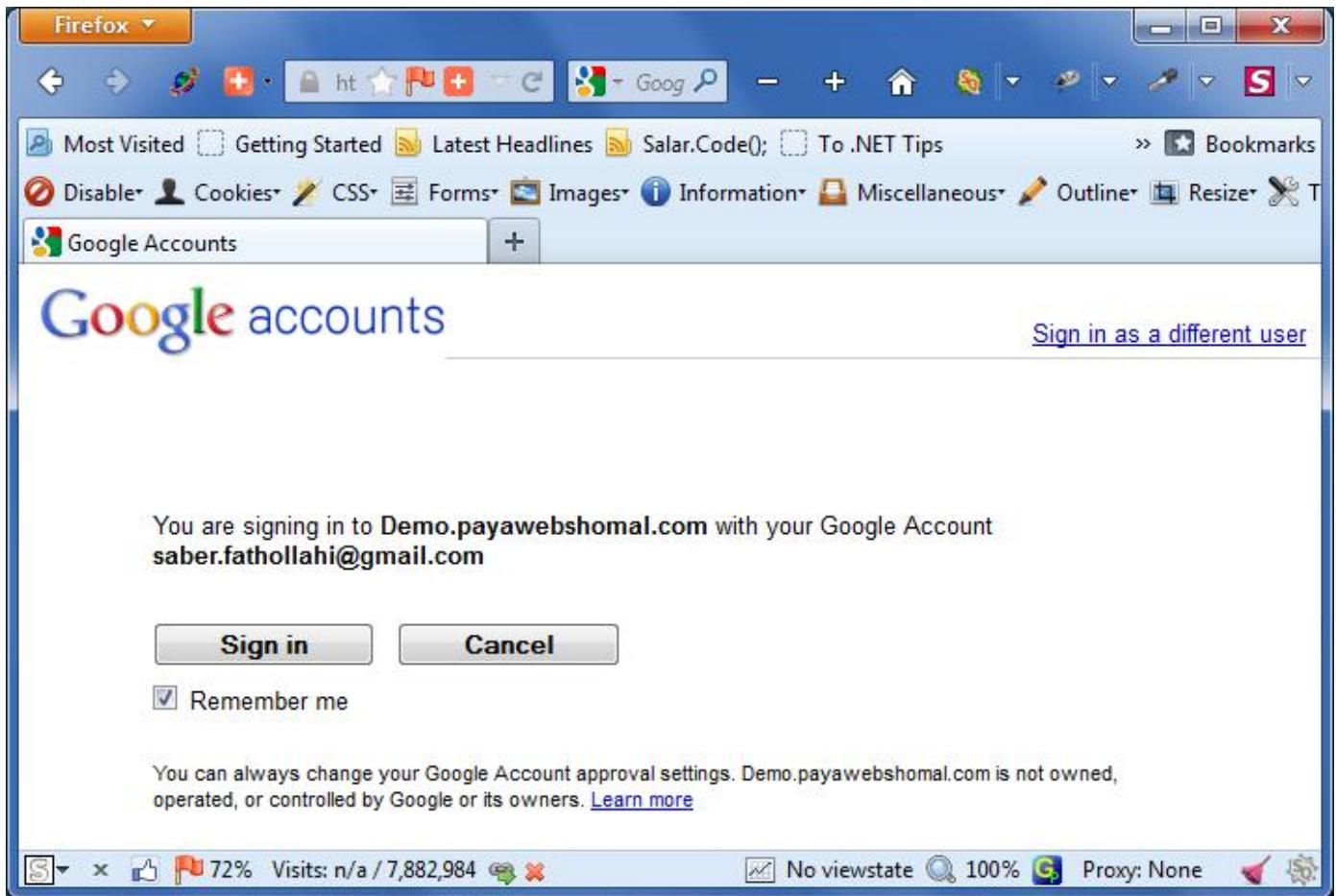
```
        case AuthenticationStatus.Failed:
            LabelMessage.Text = "Login Failed.";
            break;
    }
}

protected void ButtonLoginWithGoogle_Click(object src, CommandEventArgs e)
{
    string discoveryUri = e.CommandArgument.ToString();
    OpenIdRelyingParty openId = new OpenIdRelyingParty();
    var returnToUrl = new UriBuilder(Request.Url) { Query = "" };
    var request = openId.CreateRequest(discoveryUri, returnToUrl.Uri, returnToUrl.Uri);
    request.RedirectToProvider();
}
}
```

همانگونه که مشاهده می‌کنید در رویداد کلیک دکمه لوگین ButtonLoginWithGoogle_Click ابتدا آدرس یکتا گوگل (مقدار CommandArgument اختصاص داده شده به کنترل ImageButton) به همراه آدرس صفحه جاری وب سایت توسط متده CreateRequest از شی openId ترکیب شده و یک درخواست (Request) ساخته شده و در نهایت برای سرویس دهنده گوگل ارسال می‌شود. با اجرای پروژه با تصویر زیر روبرو می‌شوید بروی کلید لوگین کلیک نمایید.



در واقع با کلیک روی دکمه مورد نظر تکه کدی که در بالا شرح داده شد اجرا شده و ما را به صفحه ای شبیه تصویر پایین هدایت می‌کند



در صورتی که کلید Sign In انتخاب شود شما به سایت (همین برنامه) اجازه داده اید که از اطلاعات حساب کاربری شما استفاده کند. پس از کلیک به برنامه اصلی (طبق آدرس بازگشت تعیین شده در رویداد ButtonLoginWithGoogle_Click) در رویداد PageLoad صفحه لاگین اطلاعات بازگشتی از سایت سرویس دهنده (در اینجا گوگل) چک می‌شود و با توجه با پاسخ مورد نظر عملیاتی انجام می‌شود، مثلا در صورتی که شما تایید کرده باشید اطلاعات شناسایی شما توسط گوگل در کلیدی به نام GoogleIdentifier در سشن ذهیره شده و شما به صفحه اصلی سایت هدایت می‌شوید.

دموی پروژه در این [آدرس](#) قرار داده شده است.

سورس پروژه از این [آدرس](#) قابل دسترسی است.

توجه: در این [آدرس](#) شرح داده شده که چگونه دسترسی سایتها دیگر به اکانت گوگل خود را قطع کنید در پستهای آینده اتصال به تویتر و فیسبوک و سایتها دیگر توضیح داده خواهد شد.

نظرات خوانندگان

نوبت‌دهنده: مجتبی چنانی
تاریخ: ۹:۰ ۱۳۹۱/۰۵/۰۴

با سلام و تشکر از پست خوبتون

در زمانی که ما از سیستم‌های ورود و ثبت کاربران شرکت‌های دیگر استفاده می‌کنیم آیا می‌توانیم لاغ گرفته یا اینکه برای خودمان یک صفحه داشته باشیم تا ورود و خروج‌های اکانت‌های درون وبسایتمان را بررسی کنیم یا اینکه خودمان باید این بخش را کدنویسی کنیم؟

یک سئوال دیگر این است که زمانی که از openid های شرکت‌های دیگه استفاده می‌کنیم فقط احراز هویت را از این سرویس‌ها دریافت می‌کنیم یا اینکه در همه صفحات و دیگر کارهای کاربر نظارت به صورت خودکار انجام می‌شود یا اینکه باز هم باید کدنویسی کنیم؟

نوبت‌دهنده: صابر فتح الله
تاریخ: ۱۴:۹ ۱۳۹۱/۰۵/۰۴

در صورتی که بخواهید لاغ بندازید باید خود کد نویسی کرده و در دیتابیس ذخیره کنید بله می‌توان در هر صفحه با استفاده از کلیدی که در سشن کاربر ذخیره کرده اید (با توجه به اینکه سشن کاربران جداست) می‌توانید وجود کاربر را چک کنید

نوبت‌دهنده: سینا علیزاده
تاریخ: ۱۳:۲۹ ۱۳۹۲/۰۹/۰۳

آقا عالی بود ممنون
میشه برای facebook و توییتر هم بزارید ، واقعا بهش نیاز دارم ممنون میشم
کارتون عالیه موفق باشید

نوبت‌دهنده: محسن خان
تاریخ: ۱۳:۳۵ ۱۳۹۲/۰۹/۰۳

نگاهی هم به پروژه [DotNetAuth](#) داشته باشد.

نوبت‌دهنده: دادخواه
تاریخ: ۱۱:۳۱ ۱۳۹۲/۱۰/۱۶

سلام.

این سرویس چطوری برای اکانت‌های یاهو کار می‌کنه؟

سفارشی سازی ایمیل ارسالی :

در مورد ELMAH (Error Logging Module And Handlers) آقای نصیری چندین مطلب نوشته اند (+ و ...) قبل از ارسال ایمیل توسط ELMAH رخدادی به نام Mailing اجرا (Raise) می‌شود. اگر برای این رخداد یک Event Handler ایجاد کنیم، می‌توانیم جزئیات مربوط به خطای که قرار است ارسال شود را تغییر دهیم. به عنوان مثال می‌توانیم بگوئیم اگر Error ما از نوع Application Exception بود آنرا به آدرس دیگری ارسال کن و برای ایجاد یک Event Handler برای رخداد Mailing ابتدا فایل Global.asax را به پروژه اضافه کنید و کد زیر را به آن اضافه کنید :

```
void ErrorMailModuleName_Mailing(object sender, Elmah.ErrorMailEventArgs e)
{
}
```

دقت داشته باشید که در کد فوق به جای `ErrorMailModuleName` نام `HTTPModule` را بنویسید، این مازول در فایل `Web.Config` قرار دارد :

```
<httpModules>
  <add name="ErrorMail" type="Elmah.ErrorMailModule, Elmah" />
</httpModules>
```

که در کد فوق `ErrorMail` می‌باشد در نهایت :

```
void ErrorMail_Mailing(object sender, Elmah.ErrorMailEventArgs e)
{}
```

به عنوان مثال در کد زیر اگر Error از نوع Application Exception بود به آدرس `sirlafifi@gmail.com` نیز ارسال می‌شود :

```
void ErrorMail_Mailing(object sender, Elmah.ErrorMailEventArgs e)
{
    if (e.Error.Exception is ApplicationException)
    {
        e.Mail.To.Add("sirlafifi@gmail.com");
    }
}
```

ذخیره Error‌ها در دیتابیس : SQL SERVER

خطاهای Log شده را در جدولی با نام `ELMAH_Error` ذخیره می‌کند، اگر به داخل پوشه `ELMAH` توجه کرده باشید یک فایل با نام `SQLServer.sql` وجود دارد که حاوی اسکریپت مربوط به ساخت جدول فوق می‌باشد. با اجرای این اسکریپت جدول مربوطه همرا با سه SP با نام‌های `ELMAH_GetErrorsXml`, `ELMAH_GetErrorXml`, `ELMAH_LogError` ساخته می‌شوند. بعد از ساخت جدول مربوطه باید تگ زیر را در فایل `Web.Config` بنویسیم :

```
<errorLog type="Elmah.SqlErrorLog, Elmah"
  connectionStringName="..." />
```

هم نام کانکشن استرینگ را در این قسمت قرار می‌دهیم به عنوان مثال با داشتن کانکشن استرینگ زیر :

```
<connectionStrings>
  <add name="conn" connectionString="Data Source=.;Initial Catalog=test;User
ID=user1;Password=123456;" />
</connectionStrings>
```

به این صورت `connectionStringName` برابر با مقدار `name` یعنی `conn` می‌شود.

نظرات خوانندگان

نویسنده: رضا
تاریخ: ۱۳۹۱/۰۷/۰۳ ۱۱:۳۰

برای استفاده از Elmah در یک پروژه MVC باید پکیج Elmah.MVC را دریافت کنیم یا همون ?Elmah

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۰۳ ۱۱:۳۷

[توضیحات بیشتر در اینجا](#)

نویسنده: امیر
تاریخ: ۹:۳۱ ۱۳۹۲/۰۵/۳۰

با سلام؛ چند سوال:
چطوری بدون استفاده از ممبر شیپ نتوونه هر کسی این آدرس زد این صفحه بیاد. یه جورایی یا پسورد ازش بگیره یا سطح دسترسی برآش تعریف کنیم.

2- در کدوم قسمت فایل وب کانفیگ تغییر بدم که اسم فایل elamh.axd تغییر کند مثلًا بشه test.axd

نویسنده: وحید نصیری
تاریخ: ۹:۳۷ ۱۳۹۲/۰۵/۳۰

به این دو سؤال در مطالب پیشنياز بحث جاري پاسخ داده شده‌اند:

[معرفی ELMAH](#)
[مدیریت خطاهای در یک برنامه ASP.NET MVC](#)

نویسنده: امیر
تاریخ: ۱۰:۰ ۱۳۹۲/۰۵/۳۰

ممnon آقای نصیری. من خوندم ولی نگفته بودین کدوم قسمت رو تغییر بدیم تا نام elamh بشه. فقط گفته شد تغییر داده شود.

نویسنده: وحید نصیری
تاریخ: ۱۰:۳۵ ۱۳۹۲/۰۵/۳۰

در همان [مطلوب معرفی](#) عنوان شده، داریم:

```
<add verb="POST,GET,HEAD" path="elmah.axd" type="Elmah.ErrorLogPageFactory, Elmah" />
```

در اینجا مقدار path را بجای elmah.axd تغییر بدید به هر نام دیگری که علاقمند بودید.

نویسنده: امیر
تاریخ: ۱۲:۱۵ ۱۳۹۲/۰۵/۳۱

چطوری بدون استفاده از ممبر شیپ نتوونه هر کسی این آدرس زد (www.yoursite.com/elmah.axd) این صفحه بیاد. یه جورایی یا پسورد ازش بگیره یا سطح دسترسی برآش تعریف کنیم. چون سطح دسترسی من توسط کد نویسی هست

نویسنده: وحید نصیری
تاریخ: ۱۲:۲۲ ۱۳۹۲/۰۵/۳۱

در همان مطلوب پیشنهاد در مورد location بحث شده، یا حداقل اشاره شده (و اگر کدهای آن ذکر نشده، چون یک مبحث مقدماتی است و نه تکمیلی) . در یک مبحث تکمیلی فرض بر این است که شما حداقل یک فصل از مباحث ابتدایی Forms Authentication را مطالعه کردید. مثلاً می‌دونید تگ location را باید به چه صورتی تعریف کرد، یا چطور باید Role به آن اضافه کرد:

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

اگر پیشتر سایتی را در آدرس مشخصی در اینترنت داشته‌اید و اکنون تنها نرم افزار آن تغییر کرده است، اما نحوه ارائه خدمات آن خیر، لازم است بتوانید شرایط ذیل را مدیریت کنید:

- موتورهای جستجو مدام اطلاعات قبلی خود را به روز می‌کنند. اگر آدرس قبلی مقاله‌ای در سایت شما `http://site/year/month/day/title` شود وجود خارجی دارد یا خیر (این نکته را از لاغ‌های خطای سایت استخراج کردم).
- سایت‌های زیادی هستند که پیشتر به سایت شما و مطالب آن لینک داده‌اند. نمی‌توانید از آن‌ها درخواست کنید لطفاً بانک اطلاعاتی خود را به روز کنید.
- اگر فید قبلی سایت شما `http://site/feeds/posts` بوده و اکنون چیز دیگری است، باز هم نمی‌توانید از همه درخواست کنید اطلاعات خود را به روز کنند. عده‌ای اینکار را خواهند کرد و تعداد زیادی هم خیر.

برای مدیریت یک چنین مواردی می‌توان از امکانات مسیریابی موجود در ASP.NET MVC استفاده کرد؛ که نمونه‌ای عملی از آن را جهت سازگاری سایت جاری با هاست قبلی آن (بلگر) در ادامه مطالعه خواهید نمود:

الف) سازگار سازی لینک‌های قدیمی برچسب‌های سایت با ساختار جدید آن

در بلگر آدرس‌های برچسب‌ها، به صورت `http://site/search/label/name` تعریف شده است. در سایت جاری برچسب‌ها توسط کنترلر Tag مدیریت می‌شوند. برای هدایت آدرس‌های قدیمی (موجود در موتورهای جستجو یا ثبت شده در سایت‌هایی که به ما لینک داده‌اند) می‌توان از تعریف مسیریابی ذیل در فایل `global.asax` استفاده کرد:

```
routes.MapRoute(
    "old_bloger_tags_list", // Route name
    "search/label/{name}", // URL with parameters
    new { controller = "Tag", action = "Index", name = UrlParameter.Optional, area = "" }
)
// Parameter defaults
);
```

به این ترتیب به صورت خودکار تمامی آدرس‌های شروع شده با `http://site/search/label` پالیش شده و سپس قسمت `name` آن‌ها جدا سازی می‌شود. این نام به متدهای `Index` در کنترلر Tag که دارای پارامتری به نام `name` است ارسال خواهد شد.

ب) از دست ندادن خواندنگان قدیمی فیدهای سایت

دو نوع فید کلی در بلگر وجود دارد: `http://site/feeds/comments/default` و `http://site/feeds/posts/default`: اما در سایت جاری فیدها توسط کنترلری به نام Feed ارائه می‌شوند. برای سازگار سازی آدرس‌های قدیمی و هدایت آن‌ها به صورت خودکار به کنترلر فید می‌توان از دو تعریف مسیریابی ذیل استفاده کرد:

```
routes.MapRoute(
    "old_bloger_posts_feeds_list", // Route name
    "feeds/posts/default", // URL with parameters
    new { controller = "Feed", action = "Posts", name = UrlParameter.Optional, area = "" }
)
// Parameter defaults
);
routes.MapRoute(
    "old_bloger_comments_feeds_list", // Route name
    "feeds/comments/default", // URL with parameters
    new { controller = "Feed", action = "Comments", name = UrlParameter.Optional, area = "" }
)
// Parameter defaults
);
```

در اینجا دو آدرس ذکر شده به کنترلر Feed و متدهای Posts و Comments آن هدایت خواهند شد و به این نحو کاربران قدیمی سایت هیچگونه تغییری را احساس نکرده و باز هم فیدخوان‌های آن‌ها، بدون مشکل کار خواهند کرد.

ج) پردازش لینک‌های قدیمی مطالب سایت و هدایت آنها به آدرس‌های جدید
این مورد اندکی مشکل‌تر از موارد قبلی است:

```
routes.MapRoute(
    "old_bloger_post_urls",
    "{yyyy}/{mm}/{title}",
    new { controller = "Post", action = "OldBloggerLinks" },
    new { yyyy = @"\d{4}", mm = @"\d{1,2}" }
);
```

برای نمونه آدرس مقاله‌ای مانند <http://site/2012/05/ef-code-first-15.html> را در نظر بگیرید. سه قسمت سال، ماه و عنوان آن، حائز اهمیت هستند. این‌ها را در اینجا به کنترلر Post و متد OldBloggerLinks آن هدایت خواهیم کرد. همچنین برای سال و ماه آن نیز قید تعریف شده است. سال عددی 4 رقمی است و ماه عددی یک تا دو رقمی.
کدهای متد OldBloggerLinks را در اینجا مشاهده می‌کنید:

```
public virtual ActionResult OldBloggerLinks(int yyyy, int mm, string title)
{
    var oldUrl = string.Format(CultureInfo.InvariantCulture,
    "http://www.dotnettips.info/{0}/{1}/{2}", yyyy, mm.ToString("00"), title);
    var blogPost = _blogPostsService.FindBlogPost(oldUrl);
    if (blogPost != null)
        return RedirectToActionPermanent(actionName: ActionNames.Index, controllerName:
MVC.Post.Name,
                                         routeValues: new { id = blogPost.Id, name =
blogPost.Title.GetPostSlug() });
    return this.Redirect("/");
}
```

در اینجا چون ساختار لینک‌ها کلاً تغییر کرده است، ابتدا بر اساس پارامترهای دریافت شده، لینک قدیمی بازسازی می‌شود. سپس به بانک اطلاعاتی مراجعه شده و لینک قدیمی به همراه شماره مطلب مرتبط با آن یافت می‌شود (یک فیلد oldUrl برای مطالب قدیمی در بانک اطلاعاتی وجود دارد). در آخر هم به کمک متد RedirectToActionPermanent آدرس رسیده به آدرس جدید مطلب در سایت ترجمه و هدایت خواهد شد. Permanent بودن آن برای به روز رسانی خودکار اطلاعات موتورهای جستجو مفید است.

نتیجه گیری

به کمک امکانات مسیریابی توکار ASP.NET MVC می‌توان ساختار قدیمی یک سایت را به ساختار جدید آن ترجمه کرد. به این ترتیب لینک‌های قدیمی ثبت شده در صدھا سایت اینترنتی که به سایت ما اشاره می‌کنند، مجدداً بدون مشکل قابل استفاده بوده و همچنین موتورهای جستجو نیز امکان به روز رسانی اطلاعات خود را خواهند یافت.

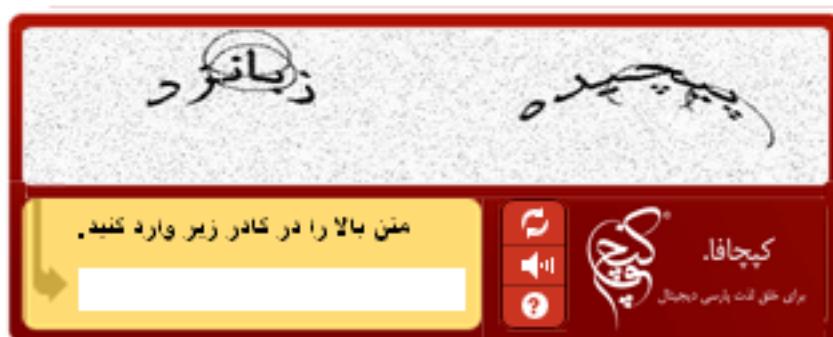
حتماً با CAPTCHA آشنا هستید. فرایندی که در طی آن متنی نمایش داده می‌شود که عمدتاً فقط یک انسان قادر به درک و پاسخگویی به آن است. با این کار از ارسال داده‌های بیهوده توسط ربات‌ها جلوگیری می‌شود. reCAPTCHA ایده‌ای است که با نمایش کلمات واقعی و اسکن شده از کتاب‌های قدیمی، بخشی از مشکلات را حل کرده و از 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 متنون 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 انسستیتو تکنولوژی ایلینویز شیکاگو، پروژه‌ای بر همین اساس اما برای متنون فارسی با عنوان CAPTCHAfa تولید کرده اند (شکل زیر) که در [این آدرس](#) در دسترس است. امیدوارم این پروژه به گونه‌ای تغییر کنے 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 روای Page_Load، کد زیر را قرار دهید. این کار برای تزریق اسکریپت CAPTCHAFA به صفحه استفاده ممکن است.

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

بررسی، صحت مقدار وارد شده توسط کاربر (مثلاً در `RowA`، `Click` یک دکمه) به صورت زیر است.

```
Captchafa captchaFa = new Captchafa();
bool isAnswerCorrect = captchaFa.IsAnswerCorrect();

if (isAnswerCorrect)
{
    پاسخ صحیح است // پاسخ صحیح نیست //
}
else
{
    پاسخ صحیح است // 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.Captcha();
<input type="submit" id="btnCaptcha" name="btnCaptcha" 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 recaptha 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 "مسلسل 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 مشتقاًم 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 سایتها فارسی مرسوم بشه

نویسنده: علیرضا اسمراهم
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 جزئیات را دقیق ببینم، اما امیدوارم تلفظ و پخش کلمات بصورت صوتی خوب عمل کند!

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

بیشترین خطاهای 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 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 شیوه‌ای که کلمه تایپ شده کاربر با عکس مقایسه میشه، به همان طریق میشه متون کاغذی را دیجیتالی کرد و نیازی به استفاده از این شیوه برای دیجیتالی 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 CAPTCHA مانند reCAPTCHA که بر اساس دیجیتالی کردن متون پیاده سازی شدند، دو کلمه رو به کاربر نشان میدن:

- (1) کلمه‌ای که توسط رایانه به درستی تشخیص داده شده
- (2) کلمه‌یا بخشی از اون که رایانه تونسته اون رو به درستی تشخیص بده.

در صورتی که کاربر، کلمه‌ای 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، متدهای GetHtml به CaptchafaGetHtml تغییر پیدا کرد، مدیریت خطاهای 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

فکر کنم که کلا پروژه تعطیل شده ...

ایجاد یک Pattern در پروژتون میتوانه نظم، سرعت و زیبایی خاصی به کدتون بده. با وجود framework‌هایی مسنه MVC و MVVM برنامه نویسان را وادار کنه که همه Action‌های یک پروژه رو به سمت کلینت ببرن. تو یک فرصت دیگه در مورد فریمورک Knockout حتما تاپیک میزارم. امروز میخواهم یک Pattern با استفاده از یک Interface و codefirst model برآتون بزارم.

گام اول: ایجاد که class property

```
Public Class Employee
    Public Property ID As Integer
    Public Property Fname As String
    Public Property Bdate As DateTime
End Class
```

گام دوم: ایجاد بانک با استفاده از CodeFirst

```
Imports System.Data.Entity
Public Class EmployeeDBContext : Inherits DbContext
    Public Property Employees As DbSet(Of Employee)
End Class
```

گام سوم: ایجاد repository با استفاده از interface

```
Interface EmployeeRepository
    ReadOnly Property All As List(Of Employee)
    Function Find(id As Integer) As Employee
    Sub InsertOrUpdate(p As Employee)
    Sub Delete(id As Integer)
    Sub Save()
End Interface
```

گام چهارم: تعریف کلاس برای implement کردن از iInterface

```
Public Class EmployeeClass : Implements EmployeeRepository
    Private DB As New EmployeeDBContext
    Public ReadOnly Property All As List(Of Employee) Implements EmployeeRepository.All
        Get
            Return DB.Employees.ToList()
        End Get
    End Property
    Public Sub Delete(id As Integer) Implements EmployeeRepository.Delete
        Dim query = DB.Employees.Single(Function(q) q.ID = id)
        DB.Employees.Remove(query)
    End Sub
    Public Function Find(id As Integer) As Employee Implements EmployeeRepository.Find
        Return DB.Employees.Where(Function(q) q.ID = id)
    End Function
    Public Sub InsertOrUpdate(p As Employee) Implements EmployeeRepository.InsertOrUpdate
        If p.ID = Nothing Then
            DB.Employees.Add(p)
        Else
            DB.Entry(p).State = Data.EntityState.Modified
        End If
    End Sub
    Public Sub Save() Implements EmployeeRepository.Save
        DB.SaveChanges()
    End Sub
End Class
```

ایجاد یک Repository در پروژه برای دستورات EF

```
End Class
```

برای استفاده تو پروژه براحتی میتوانید یک instance از class آن را ایجاد کنید و ..

```
Dim cls As New EmployeeClass  
  
Public Sub BindGrid()  
    GridView1.DataSource = cls.All  
    GridView1.DataBind()  
End Sub
```

موفق باشید

نظرات خوانندگان

نویسنده: علیرضا صالحی
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۲:۱۳

برای مواردی که خروجی یک لیست (تعدادی آیتم) باشد از `Property` استفاده نمی‌شود. مثلاً برای `A11` باید از `Method` استفاده کنید.

[Properties vs. Methods](#)

بهتر است برای خروجی متدهای `A11` نیز به جای لیست از `IQueryable` یا `IEnumerable` استفاده کنید. متدهای `Insert` و `Update` نیز به طور جداگانه تعریف شوند. (قرار است هر متدهای تنها یک وظیفه داشته باشد)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۲:۲۱

- در مورد آرایه بحث شده در MSDN. ضمن اینکه استفاده از متدهای عموماً برای حالتیکه عملیات قابل توجهی در بدنه آن قرار است صورت گیرد، [توصیه می‌شود](#). البته در اینجا چون عملیات دریافت اطلاعات از بانک اطلاعاتی می‌تواند سنگین درنظر گرفته شود، استفاده از متدهای ارجحیت دارد. خواص نمایانگر اطلاعاتی سبک و با دسترسی سریع هستند.

- خروجی لیست بهتر است. ([^](#)) + اگر ReSharper جدید را نصب کنید استفاده از `IEnumerable` را [نیز توصیه نمی‌کند](#)؛ چون ممکن است چندین بار رفت و برگشت به بانک اطلاعاتی در این بین صورت گیرد.

- مشکلی ندارد. خود EF Code first چنین متدهای دارد. ([^](#)) بحث کلاس تک وظیفه‌ای متفاوت است با متدهای که نهایتاً قرار است اطلاعات یک رکورد را در بانک اطلاعاتی تغییر دهد (اگر نبود ثبتش کند؛ اگر بود فقط همان رکورد مشخص را به روز رسانی کند).

نویسنده: ناشناس
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۲:۳۷

با سلام

دلیل استفاده از `Interface EmployeeRepository` چیه؟

دقیقاً دلیل استفاده `Interface` اینجا چیه؟

با تشکر از مطلب خوبتون.

نویسنده: ناشناس
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۲:۴۵

اینجا شاید استفاده از `IQueryable` بهتر باشد.

شاید کاربر بخواهد قبل از نمایش اطلاعات اونو فیلتر کنه یا اینکه بهتره دو متدهای `Find` داشته باشی یک آیتم و دیگری با خروجی چندین آیتم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۲:۴۸

خیر ([^](#)). طراحی یک لایه سرویس که خروجی `IQueryable` دارد نشستی دار درنظر گرفته شده و توصیه نمی‌شود. اصطلاحاً [leaky abstraction](#) هم به آن گفته می‌شود؛ چون طراح نتوانسته حد و مرز سیستم خودش را مشخص کند و همچنین نتوانسته سازوکار درونی آن را به خوبی کپسوله سازی و مخفی نماید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۲:۵۰

به دو دلیل:

- استفاده از امکان تزریق وابستگی‌ها

- امکان نوشتتن ساده‌تر آزمون‌های واحد با فراهم شدن زیر ساخت mocking اشیاء

نویسنده: ناشناس
تاریخ: ۱۳:۲ ۱۳۹۱/۰۵/۱۹

در [همین پست](#) خود شما تعداد زیاد رکوردها رو مثال زدیدو این پیاده سازی از این موضوع رنج میبره. و در مورد `IQueryable` قبول دارم و گفتم که بهتر است از دو یا چند متده `find` استفاده کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳:۲۵ ۱۳۹۱/۰۵/۱۹

منظور از آن مطلب این بود که از ابزاری که در اختیار دارید درست استفاده کنید. اگر قرار است دو یا چند جستجو را انجام دهید، اینکارها بله باید با `IQueryable` داخل یک متده انجام شود، اما خروجی متده فقط باید لیست حاصل باشد؛ نه `IQueryable` ای که انتهای آن باز است و سبب نشتشی لایه سرویس شما در لایه‌های دیگر خواهد شد.

نویسنده: میثم ثوامری
تاریخ: ۱۹:۵۴ ۱۳۹۱/۰۵/۱۹

برای برنامه نویسا پیدا کردن یک `property` راحتره در ضمن از `property` برای تزریق یا بازیابی اطلاعات از یک `object` استفاده میکنن.

`IQueryable` در واقع توسعه یافته `IEnumerable`.تفاوت عمدشون در LINQ operators که در `IQueryable` استفاده میشه.اگر هم بخوایم دلیل پیشنهادی داده باشیم اونم اینه که مدیریت حافظه در `IQueryable` رعایت شده در حالی که `List`ها کامپایلرو مجاب به اجنبام دستور تا انتهای میکنن.

نویسنده: میثم ثوامری
تاریخ: ۱۹:۵۹ ۱۳۹۱/۰۵/۱۹

مهندس با نظر دوستمون موافقم
`IQueryable` بهترین انتخاب برای `remote data source` یا `webservice` database اشاره کرد.بطور کل اگر شما از `linqtosql` مسنه ORM استفاده میکنید `IQueryable` کوئری شمارو به دستورات `sql` در `database server` تبدیل میکنه `IEquatable`: همه رکوردهای شما قبل از اینکه بسمت دیتابیس برن بصورت `object` در `memory` نگهداری میشن.

نویسنده: وحید نصیری
تاریخ: ۲۰:۱۱ ۱۳۹۱/۰۵/۱۹

`IQueryable` فقط یک `expression` است. هنوز اجرا نشده. (آن از طریق وب سرویس اشتباه است و به مشکلات `serialization` برخواهد خورد).
زمانیکه `ToList`, `First` و امثال آن روی این عبارت فراخوانی شود تبدیل به SQL شده و سپس بر روی بانک اطلاعاتی اجرا میشود. به این `deferred execution` یا اجرای به تعویق افتاده گفته میشود.
اگر این عبارت را در اختیار لایه‌های دیگر قرار دهید، یعنی انتهایی کار را بازگذاشته‌اید و حد و حدود سیستم شما مشخص نیست. شما اگر `IQueryable` بازگشت دهید، در لایه‌ای دیگر می‌شود یک `join` روی آن نوشته و اطلاعات چندین جدول دیگر را استخراج کرد؛ در حالیکه نام متده `GetUsers` بوده. بنابراین بهتر است به صورت صريح اطلاعات را به شکل `List` بازگشت دهید، تا انتهای کار باز نمانده و طراحی شما نشتشی نداشته باشد.

نویسنده: محمد عامریان
تاریخ: ۱۷:۶ ۱۳۹۱/۰۸/۱۸

با سلام من یک معماری طراحی کردم به شکل زیر
ابتدا یک اینترفیس به شکل زیر دارم

```
using System;
using System.Collections;
using System.Linq;

namespace Framework.Model
{
    public interface IContext
    {
        T Get<T>(Func<T, bool> prediction) where T : class;
        IEnumerable<T>(Func<T, bool> prediction) where T : class;
        void Insert<T>(T entity) where T : class;
        int Save();
    }
}
```

بعد یک کلاس ازش مشتق شده

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Text;

namespace Framework.Model
{
    public class Context : IContext
    {
        private readonly DbContext _dbContext;

        public Context(DbContext context)
        {
            _dbContext = context;
        }

        public T Get<T>(Func<T, bool> prediction) where T : class
        {
            var dbSet = _dbContext.Set<T>();
            if (dbSet != null)
                return dbSet.Single(prediction);

            throw new Exception();
        }

        public void Insert<T>(T entity) where T : class
        {
            var dbSet = _dbContext.Set<T>();
            if (dbSet != null)
            {
                _dbContext.Entry(entity).State = EntityState.Added;
            }
        }

        public int Save()
        {
            return _dbContext.SaveChanges();
        }

        IEnumerable<T>(Func<T, bool> prediction)
        {
            var dbSet = _dbContext.Set<T>();
            if (dbSet != null)
                return dbSet.Where(prediction).ToList();

            throw new Exception();
        }
    }
}
```

ایجاد یک Repository در پروژه برای دستورات EF

```
using System.Data.Entity;
using DataModel;

namespace Model
{
    public class EFContext : DbContext
    {
        public EFContext(string db): base(db)
        {

        }

        public DbSet<Product> Products { get; set; }
    }
}
```

و سپس کلاس دارم که او مده پیاده سازی کرده context که خودم ساختم

```
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
using System.Text;

namespace Model
{
    public class Context : Framework.Model.Context
    {
        public Context(string db): base(new EFContext(db))
        {

        }
    }
}
```

در پروژه دیگری او مدم یک کلاس context جدید ساختم

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Biz
{
    public class Context : Model.Context
    {
        public Context(string db) : base(db)
        {

        }
    }
}
```

و در کنترلر هم به این شکل ازش استفاده کردم

```
using System.Web.Mvc;
using Framework.Model;

namespace ProductionRepository.Controllers
{
    public class BaseController : Controller
    {
        public IContext DataContext { get; set; }

        public BaseController()
        {
            DataContext = new
Biz.Context(System.Configuration.ConfigurationManager.ConnectionStrings["Database"].ConnectionString);
        }
    }
}
```

```
using System.Web.Mvc;
using DataModel;
using System.Collections.Generic;

namespace ProductionRepository.Controllers
{
    public class ProductController : BaseController
    {
        public ActionResult Index()
        {
            var x = DataContext.List<Product>(s => s.Name != null);
            return View(x);
        }
    }
}
```

و این هم تست

```
using NUnit.Framework;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Web.Mvc;

namespace TestUnit
{
    [TestFixture]
    public class Test
    {
        [Test]
        public void IndexShouldListProduct()
        {
            var repo = new Moq.Mock<Framework.Model.IContext>();
            var products = new List<DataModel.Product>();
            products.Add(new DataModel.Product { Id = 1, Name = "asdasdasd" });
            products.Add(new DataModel.Product { Id = 2, Name = "adaqwe" });
            products.Add(new DataModel.Product { Id = 4, Name = "qewqw" });
            products.Add(new DataModel.Product { Id = 5, Name = "qwe" });
            repo.Setup(x => x.List<DataModel.Product>(p => p.Name != null)).Returns(products.AsEnumerable());
            var controller = new ProductionRepository.Controllers.ProductController();
            controller.DataContext = repo.Object;
            var result = controller.Index() as ViewResult;
            var model = result.Model as List<DataModel.Product>;
            Assert.AreEqual(4, model.Count);
            Assert.AreEqual("", result.ViewName);
        }
    }
}
```

نظرتون چیه آقای نصیری

نویسنده: وحید نصیری
تاریخ: ۱۷:۱۳ ۱۳۹۱/۰۸/۱۸

موارد ۱ و ۲ عنوان شده در این مطلب رو تکرار کرد: ([^](#))

نویسنده: مجید پارسا
تاریخ: ۱۲:۹ ۱۳۹۳/۰۷/۱۲

با سلام؛ سوالی که وجود داره اینه که با استفاده از repository pattern چطور میتوانیم join بزنیم. با توجه به نظرات قبلی توصیه شده است که از خروجی IQueryable نباید برای لایه داده استفاده شود.

در این صورت در هنگام نوشتن دستورات join ابتدا تمامی رکوردهای جداول مورد نظر توسط الگوی repository به حافظه load میشود، با توجه به ماهیت linq to object بودن کوئری مورد نظر(join) اجرای برنامه به لحاظ زمانی و مصرف حافظه از

کارایی خوبی برخوردار نخواهد بود.

در این حالت یا می‌بایست از خیر کارایی بالاتر گذشت یا از خروجی `IQueryble` استفاده کرد که در تضاد با پیشنهاد دوستان گرامی می‌باشد.

آیا در این حالت منطقی است `join`‌های پر استفاده را با خروجی `repository` در `IEnumerable` مربوط به خودش نوشت یا راهکار دیگری وجود دارد؟

نویسنده: وحید نصیری
تاریخ: ۱۲:۱۹ ۱۳۹۳/۰۷/۱۲

- الگوی مخزن عمومی (`Generic repository pattern`), لایه داده برنامه نیست. زمانیکه از یک ORM استفاده می‌کنید، لایه داده برنامه همان ORM است.

- الگوی مخزن عمومی، عمدۀ کارش مخفی کردن ساز و کار ORM مورد استفاده از لایه سرویس برنامه است (▲).

- اگر از الگوی عمومی مخزن استفاده می‌کنید، سطح دسترسی آن را `internal` تعریف کنید تا محدود شود به لایه سرویس برنامه. داخل لایه سرویس برنامه به هر نحوی که علاقمندید از آن استفاده کنید. نهایتاً این لایه سرویس است که خروجی `IList` یا `IEnumerable` نهایی را در اختیار مصرف کننده قرار می‌دهد.

نویسنده: مجید پارسا
تاریخ: ۱۶:۸ ۱۳۹۳/۰۷/۱۲

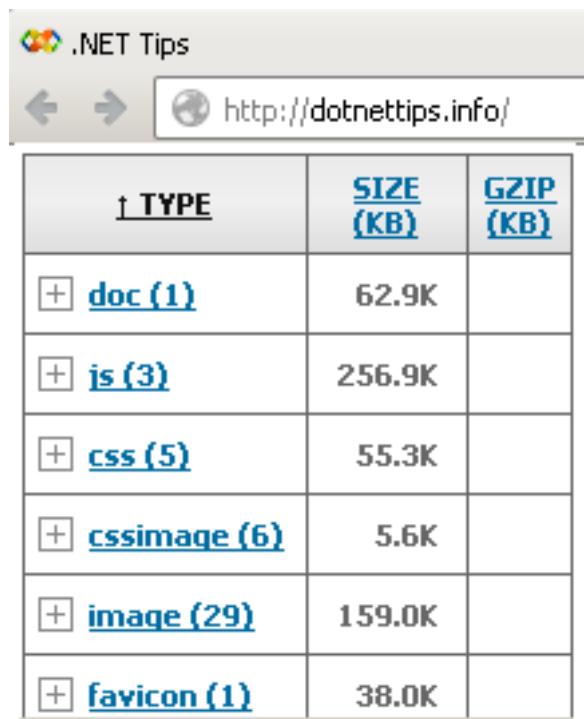
با تشکر، از آنجا که من اولین بار است که به شکل حرفه‌ای برنامه نویسی سه لایه را تجربه می‌کنم با توجه به توضیحات شما این طور متوجه شدم که پیاده سازی کلاس‌های `Repository` در لایه سرویس صورت گیرد اگر اشتباه نکنم.

در صورت امکان بیشتر موضوع رو باز کنید (منظورم آماتوری تره)
نمونه برنامه‌های سه لایه موجود در اینترنت پیدا کردم در حد `CRUD` ساده و با استفاده از الگوی مخزن عمومی بوده. مانند مثال‌های سایت `asp.net` در صورت معرفی نمونه کاملتر و واقعی‌تر ممنون می‌شوم.

نویسنده: وحید نصیری
تاریخ: ۱۷:۲۹ ۱۳۹۳/۰۷/۱۲

مراجعه کنید به [مسیر راه EF Code first](#) ، انتهای مطلب، قسمت لایه بندي پروژه‌های

زمانیکه صفحه‌ای لود می‌شود، به همراه خود محتویاتی از قبیل تصویر و فایلهای جاوا اسکریپت را نیز به سیستم کاربر حمل می‌کند. در ASP.NET می‌توان از خاصیت فشرده سازی [gzip](#), [gzzip](#) جهت ارائه محتویات سرور استفاده کرد:



The screenshot shows a table titled ".NET Tips" with a header row containing columns for TYPE, SIZE (KB), and GZIP (KB). The table lists the following data:

TYPE	SIZE (KB)	GZIP (KB)
[+] doc (1)	62.9K	
[+] is (3)	256.9K	
[+] css (5)	55.3K	
[+] cssimage (6)	5.6K	
[+] image (29)	159.0K	
[+] favicon (1)	38.0K	

برای پیاده سازی آن می‌توان از رویداد Application_PreRequestHandlerExecute در فایل Global.asax استفاده کرد:

```
using System;
using System.IO.Compression;
using System.Web;
using System.Web.UI;

namespace GZipTest
{
    public class Global : HttpApplication
    {
        void Application_PreRequestHandlerExecute(object sender, EventArgs e)
        {
            var app = sender as HttpApplication;
            if (app == null) return;

            var acceptEncoding = app.Request.Headers["Accept-Encoding"];
            var prevUncompressedStream = app.Response.Filter;

            if (!(app.Context.CurrentHandler is Page ||
                  app.Context.CurrentHandler.GetType().Name == "SyncSessionlessHandler") ||
                app.Request["HTTP_X_MICROSOFTAJAX"] != null)
                return;

            if (string.IsNullOrEmpty(acceptEncoding))
                return;

            acceptEncoding = acceptEncoding.ToLower();

            if (acceptEncoding.Contains("deflate") || acceptEncoding == "*")
```

```
{  
    // defalte  
    app.Response.Filter = new DeflateStream(prevUncompressedStream,  
CompressionMode.Compress);  
    app.Response.AppendHeader("Content-Encoding", "deflate");  
}  
else if (acceptEncoding.Contains("gzip"))  
{  
    // gzip  
    app.Response.Filter = new GZipStream(prevUncompressedStream, CompressionMode.Compress);  
    app.Response.AppendHeader("Content-Encoding", "gzip");  
}  
}  
}  
}
```

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۱۵ ۱۳:۲۲

در مورد تصویر ارسالی:

در این سایت از این روش (`System.Web.Optimization`) استفاده می‌شود. روش یاد شده در IIS7 خروجی فشرده شده با GZip نیز دارد؛ اما نه در IIS6. ولی در هر دو حالت، کش را تنظیم می‌کنند:

TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/M/D)
+ doc (1)	28.4K						
- js (3)	256.9K						
js	36.8K	14.8K			🔍	http://www.google-analytics.com/qa.js	2012/9/5
js	179.2K				🔍	http://www.dotnettips.info/Scripts/js?...	2013/9/4
js	40.8K				🔍	http://www.dotnettips.info/Content/redactor/js?...	2013/9/4

به همین جهت در بار بعدی مشاهده سایت، دیگر درخواست اضافه‌ای جهت دریافت اسکریپت‌ها و شیوه نامه‌ها ارسال نخواهد شد:

URL	Status	Domain	Time
GET www.dotnettips.info	200 OK	dotnettips.info	60.2ms
GET utm.gif?utmwv=5....tml%	200 OK	google-analytics.com	60.2ms
2 requests			

نویسنده: مهدی پایرونده
تاریخ: ۱۳۹۱/۰۶/۱۵ ۱۳:۲۸

ممون، راهکارهایی که شما قبل از کاهش درخواست بین کلاینت و سرور داده بودید حداقل برای بندۀ خیلی کارگشا بوده و این مطلب برای استفاده بهتر از امکان فشرده سازی آورده شده.
پ ن: عکس برای درک مسئله آورده شده بود!

نویسنده: علیرضا صالحی
تاریخ: ۱۳۹۱/۰۶/۱۵ ۱۵:۲۴

روی ویندوز سرورهای 2008 (IIS 7 و بالاتر) این فشرده سازی توسط IIS فقط با تنظیم ساده‌ای در web.config انجام می‌شود. در اغلب هاستینگ‌هایی هم که من استفاده کردم این قابلیت فعال بوده است و نیازی به کدنویسی نیست. و البته بستگی به نوع پروژه هم ندارد.

```
<httpCompression directory="%SystemDrive%\inetpub\temp\IIS Temporary Compressed Files">
<scheme name="gzip" dll="%Windir%\system32\inetsrv\gzip.dll" doStaticCompression="true"/>
<staticTypes>
<add mimeType="text/*" enabled="true" />
<add mimeType="message/*" enabled="true" />
<add mimeType="application/x-javascript" enabled="true" />
<add mimeType="/*/*" enabled="false" />
</staticTypes>
</httpCompression>
<urlCompression doStaticCompression="true" doDynamicCompression="false" />
```

اطلاعات بیشتر :

[HTTP Compression](#)

[Scheme](#)

[URL Compression](#)

نوبنده: احسان
تاریخ: ۲۳:۴۲ ۱۳۹۱/۰۶/۱۵

آقای نصیری ممنونم از این مطلبتون اما ای سوال :
من روی سرورم IIS 6 دارم و بدون هیچ گونه تنظیم خاصی می‌بینم که در سربرگ Response، همه Response‌ها بطور خودکار gzip شدن. بدون اینکه کدی نوشته باشم یا کاری کرده باشم.



Components

The page has a total of 10 components.

↑ TYPE	SIZE (KB)	GZIP (KB)
[+] doc (1)	10.9K	
doc	10.9K	3.6K
[+] js (2)	122.6K	
js	94.8K	32.6K
js	27.8K	4.6K
[+] css (2)	3.4K	
css	1.9K	0.7K
css	1.5K	0.6K
[+] cssimage (5)	19.2K	

آیا خود IIS6 بطور خودکار این فشرده سازی را انجام می‌دهد؟

نویسنده: وحید نصیری
تاریخ: ۲۳:۴۵ ۱۳۹۱/۰۶/۱۵

بله. سرور دار [یک سری تنظیمات رو](#) می‌توانه اعمال کنه تا اینکار برای تمام سایت‌های هاست شده خودکار شود.

نویسنده: احسان
تاریخ: ۰:۰ ۱۳۹۱/۰۶/۱۶

ممنون از توجهتون.

نویسنده: میلاد
تاریخ: ۱۴:۲۳ ۱۳۹۱/۰۶/۳۱

سلام و ممنون از مطالب مفیدتون،
من به تازگی وب سایتی ساختم و با خوندن این مطلب ، فشرده سازی رو روی سایت انجام دادم.
اما مشکل اینجاست که این فشرده سازی تنها روی سرور IIS لوکال جواب میده و وقتی که سایت رو روی هاست آپلود می‌کنم ،

هیچ فشرده سازی انجام نمیشے.
لطفا راهنماییم کنید.
با تشکر

نویسنده: مهدی پایروند
تاریخ: ۱۶:۲۴ ۱۳۹۱/۰۶/۳۱

همونطور که مهندس نصیری در کامنت اول این پست گفتند این تنظیمات بر روی `iis` نسخه 7 به بالا به این صورت هم جواب میده و لی برای نسخه‌های پائین باید از `iis` خود سرور کمک گرفت ولی در کل منظور از ایجاد این پست معرفی این نوع اعمال، برای سایتها است

نویسنده: وحید نصیری
تاریخ: ۱۶:۵۴ ۱۳۹۱/۰۶/۳۱

. عموما روی IIS6 مشکل ساز است و بهتر است از `AddHeader` استفاده شود.

نویسنده: میلاد
تاریخ: ۲۳:۵۴ ۱۳۹۱/۰۶/۳۱

سلام
ممnon از پاسختون...
با `AddHeader` هم امتحان کردم... اما موفق نشدم... لطفا یک نگاهی به تصویر ضمیمه بیاندازید... شاید چیزی دستگیرتون بشه:

Request Headers

GET /script/smoothDivScroll.css HTTP/1.1

Client

Accept: text/css,*/*;q=0.1
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:15.0) Gecko/20100101 Firefox/15.0.1

Cookies / Login

Cookie

ASP.NET_SessionId=vcr5comle4tzouramit1ksds

Miscellaneous

Referer: http://www.zamaneh.com/

Transport

Connection: keep-alive
Host: www.zamaneh.com

[Set SyntaxView](#) | [Transformer](#) | [Headers](#) | [TextView](#) | [ImageView](#) | [HexView](#)

[Raw](#) | [JSON](#) | [XML](#)

Response Headers

HTTP/1.1 200 OK

Cache

Date: Fri, 21 Sep 2012 20:15:49 GMT

Entity

Content-Length: 2296
Content-Type: text/css
ETag: "d9e9b9c2f796cd1:0"
Last-Modified: Thu, 20 Sep 2012 06:18:33 GMT

Miscellaneous

Accept-Ranges: bytes
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
X-Powered-By-Plesk: PleskWin

[Open in Google Docs Viewer](#)

[Open link in new tab](#)

[Open link in new window](#)

[Open link in new incognito window](#)

[Download file](#)

[Copy link address](#)

[Edit PDF File on PDFescape.com](#)

با تشکر.

[Open in Google Docs Viewer](#)

[Open link in new tab](#)

[Open link in new window](#)

[Open link in new incognito window](#)

[Download file](#)

[Copy link address](#)

[Edit PDF File on PDFescape.com](#)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۰۱

شما که از IIS7.5 استفاده می‌کنید یا از [این روش](#) یا از [روش‌های کش کردن اطلاعات](#) استفاده کنید.

نویسنده: میلاد
تاریخ: ۱۳۹۱/۰۷/۰۱

سلام و تشکر

از روش جناب صالحی هم استفاده کردم اما جواب نداد... سوالی که الان برام ایجاد شده اینه که آیا ممکنه هاست من استفاده از فشرده سازی رو غیرفعال کرده باشه؟

و اینکه آیا مگر کش کردن و فشرده سازی مقوله‌های جدایی نیستند؟!

با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۰۱

- ممکنه [این مورد](#) روی IIS7.5 شما فعال نباشد. باید با پشتیبانی هاست صحبت کنید.

- هستند. ولی اگر کش کردن را فعال کنید، کاربر یکبار که فایل استاتیک شما را دریافت کرد، بار بعد درخواستی را به سرور ارسال نمی‌کند. به همین جهت در کل بار کمتر و نتیجه بهتری نسبت به فشرده سازی دارد که هر بار باید اعمال شود.

نویسنده: میلاد
تاریخ: ۱۳۹۱/۰۷/۰۱

خیلی ممنون از صبر و شکیباییتون...

فقط یک سوال دیگه :

اگر این قابلیت از طرف هاست غیر فعال باشه ، روی کدهایی که درون رویداد Application_PreRequestHandlerExecute نوشته میشن هم تأثیر داره؟

اگر جواب منفی هست ، پس چرا به نظر میاد اصلاً این رویداد روی هاست اجرا نمیشه؟

ممنون

نوبنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۰۱ ۱:۱۰

- خیر.

- من دسترسی به سرور و کدهای شما برای دیباگ ندارم. از [ELMAH](#) استفاده کنید. شاید خطای در کار باشد که با ELMAH لایگ می‌شود. شاید اصلاً این روال فراخوانی نمی‌شود که بهتر است این نوع کدها را به یک [http module منتقل کرد](#) و نه اینکه مستقیماً داخل `global.asax.cs` نوشت.

نوبنده: شاهین کیانی
تاریخ: ۱۳۹۲/۱۱/۱۱ ۲:۴۴

سلام من از این کد چطوری استفاده کنم در برنامم؟

نوبنده: محسن خان
تاریخ: ۱۳۹۲/۱۱/۱۱ ۹:۳۹

همانطور که در متن نوشتند، این کدها رو باید به فایل `global.asax.cs` برنامه اضافه کنید. عیناً داخلش قرار بدید کار می‌کند.

در این مطلب می‌خواهیم شما را با نحوه بار گزاری ساعت و تاریخ سیستم سرور با استفاده از JQuery Ajax آشنا کنیم.

در بعضی از سایتها با استفاده از جاوا اسکریپت تاریخ و ساعت جاری سیستم کلاینت به او نشان داد می‌شود.

این روش یک مزیت دارد: اول اینکه این کدها سمت کلاینت اجرا می‌شون و برای سرور بار اضافی ایجاد نمی‌کنند.

و یک عیب هم دارد: در صورتی که ساعت و تاریخ روی سیستم کلاینت تنظیم نباشد، همین ساعت و تاریخ نادرست برای او نمایش داده می‌شود. همین عیب می‌تواند باعث افت کیفیت وب سایت شود.

اما راهی هست که تاریخ و ساعت سیستم سرور برای کاربر نشان داده شود و آن هم استفاده از JQuery Ajax هست. به صورتی که هر ثانیه درخواستی برای یک handler فرستاده می‌شود و آن handler نیز ساعت و تاریخ روی سرور را باز می‌گرداند و این مقدار بازگشته شده را می‌توان در تگی از صفحه وب نمایش داد.

مثال: ابتدا یک صفحه aspx می‌سازیم و تگ زیر را در آن قرار می‌دهیم:

```
<p id="datetime"></p>
```

ساعت و تاریخ بار شده از سرور در این تگ باید نشان داده شود.

سپس کدهای اسکریپت زیر را می‌نویسیم:

```
var auto_referesh = setInterval
(
    function()
    {
        $.post
        (
            "GetDateTime.ashx",
            function (result)
            {
                $('#datetime').html(result);
            }
        );
    }, 1000
);

);
```

با نوشتن این کدها هر ثانیه یک بار، بوسیله Ajax درخواستی برای یک handler به اسم GetDateTime.ashx فرستاده می‌شود. وظیفه این handler برگرداندن تاریخ و ساعت فعلی سیستم سرور است. بعد از دریافت مقدار این مقدار از این handler، آنرا در تگ با شناسه datetime قرار می‌دهیم.

کد استفاده شده در handler هم به این صورت است:

```
<%@ WebHandler Language="C#" Class="GetDateTime" %>

using System;
using System.Web;

public class GetDateTime : IHttpHandler {

    public void ProcessRequest (HttpContext context) {
        context.Response.ContentType = "text/plain";
        context.Response.Write(DateTime.Now.ToString());
    }

    public bool IsReusable {
```

```
get {
    return false;
}
}
```

در انتهای فایل ضمیمه این مثال را از این لینک دریافت کنید:

[AjaxDateTime.zip](#)

نظرات خوانندگان

نویسنده: صابر فتح الله
تاریخ: ۱۳۹۱/۰۶/۲۵ ۱۰:۰

سلام
روش شما خوبه اما عیبی که داره اینه که هر ثانیه ساعت از سرور میگیره و باعث سربار میشه به نظر من روش بهتر اینه از همون کد طرف کلاینت استفاده بشه اما زمان اولیه از سرور خونده بشه بعد طرف کلاینت این ساعت خودمون اضافه کنیم و ساعت شبیه سازی کنیم در صورتی هم که کاربر به صفحه دیگری بره یا صفحه رفرش کنه دوباره ساعت از سرور خونده میشه و الگوریتم ساعت طرف کلاینت به کار می‌افته به نظر من این روش بهینه‌تر هست حالا باز ببینیم دوستان چه نظری دارن موفق و موید باشید

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۱/۰۶/۲۵ ۱۰:۳۷

سلام
بله این روش سربار اضافی داره. روشنی که گفتید خیلی بهتر هست.
ولی خوب مسئله اینه که چطور میشه داده نوع DateTime در ASP.Net رو با jQuery خوند؟
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۵ ۱۰:۵۵

روش دوم: خود وب سرور هم با درخواست‌های Head [تاریخ رو](#) ارسال می‌کنه. به این صورت هم قابل خواندن است:

```
<script src="Scripts/jquery.min.js" type="text/javascript"></script>
<script type="text/javascript">
$(document).ready(function () {
    try {
        var result = $.ajax({ 'type': 'HEAD', 'url': '/' }).success(function () {
            var date1 = new Date(result.getResponseHeader('Date'));
            alert(date1);
        });
    }
    catch (err) {
        //...
    }
});</script>
```

و مهم‌ترین مزیتش این است که با تمام وب سرورهای استاندارد کار می‌کنه و فرقی نمی‌کنه کد شما PHP است یا ASP.NET + این رو هم باید درنظر داشت که حین پردازش تاریخ دریافتی از وب سرور باید مسایل GMT را هم لحاظ کرد تا تاریخ و زمان دریافتی با زمان ایران تطابق پیدا کند.

نویسنده: احمد احمدی
تاریخ: ۱۳۹۱/۰۶/۲۵ ۱۲:۵

سلام
همونطور که جناب فتح الله فرمودند، بهتر هست که یک بار زمان را از سرور دریافت کنید و سپس بروزرسانی را بوسیله جاوا اسکریپت انجام دهید.
بنده در ارتباط با این بحث، یک کلاس ساده نوشتم که می‌توانه عملیات بروزرسانی را انجام بده.

```

var MyTime = function () {
    var date;
    var tag;

    var init = function (hour, minute, seconds, tagId) {
        var constructor = getConstructorString(hour, minute, seconds);
        date = new Date(constructor);
        tag = document.getElementById(tagId);
        //console.log('MyTime : Init(%s, %s, %s, %s)', hour, minute, seconds, tagId);
    };

    var run = function () {
        update();
        window.setInterval(update, 1000);
        //console.log('MyTime : Run');
    };

    var update = function updateClock() {
        var h = date.getHours();
        var m = date.getMinutes();
        var s = date.getSeconds();

        s++;
        if (s == 60) { m++; s = 0; };
        if (m == 60) { h++; m = 0; };
        if (h == 13) h = 1;

        var constructor = getConstructorString(h, m, s);
        date = new Date(constructor);

        h = (h < 10) ? ("0" + h) : h;
        m = (m < 10) ? ("0" + m) : m;
        s = (s < 10) ? ("0" + s) : s;

        tag.innerHTML = h + ":" + m + ":" + s;
        //console.log('MyTime : update');
    };

    var getConstructorString = function (hour, minute, seconds) {
        //console.log('MyTime : getConstructorString');
        return '01/01/2000 ' + hour + ':' + minute + ':' + seconds;
    };

    return {
        Init: init,
        Run: run
    };
}

```

روش کار به این صورت هست که شما یک بار متد `Init` را به همراه پارامترهای (ساعت، دقیقه، ثانیه، آی دی تگ) فراخوانی می کنید، در نهایت برای اجرای ساعت، یک بار هم متد `Run` را فراخوانی می کنید.

: مثال

```

myTime = new MyTime();
myTime.Init(12, 59, 50, 'clock');
myTime.Run();

```

. امیدوارم مفید بوده باشه

تا قبل از ASP.NET 4.5، هنگام کار با GridView رسم بر این بوده که به خاصیت DataSource، یک منبع داده (مانند Bind و ...) را Bind کرده و متدهای DataBind را صدا نموده و نتیجه نمایش داده می‌شد. اما با استفاده از ویژگی‌های جدید اضافه شده (هر چند با تأخیر نسبت به Grid‌های پیشرفته دیگر) کار با این کنترل راحت‌تر و خواناتر شده است. یکی از این ویژگی‌ها را با هم بررسی می‌کنیم: با استفاده از ویژگی SelectMethod میتوان متدهای GridView معرفی کرد که وظیفه منبع داده را انجام داده و هنگام Bind فراخوانی شده و گرید را پر کند: مثال:

```
<asp:GridView ID="gvCities"
    runat="server"
    AutoGenerateColumns="False"
    ItemType="WebApplication3.City"
    SelectMethod="GetAllCities">
    <Columns>
        <asp:TemplateField HeaderText="نام">
            <ItemTemplate><%#: Item.Name %></ItemTemplate>
        </asp:TemplateField>
    </Columns>
</asp:GridView>
```

نکته مهم در این کد ItemType است. با استفاده از این خاصیت به جای اینکه مانند قبل نام فیلد‌هایی که قرار است در گرید نمایش داده شود را بصورت string معرفی کنیم (مثلاً در اینجا Eval("Name"))، اگر نام فیلد را غلط بنویسیم هنگام کامپایل خطای صادر نمی‌شود)، آنرا بصورت Strongly Type نوشته و از اشتباه جلوگیری می‌کنیم. (+) کد متدهای:

```
public IQueryable<City> GetAllCities()
{
    var context = new EFContext();
    var q = from c in context.City
            orderby c.Name
            select c;
    return q;
}
```

و سپس دستور زیر را فراخوانی می‌کنیم:

```
gvCities.DataBind();
```

اگر بخواهیم در گرید Paging داشته باشیم بصورت زیر عمل می‌کنیم:

```
<asp:GridView ID="gvCities"
    runat="server"
    AutoGenerateColumns="False"
    AllowPaging="True"
    PageSize="10"
    ItemType="WebApplication3.City"
    SelectMethod="GetAllCities">
    <Columns>
        <asp:TemplateField HeaderText="نام">
            <ItemTemplate><%#: Item.Name %></ItemTemplate>
        </asp:TemplateField>
    </Columns>
</asp:GridView>
```

که در اینجا دو خصوصیت AllowPaging و PageSize را مقدار دهی کرده ایم. این خصوصیت‌ها اجازه صفحه بندی را به گرید

می‌دهند. حال برای اینکه متد نیز برای صفحه بندی آماده شود باید سه آرگومان به آن اضافه کنیم: (نام پارامترها باید دقیقاً موارد زیر باشد)

- 1- نقطه شروع صفحه بندی را مشخص می‌کند.
- 2- تعداد سطرهایی که گردید باید نمایش دهد را مشخص می‌کند.
- 3- این پارامتر باید درتابع مقدار دهی شود (مانند مثال) تا مشخص شود نتیجه Query چند رکورد است و در نهایت گرید تعداد صفحات را بر این اساس نمایش می‌دهد.

و برای اینکه صفحه بندی را در Query هم لحاظ کنیم از دو تابع Skip و Take استفاده شده است.

```
public IQueryable<City> GetAllCities(int startRowIndex, int maximumRows, out int totalRowCount)
{
    var context = new EFContext();
    var q = from c in context.City
            select c;

    totalRowCount = q.Count();

    return q.OrderBy(x=>x.Name).Skip(startRowIndex).Take(maximumRows);
}
```

نکته مهم در این متد IQueryable بودن آن است که باعث واکشی داده‌ها بصورت صفحه به صفحه می‌شود.
دستورات SQL تولید شده در پروفایلر:

Object context #3 [/WebForm1.aspx]

Statements **Object context Usage**

Short SQL

```
SELECT ... FROM (Select COUNT(1) AS [A1] From [dbo].[City] AS [...]
SELECT ... FROM (Select [Extent1].[Id] AS [Id], [Extent1].[Name]... WHERE [Extent1].[row_number] > 10
```

Details **Stack Trace**

```
1 SELECT TOP (10) [Extent1].[Id] AS [Id],
2                   [Extent1].[Name] AS [Name]
3 FROM   (SELECT [Extent1].[Id] AS [Id],
4                   [Extent1].[Name] AS [Name],
5                   row_number() OVER (ORDER BY [Extent1].[Name] ASC) AS [row_number]
6                   FROM   [dbo].[City] AS [Extent1]) AS [Extent1]
7 WHERE  [Extent1].[row_number] > 10
8 ORDER  BY [Extent1].[Name] ASC
```

همانطور که مشاهده می‌کنید دو دستور SQL تولید شده، یکی برای بازگرداندن تعداد رکوردها و یکی هم برای واکشی داده‌ها به

اندازه تعداد رکوردهای مجاز در هر صفحه.

نظرات خوانندگان

نویسنده: saeid
تاریخ: ۱۳۹۱/۰۶/۳۰ ۱:۲۲

با سلام خدمت دوست عزیزم و تشکر از زحمت شما. سوالم این بود که واقعاً این مورد در ورژن‌های قبلی امکان پذیر نبود؟!
<http://books.google.com/books?id=wAsBOIZFXV4C&pg=PA338&lpg=PA338&dq>SelectMethod+gridview+asp.net+startRowIndex+maximumRows+totalRowCount&source=b1&ots=sA1GqNuQr9&sig=vz0aA6EQ6h1w3iOTQ45dppyMwtw&hl=en&sa=X&ei=yztaUKTFJYSJhQf7o4CoDQ&ved=0CFAQ6AEwBQ#v=onepage&q>SelectMethod%20gridview%20asp.net%20startIndex%20maximumRows%20totalRowCount&f=false>

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۳۰ ۸:۶

نه به صورت [Strongly Type Binding](#) که توضیح داده شد.

نویسنده: saeid
تاریخ: ۱۳۹۱/۰۶/۳۱ ۱۳:۲۴

بله درسته. ممنون از لینکی که به اشتراک گذاشتید. راستش مطلب رو بطور کامل نخونده بودم.

قابلیت **CompositeScript** پس از گذشت مدتی که از توسعه پروره مورد نظرتان می‌گذرد احتمالاً فایل‌های javascript زیادی در پروره شما استفاده می‌شود که هم مدیریت و هم بار سنجینی بر سرعت بارگذاری اولیه سایت شما به دلیل زمان بارگذاری فایل‌های javascript خواهد گذاشت. در ASP.NET چندین روش برای مدیریت فایل‌های css, javascript وجود دارد. ساده‌ترین روش استفاده از امکانات خود CompositeScript است. با قابلیت CompositeScript براحتی می‌توانید فایل‌های javascript خود را با هم ادغام و در یک فایل Client ScriptResource.axd برای ScriptManager استفاده نمایید. برای این کار کافیست از تگ جدید **CompositeScript** در کنترل ScriptManager استفاده نمایید. در کد زیر این قابلیت نمایش داده شده است:

```
<asp:ScriptManager ID="ScriptManager" runat="server" ScriptMode="Release">
  <CompositeScript>
    <Scripts>
      <asp:ScriptReference Path("~/Scripts/jquery-1.7.2.min.js") />
      <asp:ScriptReference Path "~/Scripts/jquery.ui.core.min.js" />
      <asp:ScriptReference Path "~/Scripts/jquery.ui.widget.min.js" />
      <asp:ScriptReference Path "~/Scripts/jquery.ui.mouse.min.js" />
      <asp:ScriptReference Path "~/Scripts/jquery.ui.sortable.min.js" />
    </Scripts>
  </CompositeScript>
</asp:ScriptManager>
```

علاوه بر آن حتماً خاصیت **ScriptMode** کنترل ScriptManager را بر روی **Release** تنظیم نمایید تا از حداکثر کارایی و کش فایل‌ها استفاده نمایید.

دسترسی به **ScriptManager** در صفحات دیگر و الصاق فایل‌های خاص

برخی از فایل‌های Javascript فقط در صفحات خاصی استفاده شده اند و لازم نیست در هر صفحه بارگذاری شود برای این کار کافی است فایل مورد نظر را در صفحه خاص به **ScriptManager** اضافه نمایید. البته ابتدا لازم است به **ScriptManager** دسترسی داشته باشیم. کد زیر نحوه دسترسی به آن را نمایش داده است:

```
ScriptManager scriptManager = ScriptManager.GetCurrent(this.Page);
scriptManager.CompositeScript.Scripts.Add(new ScriptReference("~/Scripts/jquery.json.min.js"));
```

ادامه دارد...

نظرات خوانندگان

نویسنده: ایمان اسلامی
تاریخ: ۸:۴۸ ۱۳۹۱/۰۷/۱۷

با سلام و تشکر از مطلب خوبتان
در مورد فایلهای css هم روشی وجود دارد؟

نویسنده: مجتبی کاویانی
تاریخ: ۹:۳۳ ۱۳۹۱/۰۷/۱۷

با استفاده از ScriptManager خیر! اما در مقاله‌های بعدی با روش دیگری توضیح خواهم داد.

نویسنده: میثم فغفوری
تاریخ: ۱۳:۱۵ ۱۳۹۲/۱۱/۱۲

ممnon عالی بود فقط یک سوال:
من با افزونه YSlow که بررسی میکنم ظاهرا فایل ایجاد شده رو کش میکنه به مدت ۱ سال حالا اگر تو این مدت تغییری تو این اسکریپت‌ها داده باشیم آپدیت میکنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۴۸ ۱۳۹۲/۱۱/۱۲

اگر به انتهای لینک آن دقت کنید، یک کوئری استرینگ اضافه شده. مقدار آن در حقیقت هش فایل‌های مورد استفاده است. با تغییری در این فایل‌ها، این هش هم تغییر خواهد کرد. مرورگر با مشاهده‌ی آدرس جدید، درخواست جدیدی را به سرور برای دریافت فایل‌های به روز شده ارسال می‌کند.

نویسنده: محمد رضا زا
تاریخ: ۱۳:۲۵ ۱۳۹۴/۰۱/۱۹

وقتی از مستر پیج استفاده می‌کنیم این ContentPlaceHolder رو تو تگ script manager هد باید اضافه کنیم؟ یا جای دیگه و بعد از استفاده کردن دیگه تو کل صفحات نیازی به فراخونی نیست؟ و در هر جایی می‌شه از فایل‌ای jquery استفاده کرد؟

پیشنازهای:

[چگونه با استفاده از لوسین مطالب را ایندکس کنیم؟](#)[چگونه از افزونه jQuery Auto-Complete استفاده کنیم؟](#)[نحوه استفاده صحیح از لوسین در ASP.NET](#)

اگر به جستجوی سایت دقت کرده باشید، قابلیت ارائه پیشنهاداتی به کاربر توسط یک Auto-Complete به آن اضافه شده است. در مطلب جاری به بررسی این مورد به همراه دو مثال Web forms و MVC پرداخته خواهد شد.



قسمت عمده مطلب جاری با پیشنازهای یاد شده فوق یکی است. در اینجا فقط به ذکر تفاوت‌ها بسته خواهد شد.

الف) دریافت لوسین

از طریق [NuGet](#) آخرین نگارش را دریافت و به پروژه خود اضافه کنید. همچنین Lucene.NET Contrib را نیز به همین نحو دریافت نمائید.

ب) ایجاد ایندکس

کدهای این قسمت با مطلب بر جسته سازی قسمت‌های جستجو شده، یکی است:

```
using System.Collections.Generic;
using System.IO;
using Lucene.Net.Analysis.Standard;
using Lucene.Net.Documents;
using Lucene.Net.Index;
using Lucene.Net.Store;
using LuceneSearch.Core.Model;
using LuceneSearch.Core.Utils;

namespace LuceneSearch.Core
{
    public static class CreateIndex
    {
        static readonly Lucene.Net.Util.Version _version = Lucene.Net.Util.Version.LUCENE_30;

        public static Document MapPostToDocument(Post post)
        {
            var postDocument = new Document();
            postDocument.Add(new Field("Id", post.Id.ToString(), Field.Store.YES,
Field.Index.NOT_ANALYZED));
            var titleField = new Field("Title", post.Title, Field.Store.YES, Field.Index.ANALYZED,
Field.TermVector.WITH_POSITIONS_OFFSETS);
        }
    }
}
```

```

        titleField.Boost = 3;
        postDocument.Add(titleField);
        postDocument.Add(new Field("Body", post.Body.RemoveHtmlTags(), Field.Store.YES,
Field.Index.ANALYZED, Field.TermVector.WITH_POSITIONS_OFFSETS));
        return postDocument;
    }

    public static void CreateFullTextIndex(IEnumerable<Post> dataList, string path)
    {
        var directory = FSDirectory.Open(new DirectoryInfo(path));
        var analyzer = new StandardAnalyzer(_version);
        using (var writer = new IndexWriter(directory, analyzer, create: true, mfl:
IndexWriter.MaxFieldLength.UNLIMITED))
        {
            foreach (var post in dataList)
            {
                writer.AddDocument(MapPostToDocument(post));
            }

            writer.Optimize();
            writer.Commit();
            writer.Close();
            directory.Close();
        }
    }
}
}

```

تنها تفاوت آن اضافه شدن `titleField.Boost = 3` می‌باشد. توسط خواهیم گفت که اهمیت عبارات ذکر شده در عناوین مطالب، بیشتر است از اهمیت متون آن‌ها.

ج) تهیه قسمت منبع داده Auto-Complete

```

namespace LuceneSearch.Core.Model
{
    public class SearchResult
    {
        public int Id { set; get; }
        public string Title { set; get; }
    }
}

```

```

using System.Collections.Generic;
using System.IO;
using Lucene.Net.Index;
using Lucene.Net.Search;
using Lucene.Net.Store;
using LuceneSearch.Core.Model;
using LuceneSearch.Core.Utils;

namespace LuceneSearch.Core
{
    public static class AutoComplete
    {
        private static IndexSearcher _searcher;

        /// <summary>
        /// Get terms starting with the given prefix
        /// </summary>
        /// <param name="prefix"></param>
        /// <param name="maxItems"></param>
        /// <returns></returns>
        public static IList<SearchResult> GetTermsScored(string indexPath, string prefix, int maxItems
= 10)
        {
            if (_searcher == null)
                _searcher = new IndexSearcher(FSDirectory.Open(new DirectoryInfo(indexPath)), true);

            var resultsList = new List<SearchResult>();
            if (string.IsNullOrWhiteSpace(prefix))
                return resultsList;
        }
    }
}

```

```
prefix = prefix.ApplyCorrectYeKe();

var results = _searcher.Search(new PrefixQuery(new Term("Title", prefix)), null, maxItems);
if (results.TotalHits == 0)
{
    results = _searcher.Search(new PrefixQuery(new Term("Body", prefix)), null, maxItems);
}

foreach (var doc in results.ScoreDocs)
{
    resultsList.Add(new SearchResult
    {
        Title = _searcher.Doc(doc.Doc).Get("Title"),
        Id = int.Parse(_searcher.Doc(doc.Doc).Get("Id"))
    });
}

return resultsList;
}
}
```

توضیحات:

برای نمایش Auto-Complete نیاز به منبع داده داریم که نحوه ایجاد آن را در کدهای فوق ملاحظه می‌کنید. در اینجا توسط جستجوی سریع لوسین و امکانات PrefixQuery آن، به تعدادی مشخص (`maxItems`)، رکوردهای یافت شده را بازگشت خواهیم داد. خروجی حاصل لیستی است از `SearchResult`ها شامل عنوان مطلب و `Id` آن. عنوان را به کاربر نمایش خواهیم داد؛ از `Id` برای هدایت او به مطلبی مشخص استفاده خواهیم کرد.

د) نمایش Auto-Complete در

```
using System.Text;
using System.Web.Mvc;
using LuceneSearch.Core;
using System.Web;

namespace LuceneSearch.Controllers
{
    public class HomeController : Controller
    {
        static string _indexPath = HttpRuntime.AppDomainAppPath + @"App_Data\idx";

        public ActionResult Index(int? id)
        {
            if (id.HasValue)
            {
                //todo: do something
            }
            return View(); //Show the page
        }

        public virtual ActionResult ScoredTerms(string q)
        {
            if (string.IsNullOrWhiteSpace(q))
                return Content(string.Empty);

            var result = new StringBuilder();
            var items = AutoComplete.GetTermsScored(_indexPath, q);
            foreach (var item in items)
            {
                var postUrl = this.Url.Action(actionName: "Index", controllerName: "Home", routeValues:
new { id = item.Id }, protocol: "http");
                result.AppendLine(item.Title + "|" + postUrl);
            }

            return Content(result.ToString());
        }
    }
}
```

@{

```

ViewBag.Title = "جستجو";
var scoredTermsUrl = Url.Action(actionName: "ScoredTerms", controllerName: "Home");
var bulletImage = Url.Content("~/Content/Images/bullet_shape.png");
}
<h2>جستجو</h2>
<div align="center">
    @Html.TextBox("term", "", htmlAttributes: new { dir = "ltr" })
    <br />
    را وارد نماید ۱۰ جهت آزمایش
</div>
@section scripts
{
    <script type="text/javascript">
        EnableSearchAutocomplete('@scoredTermsUrl', '@bulletImage');
    </script>
}

```

```

function EnableSearchAutocomplete(url, img) {
    var formatItem = function (row) {
        if (!row) return "";
        return "<img src='" + img + "' /> " + row[0];
    }

    $(document).ready(function () {
        $("#term").autocomplete(url, {
            dir: 'rtl', minChars: 2, delay: 5,
            mustMatch: false, max: 20, autoComplete: false,
            matchContains: false, scroll: false, width: 300,
            formatItem: formatItem
        }).result(function (evt, row, formatted) {
            if (!row) return;
            window.location = row[1];
        });
    });
}

```

توضیحات:

- ابتدا ارجاعاتی را به jQuery، افزونه Auto-Complete و اسکریپت سفارشی تهیه شده، در فایل layout پروژه تعریف خواهیم کرد.
- در اینجا سه قسمت را مشاهده می‌کنید: کدهای کنترلر، View متناظر و اسکریپتی که Auto-Complete را فعال خواهد ساخت.
- قسمت مهم کدهای کنترلر، دو سطر زیر هستند:

```

result.AppendLine(item.Title + " | " + postUrl);
return Content(result.ToString());

```

مطابق نیاز افزونه انتخاب شده در مثال جاری، فرمت خروجی مدنظر باید شامل سطرهایی حاوی متن قابل نمایش به همراه یک Id (یا در اینجا یک آدرس مشخص) باشد. البته ذکر این Id اختیاری بوده و در اینجا جهت تکمیل بحث ارائه شده است.

return Content هم سبب بازگشت این اطلاعات به افزونه خواهد شد.

- کدهای View متناظر بسیار ساده هستند. تنها نام TextBox تعریف شده مهم می‌باشد که در متدهای اسکریپتی ScoredTerms استفاده شده است. به علاوه، نحوه مقدار دهی آدرس دسترسی به اکشن متدهای EnableSearchAutocomplete نیز مهم می‌باشد.

- در متدهای اسکریپتی اسکریپتی افزونه Auto-Complete را ملاحظه می‌کنید.

جهت آن، به راست به چپ تنظیم شده است. با ۲ کاراکتر ورودی فعال خواهد شد با وقفه‌ای کوتاه. نیازی نیست تا انتخاب کاربر از لیست ظاهر شده حتیماً با عبارت جستجو شده صد درصد یکی باشد. حداقل ۲۰ آیتم در لیست ظاهر خواهند شد. اسکرول بار لیست را حذف کرده‌ایم. عرض آن به ۳۰۰ تنظیم شده است و نحوه فرمت دهی نمایشی آن را نیز ملاحظه می‌کنید. برای این منظور از متدهای formatItem استفاده شده است. آرایه row در اینجا در برگیرنده اعضای Title و Id ارسالی به افزونه است. اندیس صفر آن به عنوان دریافتی اشاره می‌کند.

همچنین نحوه نشان دادن عکس العمل به عنصر انتخابی را هم ملاحظه می‌کنید (در متدهای result مقدار دهی شده). window.location را به عنصر دوم آرایه row هدایت خواهیم کرد. این عنصر دوم مطابق کدهای اکشن متدهای شده، به آدرس یک صفحه اشاره می‌کند.

۵ نمایش Auto-Complete در ASP.NET WebForms

قسمت عمده مطالب فوق با وب فرم‌ها نیز یکی است. خصوصاً توضیحات مرتبط با متدها EnableSearchAutocomplete ذکر شده.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="LuceneSearch.WebForms.Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <title>جستجو</title>
    <link href="Content/Site.css" rel="stylesheet" type="text/css" />
    <script src="Scripts/jquery-1.7.1.min.js" type="text/javascript"></script>
    <script src="Scripts/jquery.autocomplete.js" type="text/javascript"></script>
    <script src="Scripts/custom.js" type="text/javascript"></script>
</head>
<body dir="rtl">
    <h2>جستجو</h2>
    <form id="form1" runat="server">
        <div align="center">
            <asp:TextBox runat="server" dir="ltr" ID="term"></asp:TextBox>
            <br />
            را وارد نمائید lu جهت آزمایش
        </div>
    </form>
    <script type="text/javascript">
        EnableSearchAutocomplete('Search.ashx', 'Content/Images/bullet_shape.png');
    </script>
</body>
</html>
```

```
using System.Text;
using System.Web;
using LuceneSearch.Core;

namespace LuceneSearch.WebForms
{
    public class Search : IHttpHandler
    {
        public static string _indexPath = HttpRuntime.AppDomainAppPath + @"\App_Data\idx";

        public void ProcessRequest(HttpContext context)
        {
            string q = context.Request.QueryString["q"];
            if (string.IsNullOrWhiteSpace(q))
            {
                context.Response.Write(string.Empty);
                context.Response.End();
            }

            var result = new StringBuilder();
            var items = AutoComplete.GetTermsScored(_indexPath, q);
            foreach (var item in items)
            {
                var postUrl = "Default.aspx?id=" + item.Id;
                result.AppendLine(item.Title + "|" + postUrl);
            }

            context.Response.ContentType = "text/plain";
            context.Response.Write(result.ToString());
            context.Response.End();
        }

        public bool IsReusable
        { get { return false; } }
    }
}
```

در اینجا بجای Controller از یک Generic handler استفاده شده است (Search.ashx).

```
result.AppendLine(item.Title + "|" + postUrl);
context.Response.Write(result.ToString());
```

در آن، عنوان مطالب یافت شده به همراه یک آدرس مشخص، تهیه و در Response نوشته خواهد شد.

کدهای کامل مثال فوق را از اینجا می‌توانید دریافت کنید:

[LuceneSearch.zip](#)

همچنین باید دقت داشت که پروژه MVC آن از نوع MVC4 است (VS2010) و فرض براین می‌باشد که IIS Express 7.5 را نیز پیشتر نصب کرده‌اید.

کلمه عبور فایل: dotnettips91

نظرات خوانندگان

نویسنده: محسن
تاریخ: ۱۳۹۱/۱۰/۰۷ ۱۲:۵۹

ممنون از این مطلب مفید. آیا با استفاده روش فوق میشه چند جدول رو با هم ایندکس کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۰۷ ۱۳:۳

بله. در اینجا مشکلی با ثبت اطلاعات از چندین جدول مختلف، در یک ایندکس وجود ندارد. برای مدیریت جستجوی بهتر روی آن‌ها یک روش این است که در متدهای `MapPostToDocument`، `fieldName` را به نام مثلاً `TableName` اضافه کنید تا بشود در حین جستجو از آن استفاده کرد.

نویسنده: محمد حبیبی
تاریخ: ۱۳۹۱/۱۰/۲۵ ۱۴:۴

من یک تکست باکس ایجاد کرم که از همین Autocomplete شما استفاده میکنه. منتها یک دکمه جستجو هم قرار دادم که با کلیک کردن روی اون دوباره از لوسین استفاده میکنه و چندین رکورد رو هم بازگشت میده.
اما چون باید یک سری رشته که نتیجه‌ی لوسین هست رو برگردانم از `return Content` استفاده میکنم ولی نتایج رو در یک صفحه‌ی جدید و بدون Master Page میاره.

نحوه‌ی نمایش نتایج اطلاعات لوسین در یک View که حاوی یک Master Page هم هست چطوریه؟
مثلاً نتایج زیر همون تکست باکس نمایش داده بشه.
آیا باید نوع اکشن چیز دیگه‌ای باشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۲۵ ۱۴:۲۱

نتیجه جستجوی لوسین مثلاً در مطلب فوق لیستی از `SearchResult`ها است. همین لیست را در اکشن متدهای `GetList` و `GetDetail` در `View` منتظر یک حلقه درست کرده و حاصل را به صورت دلخواهی فرمت کنید.
علت استفاده از `return Content` در مثال بالا، نیاز افزونه جی‌کوئری استفاده شده به خروجی ساده متنی است. در حالتهای دیگر از `return View` معمولی استفاده کنید.

نویسنده: مرتضی
تاریخ: ۱۳۹۱/۱۱/۰۵ ۱۲:۴۵

سلام آقای نصیری ممنون از اینکه اطلاعاتون رو در اختیار دیگران قرار میدید بnde از کدهای بالا تویه سایتم استفاده کردم و جستجوی سایت مثل ساعت داره کار میکنه فقط تویه ظاهرش به مشکل برخوردم بnde میخواهم اون box رو که اطلاعات توش نمایش داده میشه از سمت راست تکس باکس تراز بشه مثل همون عکسی که بالا گذاشتید از بnde از سمت چپ تراز شده و یه سوال دیگه، اون عکس `loading` رو هنگام جستجو نمایش نمیده آدرس دهی اون هم درسته البته تکس باس بnde position: absolute هستش نمیدونم ماله اینه که نمایش نمیده یا مثلاً میره یه قسمت دیدگی از صفحه. ممنون میشم راهنمایی کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۰۵ ۱۳:۱

این موارد را باید با اصلاح اسکریپت یا `css` مربوط به `auto-complete` مدیریت کنید.
مثلاً محل قرارگیری منوی بازشونده به صورت زیر مقدار دهی شده. این را در فایل `jquery.autocomplete.js` یافته و اصلاح کنید:
`left: offset.left - options.width + 125`

یا تصویر loading در css به نحو زیر تعیین شده:

```
.ac_loading
{
background: white url('Images/indicator.gif') left center no-repeat;
}
```

نویسنده: امیر
تاریخ: ۱۳۹۱/۱۱/۰۹ ۱۷:۴۲

سلام: من یه مشکل دارم وقتی بعد از جستجو گزینه‌ای رو انتخاب می‌کنم صفحه refresh می‌شه و متن انتخاب شده پاک می‌شه میشه لطفا منو راهنمایی کنید مرسی

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۰۹ ۱۸:۶

بله. به همین نحو طراحی شده. زمانیکه یک گزینه انتخاب می‌شود، سطر زیر، کاربر را به صفحه متناظر هدایت می‌کند:

```
window.location = row[1];
```

این سطر در سمت کلاینت، مساوی Response.Redirect سمت سرور است. می‌توانید اینجا متن انتخابی رو به صورت مثلاً یک کوئری استرینگ تعریف کنید و بعد در صفحه‌ای دیگر دریافت و نمایش بدهید.

نویسنده: محسن.د
تاریخ: ۱۳۹۱/۱۱/۱۷ ۱:۵

بسیار عالی بود. تنها یک مشکل برای من در ارتباط با کدهای آموزشی که برای استفاده از لوسین در اینترنت پیدا کردم وجود دارد و اون هم اینکه در اونها کلاس و توابع به صورت استاتیک تعریف و در کنترلرها و یا کلاس‌های لایه سرویس فراخوانی شدن. این مسئله باعث اشکال در نوشتن آزمون واحد برای کنترلرها و یا متدها نمی‌شے؟

آیا در استاتیک معرفی کردن کلاس و توابع علت و مزیت خاصی وجود داره؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۷ ۸:۵۵

- اگر از لایه سرویس استفاده می‌کنید که نهایتاً با یک سری اینترفیس در کنترلرها کار خواهد کرد.
- بله. توضیح دادم در مطلب «[نحوه استفاده صحیح از لوسین در ASP.NET](#)».

نویسنده: میهمان
تاریخ: ۱۳۹۱/۱۱/۲۱ ۲۰:۱۵

با سلام
امکانش هست که یه توضیح هم در مورد نحوه کار مستقیم با دیتابیس بدین؟
کدام قسمت‌ها باید تغییر کتد و کدام قسمت‌ها باید حذف و اضافه شوند
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۱ ۲۰:۳۰

لطفاً اولین پیشناز عنوان شده را مطالعه کنید.

نویسنده: میهمان

تاریخ: ۱۳۹۱/۱۱/۲۲

با سلام مجدد؛ تمامی پیشنبازهای این مقاله را مطالعه کردم و خط به خط کدهای پروژه ای را که زحمت کشیدید و تهیه کرده اید را بررسی کردم. شما در اولین پیشنباز عنوان شده فرموده بودید "کتابخانه Lucene مستقل است از منبع داده مورد استفاده و تنها اطلاعاتی با فرمت شیء Document معرفی شده به آن را می‌شناسد." و در پروژه، در ابتدا کل داده‌ها را به یک document تبدیل کردید و سپس آنها را به IndexWriter اضافه کردید. و سپس جستجوهای خود را روی این دیکشنری انجام می‌دادید.

حالا سوال من این است: من در ابتدا باید تمامی اطلاعات را از طریق عنوان شده به یک فایل دیکشنری تبدیل کنم و هر مطلب جدید که اضافه می‌شود، به این دیکشنری نیز اضافه شود؟ اگر تعداد رکوردها بالای میلیون باشد؟ راهی وجود ندارد که ما مستقیم به خود دیتابیس کار کنیم؟ و یا نه من ابتدا باید روی دیتابیس کلمه مورد نظر را جستجو کتم و پس از یافتن آن، آنرا به دیکشنری اضافه کنم؟ بهترین راه چیست که برای داده‌های بالا جوابگو باشد

با تشکر فراوان

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۱۱/۲۲

- لوسین مستقل است از بانک اطلاعاتی. همچنین یکبار باید این ایندکس را تهیه کنید. اگر تعداد رکوردهای شما بالا است، فقط همان بار اول است که کار تهیه زمانبر خواهد بود. برای دفعات بعد در حد اضافه کردن چند سند لوسین به آن یا به روز رسانی و حذف است و کار دیگری ندارد.

- پس از تهیه ایندکس، جستجوی لوسین کاری به بانک اطلاعاتی شما ندارد. بر روی ایندکس خودش انجام می‌شود و نیازی به جستجوی مجدد در بانک اطلاعاتی شما نیست. یک سیستم مستقل است.

این روش متداول کار با لوسین است و حالت دیگری هم ندارد. این مستقل بودن هم یک مزیت است. برای مثال SQL Server CE یا خیلی از بانک‌های اطلاعاتی دیگر Full Text Search توکار ندارند. اینجا لوسین خوب جواب می‌دهد. ضمن اینکه من در یک دمو استفاده از لوسین برای ایندکس کردن کل اطلاعات ویکی‌پدیا رو دیدم. تهیه ایندکس آن یک روز کار برده بوده (با توجه به حجم اطلاعات بالای ویکی‌پدیا)، اما جستجوی آن فوق العاده سریع و با کیفیت بود. این ویدیو رو در اینجا می‌توانید مشاهده کنید:

[Full-text search with Lucene and neat things you can do with it](#)

نویسنده: morteza

تاریخ: ۱۳۹۱/۱۲/۰۱

سلام آقای نصیری میخواستم ببینم آیا شما اطلاعی از نحوه غیر فعل کردم اینتر تهیه این پلاگین دارید بنده میخوام وقتی کاربر اینتر رو فشار داد جستجوی معمولی انجام بشه و اگه خواست از مواردی که توسط autocomplete براش آمده توسط موس یکی رو انتخاب کنه و به صفحه مریبوطه هدایت بشه ممنون.

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۱۲/۰۱

کلید enter با case KEY.RETURN در این افزونه جی‌کوئری مدیریت می‌شود.

نویسنده: علیرضا پایدار

تاریخ: ۱۳۹۱/۱۲/۲۰

یک سوال خدمت شما داشتم اینکه :

وقتی بخوایم از 2 جدول متفاوت جستجو کنیم به نظر شما اطلاعات هر جدول را جداگانه ایندکس کنیم (منظورم همان کاری که متد CreateFullTextIndex شما انجام میده) یا خیر؟
البته منظور دو جدولی که با هم رابطه دارند.

چندتا لینک در مورد لوسین:

<http://www.thebestcsharpprogrammerintheworld.com/blogs/how-to-create-and-search-a-lucene-net-index-in-4-simple-steps-using-c-sharp-step-1.aspx>

<http://www.codeproject.com/Articles/272309/Lucene-Search-Programming>

<http://www.codeproject.com/Articles/320219/Lucene-Net-ultra-fast-search-for-MVC-or-WebForms>

ممnon

نویسنده: وحید نصیری
تاریخ: ۱۱:۳۲ ۱۳۹۱/۱۲/۲۰

من همه رو داخل یک ایندکس ثبت می‌کنم. فقط یک فیلد اضافه‌تر به نام «نام جدول» مورد نظر نیاز هست تا بشود روی آن کوئری خاص گرفت یا اینکه کلا روی تمام رکوردها جستجو کرد به یکباره.

نویسنده: مهمان
تاریخ: ۷:۵۵ ۱۳۹۲/۰۳/۱۹

ضمن خسته نباشد
اگر نیاز داشته باشیم که بعد از انتخاب گزینه کاربر به صفحه دیگه هدایت نشه و فقط آیتم انتخابی در تکست باکس بمونه باید چه کاری انجام داد. مرسى

نویسنده: وحید نصیری
تاریخ: ۱۷:۱۳ ۱۳۹۲/۰۳/۱۹

سطر window.location رو حذف کنید.

نویسنده: imo0
تاریخ: ۱۰:۵۵ ۱۳۹۲/۰۷/۲۷

سلام؛ تویه اولین پیشنيازها که کار با لوسین رو گفته بودين توابعی مثله SearchByPartialWords و Query و غيره ايجاد شده ولی تو اينجا يه جور دیگه جستجو کردين توی ايندکسها . میخواستم بدونم او توابع رو مگه واسه سرج تو ايندکسها ننوشتin؟ تازه اينجا شما دارين متن Title رو ميفرستين سمت کاربر . اين highlight کردن عبارت جستجو شدش پس چي شد؟

نویسنده: وحید نصیری
تاریخ: ۱۱:۳۵ ۱۳۹۲/۰۷/۲۷

- اين هم يك روش ديگر هست. از آن روش‌های ذكر شده در پیشنيازها هم می‌توانيد استفاده کنيد.
- highlight کردن عبارت جستجو شد در عنوان بازگشت داده شده، توسط افزونه auto-complete انجام می‌شود؛ خودکار است.

نویسنده: imo0
تاریخ: ۱۱:۴۶ ۱۳۹۲/۰۷/۲۷

تو اون تابع Query که نوشتin يه دونه highlight ايجاد کردين . اگه توسط auto-complete خودکار انجام ميشه پس نقش اين چيه؟ ميشه بگين کدام روش بهتره ؟ يا فرقی نميکنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۲۷ ۱۱:۵۲

دو بحث وجود دارد:

- بر جسته سازی قسمتی از عبارت جستجو شده در لیست نمایش داده شده توسط افزونه auto-complete. این مورد خودکار است و توسط افزونه انجام می‌شود.

- بر جسته سازی قسمتی از عبارت جستجو شده در نتایج یک جستجوی کامل بعدی که قرار است highlight آن توسط ما با کدنویسی خاصی انجام شود. مراجعه کنید [به این مطلب](#) برای توضیحات بیشتر.

نویسنده: ایمان اسلامی
تاریخ: ۱۳۹۳/۰۲/۲۲ ۱۵:۵۶

با سلام

من از لوسین در asp.net web form استفاده کردم و کاملا هم راضی هستم . فقط یه مشکلی داشتم. اگه بخوام غیر از عبارت مورد نظر برای جستجو یه مقدار دیگه رو هم بفرستم باید چکار کنم؟
dropdown مثلا به داشته باشیم و بخوایم مثلا مقدار categoryid query را هم پاس بدیم تا به مربوطه اضافه بشه. من رو اضافه کردم اما مقدارش رو نمیشه در هندر مربوط به Search دریافت کرد مثل `q`.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۲۲ ۱۷:۲۶

افزونه‌ی مورد استفاده در آن به نام `jquery.autocomplete.js` سورسش پیوست هست. `ajax` را در آن جستجو کنید به مورد ذیل خواهید رسید:

```
$.ajax({
  //...
  data: $.extend({
    q: lastWord(term),
    limit: options.max
  }, extraParams),
  //...
});
```

در اینجا `q` مشخص است و آن متغیر. بنابراین برای ارسال اطلاعات اضافی باید `extraParams` را مانند کدهای ذیل مقدار دهی کرد. نحوه‌ی دریافت عناصر آن در سمت سرور، مانند نحوه‌ی دریافت `q` است.

```
$(document).ready(function () {
  $("#term").autocomplete(url, {
    // ....
    extraParams: { id: 12, xyz: "test" }
  }).result(function (evt, row, formatted) {
    // ....
  });
});
```

نویسنده: ایمان اسلامی
تاریخ: ۱۳۹۳/۰۳/۰۷ ۱۸:۱۱

اگر بخوام لیستی که برای نتیجه جستجو باز میشه رو `Custom` کنم مثلا یه عکس بهش اضافه کنم یا یه جدول ساخته شده با HTML رو توش نشون بدم با یه سری جزئیات ، آیا امکان پذیره؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۰۷ ۱۹:۳۰

نحوه سفارشی سازی هر آیتم لیست را در مطلب فوق، در متدهای `formatItem` می‌توانید مشاهده کنید.

نویسنده: علیرضا
تاریخ: ۱۳۹۳/۰۴/۰۳ ۱۰:۵۸

با سلام؛ تکست باکسی که می‌خام جستجو از طریق اون انجام بشه در یک مستر پیج قرار داره و زمانی که صفحه و مستر پیج در یک پوشه قرار دارند مشکل نیست ولی اگر صفحه در یک پوشه دیگر قرار گیرد جواب نمی‌ده. توضیح اینکه از متدهم استفاده کردم ولی جواب نداد:

```
EnableSearchAutocomplete('<%=ResolveClientUrl("~/Handlers/find.ashx")%>','<%=ResolveClientUrl("Images/bullet_shape.png")%>');
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۰۳ ۱۱:۱۴

- نحوه استفاده از افزونه Firebug برای دیباگ برنامه‌های ASP.NET مبتنی بر jQuery
- بررسی کنید آیا ID جعبه متنی در آن صفحه، term است یا چیز دیگری؟ اگر تغییر کرده، ClientIDMode=Static بهتر است تنظیم شود.

نویسنده: علیرضا
تاریخ: ۱۳۹۳/۰۴/۰۳ ۱۸:۳۱

با تشکر از پاسخ شما، id درست بود و با فایرباگ هم بررسی کردم و خطای زیر را می‌داد:

```
TypeError: $(...).autocomplete is not a function
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۰۳ ۱۸:۵۴

این خطا یعنی اسکریپت‌های اصلی به صفحه پیوست نشده‌اند. برای تعریف آن‌ها هم از ResolveClientUrl استفاده کنید:

```
<script type="text/javascript" src='<%: ResolveClientUrl("~/path....") %>'></script>
```

نویسنده: محمد زارع
تاریخ: ۱۳۹۳/۱۰/۱۲ ۱۰:۲۸

سلام. من برای ثبت چندتا جدول مختلف در یک ایندکس فیلد TableId هم گرفتم برای اینکه مشخص بشه سند مربوط به کدام جدوله. حالا راهی هست که به جز Id فیلد دیگه ای هم برای بروزرسانی یا حذف یک سند ارسال کرد؟ مثلاً با Id و TableId یه سند رو حذف یا بروزرسانی کرد نه با Id خالی. یعنی با این دستور:

```
indexWriter.DeleteDocuments(new Term("Id", id.ToString()))
```

مثلاً یه محصول داشته باشیم و یه مقاله با آیدی یکسان چجوری میشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۱۲ ۱۰:۴۱

متدهم یک کوئری هم قبول می‌کند:

```
var query = new BooleanQuery();
query.Add(new BooleanClause(new TermQuery(new Term("id", id.ToString(CultureInfo.InvariantCulture))), Occur.SHOULD));
// query.Add...
// query.Add...
// ...
writer.DeleteDocuments(query);
```

نوبتند: آرش مصیر
تاریخ: ۱۳۹۳/۱۲/۲۸ ۱۲:۱۹

با سلام

آیا در حال حاضر این روش همچنان روش بهینه و مناسب برای جستجوی auto-complete است؟ یا روش‌های جدیدتری هم وجود دارد؟ (تاریخ انتشار این کتابخانه 2012 بوده است)
با تشکر

نوبتند: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۲۸ ۱۲:۲۴

بله. نسخه 4 آن هنوز نهایی نشده است: « [نگاهی به وضعیت کتابخانه Lucene.NET](#) »

نوبتند: محمدرضا نجفی پور
تاریخ: ۱۳۹۴/۰۳/۱۰ ۷:۴۲

من از این روش استفاده کردم. یه مشکلی که هست، "ک" و "ی" و "ه" رو که تایپ می‌کنم به عنوان خروجی چیزی بهم نمیده. با اینکه توی عنوان مطالب سایت کلمات مرتبط هستند.

نوبتند: وحید نصیری
تاریخ: ۱۳۹۴/۰۳/۱۰ ۹:۳۹

- یکبار لیست stop words معرفی شده را بررسی کنید. full text search کلمات و یا حروفی هستند که در ایندکس نخواهند شد.
- بحث یکسان سازی [ی](#) و [ک](#) فارسی و عربی را بررسی کنید.
- همچنین با استفاده از برنامه‌ی [luke](#) بررسی کنید که چه چیزی ایندکس شده.

در سایت جاری [مطالب زیادی](#) درباره ASP.NET MVC نوشته شده است. این مطلب و قسمت بعدی آن مروری خواهد داشت بر .ASP.NET MVC ها در Best Practice

استفاده از NuGet Package Manager برای مدیریت وابستگی‌ها

درباره اهمیت NuGet برای مصرف کنندگان قبل این مطلب نوشته شده است. بجای صرف وقت برای اینکه بررسی کنیم آیا این نسخه‌ی جدید کتابخانه‌ی X یا اسکریپت jQuery آمده است یا خیر، می‌توان این وظیفه را به NuGet سپرد. علاوه بر این NuGet مزیت‌های دیگری هم دارد؛ مثلاً تیم‌های برنامه نویسی می‌توانند کتابخانه‌های مشترک خودشان را در مخزن‌های سفارشی NuGet قرار دهند و توزیع و Versioning آن را به NuGet بسپارند.

تکیه بر Abstraction (انتزاع)

Abstraction در طراحی سیستم‌ها منجر به تولید نرم افزار‌های [Loosely coupled](#) با قابلیت نگهداری بالا و همچنین فراهم شدن زمینه برای نوشتمن Unit Test می‌شود.

اگر به مطالب قبلی وب سایت برگردیم در مطلب [ASP.NET MVC چرا گفته شد که :](#)

2) دستیابی به کنترل بیشتر بر روی اجزای فریم ورک :

در طراحی ASP.NET MVC همه جا interface‌ها قابل مشاهد هستند. همین مساله به معنای افزونه پذیری اکثر قطعات تشکیل دهنده ASP.NET MVC است؛ برخلاف ASP.NET web forms. برای مثال تابحال چندین view engine, routing engine وغیره توسط برنامه نویس‌های مستقل برای ASP.NET MVC طراحی شده‌اند که هیچ‌کدام با ASP.NET web forms میسر نیست. برای مثال از view engine پیش فرض آن خوشتان نمی‌آید؟ [عوضش کنید!](#) سیستم اعتبار سنجی توکار آن را دوست ندارید؟ آن را با یک نمونه بهتر تعویض کنید و الی آخر ...

به علاوه طراحی بر اساس interface‌ها یک مزیت دیگر را هم به همراه دارد و آن هم ساده سازی [mocking](#) (تقلید) آن‌ها است جهت ساده سازی نوشتمن آزمون‌های واحد.

از کلمه‌ی کلیدی New استفاده نکنید

هر جا ممکن است کار و هله سازی اشیاء را به لایه و حتی Framework دیگری بسپارید. هر زمان اشیاء نرم افزار خودتان را با کلمه‌ی new و هله سازی می‌کنید اصل Abstraction را فراموش کرده اید. هر زمان اشیاء نرم افزار را مستقیم و هله سازی می‌کنید در نظر داشته باشید می‌توانید آنها را به صورت وابستگی تزریق کنید.

در همین رابطه مطالب زیر را از دست ندهید :

[تزریق وابستگی \(dependency injection\)](#) به زبان ساده [تزریق وابستگی \(Dependency Injection\)](#) و توسعه پذیری

از HttpContext مستقیماً استفاده نکنید (از HttpContextBase استفاده کنید)

از 4.NET به بعد فضای نامی تعریف شده که در بر دارنده کلاس‌های انتزاعی (Abstraction) خیلی از قسمت‌های اصلی ASP.NET است. یکی از مواردی که در توسعه‌ی ASP.NET معمولاً زیاد استفاده می‌شود، شیء HttpContext است. استفاده از HttpContext را به استفاده از HttpContextBase ترجیح دهید. اهمیت این موضوع در راستای اهمیت انتزاع (Abstraction) می‌باشد.

از "رشته‌های جادویی" اجتناب کنید

استفاده از رشته‌های جادویی در خیلی از جاها کار را ساده می‌کند؛ بعضی وقت‌ها هم به آنها نیاز است اما مشکلات زیادی دارند :

رشته‌ها معنای باطنی ندارند (مثلاً : دشوار است که از روی نام یک ID مشخص کنم این ID چطور به ID دیگری مرتبط است و یا اصلاً ربط دارد یا خیر)

با اشتباهات املایی یا عدم رعایت حروف بزرگ و کوچک ایجاد مشکل می‌کنند.

به واکنش خوبی نشان نمی‌دهند. (برای درک بهتر [این مطلب](#) را بخوانید).

برای درک بهتر 2، یک مثال بررسی می‌شود؛ اولی از رشته‌های جادویی برای دسترسی به داده در ViewData استفاده می‌کند و دومی refactor شده‌ی مثال اول است که از strongly type مدل برای دسترسی به همان داده استفاده می‌کند.

```
<p>
    <label for="FirstName">First Name:</label>
    <span id="FirstName">@ViewData["FirstName"]</span>
</p>
```

مثال دوم :

```
<p>
    <label for="FirstName">First Name:</label>
    <span id="FirstName">@Model.FirstName</span>
</p>
```

مثلاً مثال دوم مشکلات رشته‌های جادویی را ندارد.
در رابطه با [این مطلب](#) Magic strings را مطالعه بفرمایید.

از نوشتن HTML در کدهای "Backend" خودداری کنید

با توجه به اصل جداسازی وابستگی‌ها (Separation of Concerns) وظیفه‌ی کنترلرها و دیگر کدهای backend رندر کردن HTML نیست. (ساده سازی کنترلرها) البته در نظر داشته باشید که قطعاً تولید HTML در متدهای کمکی کلاس‌هایی که " تنها" وظیفه‌ی آنها کمک به View‌ها جهت تولید کد هست ایرادی ندارد. این کلاس‌ها بخشی از View در نظر گرفته می‌شوند نه کدهای "backend".

در View‌ها Business logic انجام ندهید

معکوس بند قبلی هم کاملاً صدق می‌کند ، منظور این است که View‌ها تا جایی که ممکن است باید حاوی کمترین Business logic ممکن باشند. در واقع تمرکز View‌ها باید استفاده و نحوه‌ی نمایش داده ای که برای آن‌ها فراهم شده باشد نه انجام عملیات روی آن.

استفاده از Presentation Model را به استفاده مستقیم از Business Object‌ها ترجیح دهید

در مطالب مختلف وب سایت اشاره به اهمین ViewModel‌ها شده است. برای اطلاعات بیشتر بند ج [آموزش 11](#) از سری آموزش‌های ASP.NET MVC را مطالعه بفرمایید.

If‌های شرطی را در View‌ها در متدهای کمکی کپسوله کنید

استفاده از شرط‌ها در View کار توسعه دهنده را برای یک سری اعمال ساده می‌کند اما می‌تواند باعث کمی کثیف کاری هم شود.
مثال:

```
@if(Model1.IsAnonymousUser) {
    
} else if(Model1.IsAdministrator) {
    
```

```
} else if(Model.Membership == Membership.Standard) {
    
} else if(Model.Membership == Membership.Preferred) {
    
}
```

می‌توان این کد که تا حدودی شامل منطق تجاری هم هست را در یک متد کمکی کپسوله کرد :

```
public static string UserAvatar(this HtmlHelper<User> helper)
{
    var user = helper.ViewData.Model;
    string avatarFilename = "anonymous.jpg";
    if (user.IsAnonymousUser)
    {
        avatarFilename = "anonymous.jpg";
    }
    else if (user.IsAdministrator)
    {
        avatarFilename = "administrator.jpg";
    }
    else if (user.Membership == Membership.Standard)
    {
        avatarFilename = "member.jpg";
    }
    else if (user.Membership == Membership.Preferred)
    {
        avatarFilename = "preferred_member.jpg";
    }
    var urlHelper = new UrlHelper(helper.ViewContext.RequestContext);
    var contentPath = string.Format("~/content/images/{0}", avatarFilename);
    string imageUrl = urlHelper.Content(contentPath);
    return string.Format("<img src='{0}' />", imageUrl);
}
```

اکنون برای نمایش آواتار کاربر به سادگی می‌توان نوشت :

```
@Html.UserAvatar()
```

به این ترتیب کد ما تمیزتر شده ، قابلیت نگهداری آن بالاتر رفته ، منطق تجاری یک بار و در یک قسمت نوشته شده از این کد در جاهای مختلف سایت می‌توان استفاده کرد و اگر لازم به تغییر باشد با تغییر در یک قسمت همه جا اعمال می‌شود.

منتظر قسمت بعدی باشید.

نظرات خوانندگان

نویسنده: محسن.د

تاریخ: ۱۳۹۱/۰۸/۱۲ ۱۹:۴۵

نکات جالبی بود.

یک نکته که به شخصه اون رو تجربه کرده ام (خصوص در مورد استفاده از best practice) وضعیتی است که اصطلاحاً بهش overkilling میگن (اگر اشتباه نکنم) ، یعنی بعضی وقتاً دیگه زیاده روی میشه . مثلا در به کاربردن اینترفیسها و یا loose coupling .

برای مثال در مورد تشخیص زمان استفاده از اینترفیس [در این کتاب](#) ، یکی از بهترین راهکارها استفاده از آزمون واحد معرفی شده

بسیار عالی میشه در صورت امکان در این مورد هم نکاتی را ذکر کنید .

نویسنده: افشار محبی

تاریخ: ۱۳۹۱/۰۸/۲۴ ۸:۲۶

خیلی مفید و عملی بود. ممنون.

نویسنده: سیروان عفیقی

تاریخ: ۱۳۹۱/۰۸/۲۴ ۱۴:۱۹

عالی بود ممنون،

بهتر نیست به جای این if و else ها از [Dictionary](#) استفاده کنیم؟

نویسنده: شاهین کیاست

تاریخ: ۱۳۹۱/۰۸/۲۴ ۱۴:۲۰

خواهش می کنم.

چرا همینطوره.

کدها همه مثال هستند و هدف این مطلب چیز دیگری بود.

هر از چندگاهی که به لاغ‌های خطای برنامه مراجعه می‌کنم، درخواست‌هایی (حملاتی) با این مشخصات ثبت شده:

```
REQUEST_METHOD: OPTIONS
REQUEST_METHOD: PROPFIND
HTTP_USER_AGENT: Microsoft-WebDAV-MiniRedir/6.1.7600
SCRIPT_NAME: /ipc$
```

برای بستن این نوع درخواست‌های ویژه (که عموماً برای دسترسی به اطلاعات شیرپوینت و یا سرور بکار می‌روند)، فقط کافی است فایل web.config برنامه را به نحو زیر اصلاح کنیم:

```
<system.web>
  <httpHandlers>
    <add path="*" verb="OPTIONS, PROPFIND" type="System.Web.DefaultHttpHandler" />
  </httpHandlers>
</system.web>

<system.webServer>
  <handlers>
    <add name="propf" path="*" verb="OPTIONS, PROPFIND" type="System.Web.DefaultHttpHandler" />
  </handlers>
</system.webServer>
```

نظرات خوانندگان

نویسنده: الهام
تاریخ: ۱۳۹۱/۰۸/۱۵ ۸:۴۳

سلام آقای نصیری

ببخشید میشه توضیح بدین این اصلاحیه چه عملی انجام میده؟ به نظر میرسه این نوع درخواستها که هنوز هندل میشن

با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۱۵ ۸:۵۱

بله. پیغام "Not Implemented 501" را به درخواست‌های PROPFIND ارائه می‌دم.

موقع بسیاری پیش می‌آید که در زمان کار با یک نرم افزار تحت وب زمان اشکال زدایی پیش می‌آید که به دلیل موجود بودن داده در حافظه کش برنامه نمی‌تواند داده‌های واقعی را ببیند و داده‌های موجود در حافظه کش را مشاهده می‌کند (بیشتر مواقعی که از طریق بانک اطلاعاتی مستقیماً اقدام به حذف و اضافه داده می‌کنیم) در این بخش یک کلاس آماده کرده‌ام که همیشه خودم در نرم افزارهای استفاده می‌کنم.

شما می‌توانید این کلاس را به یک GridView یا کنترل‌های دیگر بایند کرده و کلیدهای موجود در حافظه کش را مشاهده کنید، و در صورتی که خواستید یک کلید خاص را از حافظه کش حذف نمایید (البته این کلاس بیشتر برای مدیر نرم فزار کاربرد دارد).

می‌توانید فایل مورد نظر را از طریق لینک [کلاس کمکی جهت مشاهده آیتم‌های موجود در حافظه کش و حذف آنها](#) دانلود نمایید.
در کلاس زیر هر کدام از قسمت‌ها را شرح می‌دهیم.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Web;
using System.Web.Caching;

namespace PWS.BLL
{
    /// <summary>
    /// آیتم‌های حافظه کش
    /// </summary>
    [DataObject(true)]
    public class CacheItems
    {
#region Constructors (2)

        /// <summary>
        /// سازنده اصلی
        /// </summary>
        /// <param name="cacheItem">عنوان آیتم ذخیره شده در حافظه کش</param>
        public CacheItems(String cacheItem)
        {
            CacheItem = cacheItem;
        }

        /// <summary>
        /// سازنده پیش فرض
        /// </summary>
        public CacheItems(){}
#endregion Constructors

#region Properties (2)

        /// <summary>
        /// کش کانتکست جاری
        /// </summary>
        /// <value>
        /// The cache.
        /// </value>
        private static Cache Cache
        {
            get {return HttpContext.Current.Cache; }
        }

        /// <summary>
        /// عنوان آیتم ذخیره شده در حافظه کش
        /// </summary>
        public String CacheItem{ get; set;}
#endregion Properties

#region Methods (4)
```

کلاس کمکی جهت مشاهده آیتم های موجود در حافظه کش و حذف آنها

```
// Public Methods (3)

    /// <summary>
    /// لیست تمام آیتم های ذخیره شده در حافظه کش
    /// </summary>
    /// <returns></returns>
    public List<CacheItems> GetCaches()
    {
        var items = new List<CacheItems>();
        بازیابی کل کلیدهای موجود در حافظه کش و اضافه کردن آن به لیست مربوطه//
        var enumerator = Cache.GetEnumerator();
        while (enumerator.MoveNext())
        {
            items.Add(new CacheItems(enumerator.Key.ToString()));
        }
        return items;
    }

    /// <summary>
    /// حذف آیتم جاری از حافظه کش
    /// </summary>
    public void RemoveItemFromCache()
    {
        RemoveItemFromCache(CacheItem);
    }

    /// <summary>
    /// حذف کردن یک آیتم از حافظه کش
    /// </summary>
    /// <param name="key"></param>
    public static void RemoveItemFromCache(string key)
    {
        PurgeCacheItems(key);
    }

// Private Methods (1)

    /// <summary>
    /// حذف کردن یک ایتم از حافظه کش با پیشوند وارد شده
    /// </summary>
    /// <param name="prefix"></param>
    private static void PurgeCacheItems(String prefix)
    {
        prefix = prefix.ToLower();
        var itemsToRemove = new List<String>();
        لیست آیتم های موجود در حافظه کش//
        var enumerator = Cache.GetEnumerator();
        while (enumerator.MoveNext())
        {

            if (enumerator.Key.ToString().ToLower().StartsWith(prefix))
                itemsToRemove.Add(enumerator.Key.ToString());
        }
        لیست مورد نظر را بیمایش کرده و گزینه های آن را از حافظه کش حذف می کنیم//
        foreach (var itemToRemove in itemsToRemove)
            Cache.Remove(itemToRemove);
    }

#endregion Methods
}
```

می کنیم

در صورتی که کلید مورد نظر با پارامتر وارد شده شروع شده باشد آن را به یک لیست اضافه//

موفق و موید باشید

نظرات خوانندگان

نویسنده: سی شارپ 2012
تاریخ: ۱۳۹۲/۰۸/۲۵ ۱۳:۲۶

چگونه از این کلاس استفاده کنیم ؟
لطفا مثال بزنید

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۸/۲۶ ۱۲:۹

به نظرم کاملا واضحه
شما می توانید متد GetCaches را به یک گرید ویو بایند کنید.
با این کار شما لیست ایتم های موجود در کش دارید
در مرحله بعد برای حذف هر ایتم از کش می توانید ایتم انتخاب شده را به تابع PurgeCacheItems(String prefix) پاس داده و
ایتم مورد نظر حذف کنید.
کد نوشته شده چیز چندان پیچیده ای نداره اگر م باز مشکل دارید لطفا مطرح کنید
موفق و موید باشید

همانگونه که اطلاع دارید یکی از روش‌های سرقت اطلاعات استفاده از نرم افزارهای جاسوس صفحه کلید ([Key Logger](#)) است، البته ثبت کلیدهای فشرده شده می‌تواند توسط سخت افزارهایی که سر راه سوکت صفحه کلید و کیس قرار می‌گیره، انجام بشه. در صورتی که چنین سخت افزاری (مخصوصاً در کافی نت‌ها) روی کامپیوتر کاربر نصب باشه، یا توسط ویروس و [بدافزارها](#) اینگونه نرم افزارهایی روی سیستم کاربر قرار بگیره هر کلیدی که توسط کاربر روی صفحه کلید فشرده می‌شده توسط اینها ثبت شده و در موقع مناسب برای فرد سازنده به طرقی (ایمیل یا ارتباط از طریق [برنامه‌های مبتنی بر سوکت](#)) حتی بسیاری از این برنامه‌ها پا را فراتر گذاشته و عنوان پنجره ای که کلیدها در آن فشرده شده نیز ثبت می‌شود (توسط توابع API ویندوز- البته اگر دوستان مایل باشن و از نظر مدیریت سایت ایرادی نداشته باشه، نحوه طراحی این نوع برنامه‌های جاسوس سخت افزار، صفحه کلید، یا ماوس آموزش می‌دم توی همین سایت)، حال برای امنیت برنامه‌های تحت وب یا ویندوز چگونه می‌توان در زمان ورود اطلاعات حساس مانند کلمه عبور یا شماره کارت اعتباری این امنیت را برای کاربر ایجاد کرد که داده‌هایش توسط این سخت افزارها یا بدافزارها جایی ثبت نشود؟

بله، حدس شما درست است استفاده از صفحه کلیدهای مجازی می‌شود یکی از بهترین راه‌های ممکن هست، چون در این روش‌ها کلید به صورت سخت افزاری فشرده نمی‌شود (کلید فشرده شده به صفت پیام‌های ویندوز نمی‌رود) در نتیجه نرم افزارها سخت افزارهای جاسوس نمی‌توانند این اطلاعات را ثبت کنند. و کاربر با خیال راحت می‌تواند داده‌های خود را وارد نمایند (تاکید می‌کنم این روش فقط جلو این نرم افزارها یا سخت افزارها را می‌گیرد و تضمینی برای اینکه در زمان ارسال داده‌های شما لو نزود ندارد).

خوب حال چه باید کرد؟

یک راه می‌تواند پیاده‌سازی صفحه کلید مجازی با کدهای طرف کلاینت مانند جاوا اسکریپت و وی‌بی اسکریپت است، اما [گروهی](#) پلاگینی را توسعه داده‌اند که با چند خط کدنویسی ساده به راحتی می‌توانید یک صفحه کلید مجازی چندزبانه (با هر زبانی که دلخون می‌خواهد) داشته باشید و از اون در برنامه‌های خودتون استفاده کنید.

نحوه‌ی نصب:

ایتدا فایل‌های مورد نیاز را از سایت [سازنده](#) که شامل [فایل جاوا اسکریپت](#)، [فایل استایل](#) و [یک تصویر](#) (آخرین نسخه) یا از این آدرس به صورت کامل (در حال حاضر نسخه 1.49) دریافت کرده، پس از دریافت فایل‌ها آنها را در هاست خود بارگزاری (آپلود) نمائید. سپس کدهای زیر را در صفحه‌ای که می‌خواهید صفحه کلید نمایش یابد در بین `<head>` ... `</head>` قرار دهید.

```
<script type="text/javascript" src="keyboard.js" charset="UTF-8"></script>
<link rel="stylesheet" type="text/css" href="keyboard.css">
```

حالا فقط کافی است به `input`ها و یا هر ورودی دیگر خود `class="keyboardInput"` بدهید.

مثال:

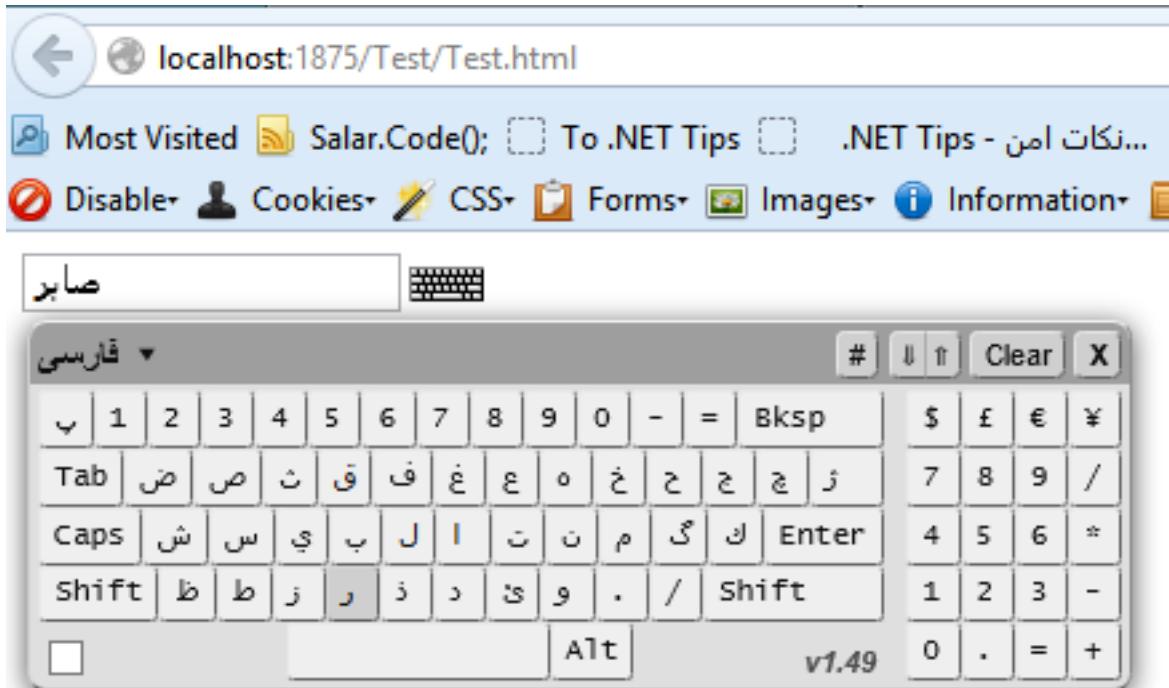
```
<input type="text" value="" class="keyboardInput">
```

در نهایت کد صفحه شما باید اینگونه باشد:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title></title>
  <script type="text/javascript" src="keyboard.js" charset="UTF-8"></script>
  <link rel="stylesheet" type="text/css" href="keyboard.css"/>
</head>
<body>
  <input type="text" value="" class="keyboardInput"/>
</body>
```

```
</html>
```

با این کار پس از اجرای صفحه مورد نظر خروجی شما مانند تصویر زیر خواهد بود، جهت محدود کردن کلیدها و عملیات دلخواه و سفارشی سازی با پارامترهای دلخواه می‌تواند از دموهای موجود در [سایت سازنده](#) بهره بگیرید.



موفق و موید باشید.

نظرات خوانندگان

نویسنده: مجتبی
تاریخ: ۱۳۹۱/۰۹/۰۶ ۱۸:۰

سلام

تا جایی که یادم می‌داد، برنامه‌هایی هم بودن که نقاطی که موس روشنون کلیک کرده بود رو ذخیره می‌کردن (چطوریش رو نمی‌دونم!).

اگه همچین برنامه‌هایی وجود داشته باشن حتی اگه مثل صفحات پرداخت آنلاین بانک‌ها که جای ۱۰ کاراکتر عدد، تغییر می‌کنه، با آنالیز شماره کارت‌ها و مسائلی از قبیل اینکه هر بانک با شماره خاصی شروع می‌شه و...، فکر کنم بدست آوردن اطلاعات حساس کار سختی نباش... درست می‌گم؟

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۱/۰۹/۰۷ ۱۱:۵۸

دوست من کار نشد نداره، امنیت نسبی هست

بله میشه این کار انجام داد به راحتی امکان پذیر هست، اما ذخیره نقاط چه کاربردی می‌تونی داشته باشه؟

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۰۹/۰۷ ۱۶:۴۶

مطلوب مفیدی بود.

ممnonem

نویسنده: nojirom
تاریخ: ۱۳۹۱/۱۰/۱۹ ۱۵:۵۸

با تشکر از آقای فتح اللهی عزیز بابت این مطلب فقط زبان پیش فرض کیبورد راچگونه فارسی تنظیم کنیم

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۱/۱۰/۲۰ ۷:۳۷

سلام

دوست من در فایل اسکریپت مربوطه اوایل فایل این خط پیدا کرده

```
this.VKI_kts = this.VKI_kt = "US International"; // Default keyboard layout
```

و آن را به هر زبانی که دلخواه شماست تغییر دهید مثلا برای فارسی اینگونه است:

```
this.VKI_kts = this.VKI_kt = "\u0641\u0627\u0631\u0633\u06cc"; // Default keyboard layout
```

البته من توی همون فایل "fa" جستجو کردم و عنوان اون در این قسمت قرار دادم
موفق و موید باشید

نویسنده: حیدری
تاریخ: ۱۳۹۱/۱۲/۱۸ ۲۰:۵۶

سلام

مرسى ، دستتون درد نکنه
ایشالا همیشه موفق باشد

نویسنده: مرتضی محمودی
تاریخ: ۲۰:۱۵ ۱۳۹۲/۰۸/۲۴

درود
منو راهنمایی کنین که چطور از این کلاس برای `RadTextBox` یا دیگر `Input`‌های تلریک استفاده کنم
باتشکر....

نویسنده: محسن خان
تاریخ: ۲۰:۵۷ ۱۳۹۲/۰۸/۲۴

چه فرقی می‌کنه؟ [به خاصیت](#) یا کلاس `CSS` اش مقدار `keyboardInput` رو نسبت بده. تمام این کنترل‌های مصنوعی، نهایتاً به `input`‌های استاندارد HTML تبدیل و رندر می‌شن. و گرنه مرورگر نمی‌تونه اون‌ها رو تشخیص بده یا پردازش کنه.

نویسنده: مرتضی محمودی
تاریخ: ۲۲:۲ ۱۳۹۲/۰۸/۲۴

آخه سمت سرور که میره دیگه کلاس میپره. من تکست هام داخل `AjaxPanel` هست.

نویسنده: محسن خان
تاریخ: ۲۳:۱۲ ۱۳۹۲/۰۸/۲۴

[دوباره باید بایندش کنی](#).

نویسنده: مهیار نظری پور
تاریخ: ۱۹:۱۳ ۱۳۹۳/۰۴/۲۵

بی نهایت درود؛ بنده، به یک کیبورد، به شکل کی پد (تنها اعداد، بصورت ماشین حسابی) نیاز مندم.
برای درج در `فیلد` "شماره تماس" `فرم رزروشین` سایتم.
در نت خیلی گشتم، اماً چیزی نیافتم.
دقیقاً چیزی مثل کی پد درگاه آنلاین بانک، که برای درج شماره حساب و این‌ها بکار میره روی سایت.

نویسنده: صابر فتح الهی
تاریخ: ۰:۳۹ ۱۳۹۳/۰۴/۲۷

سلام
خوب میتوانید از همان کدهای درگاه بانک الگو برداری کنید
تلاشتون بکنید اگر نتوانستید من کدش برآتون بگیرم
موفق و موید باشد

بعضی مواقع بهتر است یک دکمه در حال انجام پردازش‌های سمت سرور غیرفعال شود و وقتی عملیات سمت سرور به پایان رسید این دکمه دوباره فعال شود. غیرفعال کردن یک دکمه به این دلیل انجام می‌شود که از postback مجدد در حین postback شدن صفحه جلوگیری شود.

فرض کنید در رویداد کلیک یک دکمه کدی نوشته اید که اطلاعات یک دانشجو را ذخیره کند. کاربر نرم افزار بعد از یک بار کلیک روی دکمه، درخواستی به سرور می‌فرستد و این باعث می‌شود کدهای درون رویداد کلیک دکمه اجرا شوند، به دلیل این که کاملاً این کدها اجرا نشده اند، صفحه هم postback نشده است. کاربر به هر دلیل صبر نمی‌کند تا این پردازش تمام شود (شاید او نمی‌داند عملیاتی در سمت سرور در حال اجرا شدن هست چون پیغامی به او نشان داده نشده است)، در نتیجه باز هم روی همین دکمه کلیک می‌کند و باعث می‌شود در حین پردازش کدهای درون دکمه توسط سرور، دوباره درخواستی به سمت سرور فرستاده شود و باز هم کدهای درون همین دکمه به اجرا در آیند. به عبارت بهتر در حین postback شدن صفحه دوباره درخواست کردن صفحه را می‌دهد و سرور هم در حین انجام کدهای درون دکمه با درخواست قبلی، دوباره کدهای درون رویداد کلیک را اجرا می‌کند. در این مثال این کار باعث می‌شود کدهای رویداد کلیک دکمه، به تعداد کلیک‌های انجام شده اجرا شوند. در نتیجه به همین تعداد می‌تواند یک دانشجوی مستقل در دیتابیس ذخیره شود. این می‌تواند مشکلات بسیاری را همراه داشته باشد.

به همین دلیل بهتر است با کلیک روی این دکمه این کارها اتفاق بیفتد:

- الف) غیرفعال کردن دکمه در حین انجام پردازش‌های سمت سرور
- ب) نشان دادن یک پیغام به کاربر در حین انجام پردازش‌های سمت سرور
- ج) فعال کردن دکمه بعد از انجام پردازش‌های سمت سرور

برای غیرفعال کردن دکمه در حین انجام پردازش‌های سمت سرور نمی‌توان از کدهای سمت سرور استفاده کرد. چون تا کاملاً صفحه postback نشود نمی‌توان این کدها را به صفحه اعمال کرد. پس این گزینه کنار می‌رود.
راه حل بسیار خوب استفاده از جاوا اسکریپت است. مثال زیر را ببینید:

```
<asp:Button runat="server"
  ID="btnProcess"
  Text="پردازش"
  onclick="btnProcess_Click"
  OnClientClick="this.disabled = true; this.value = '';
    ...
    در حال پردازش اطلاعات";
  UseSubmitBehavior="false"
  />
<asp:Label runat="server" ID="lblMessage" Text=""></asp:Label>
```

در رویداد OnClientClick کارهای (الف) و (ب) انجام می‌شوند و با false کردن مقدار رویداد UseSubmitBehavior کار (ج) انجام می‌شود.

و در رویداد کلیک دکمه کد زیر را بنویسید:

```
protected void btnProcess_Click(object sender, EventArgs e)
{
  // insert student in database
  System.Threading.Thread.Sleep(2000);
  lblMessage.Text = "پردازش اطلاعات به پایان رسید";
}
```

در این رویداد باید یک دانشجو اضافه شود. برای ایجاد یک پردازش سمت سروری دو ثانیه ای از متده Sleep استفاده شده است.

کد برگرفته شده از : dotnetforum.1k

نظرات خوانندگان

نویسنده: مجتبی صحرائی
تاریخ: ۱۳۹۱/۰۹/۱۲ ۶:۹

به نظر میاد وقتی تو صفحه validator داشته باشیم و postback جلوی شدن صفحه رو تحت شرایطی میگیره؛ دکمه در حالت غیر فعال باقی خواهد موند

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۱/۰۹/۱۲ ۷:۳۱

در صورتی که بتونیم با بررسی های سمت سروری جلوی postback های همزمان رو بگیریم امنیت بالاتری خواهیم داشت.
(منظورم استفاده از custom validator هست که در سمت سرور و بسته به شرایطی که گفته شده validate می شود)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۱۲ ۸:۳۸

برای رفع این مشکل (غیرفعال کردن سازگار با اعتبار سنجی سمت کاربر) باید به نحو زیر عمل کرد:

```
UseSubmitBehavior="false"  
OnClientClick="if ((typeof(Page_ClientValidate) == 'function') && (Page_ClientValidate() == false)) {  
    return false;} this.disabled=true;"
```

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۱/۰۹/۲۴ ۲۰:۳۶

ممnon از کار شما من این روش تست کردم اما زمانی که با یک تاییده همراه باشے کد تقریبا از کار میافته، صفحه به سرور پست میشه اما عملیاتی صورت نمیگیره.
کدی که نوشتتم:

```
$(“input[type=submit]”).click(function () {  
    if ((typeof (Page_ClientValidate) == 'function') && (Page_ClientValidate() == false)) {  
        return false;  
    }  
    if (!confirm("آیا مطمئن هستید؟"))  
        return false;  
    this.disabled = true;  
    this.value = '... در حال پردازش اطلاعات';  
    return true;  
});
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۲۴ ۲۱:۱

در jQuery برای لغو ارسال به سرور، بهتر است بجای e.preventDefault() از return false استفاده شود:

```
....click(function(e) {  
    e.preventDefault() // stop the automatic form submission
```

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۱/۰۹/۲۵ ۸:۳۰

غیر فعال کردن یک دکمه در حین انجام پردازش های سمت سرور

سلام این روش استفاده کردم (با return و بدون return) اما جواب نمی ده و فرم ارسال میشه، عملیات هم انجام میشه چه کاربر تایید کنه یا نکنه

نویسنده: **وحید نصیری**
تاریخ: ۹:۴۱ ۱۳۹۱/۰۹/۲۵

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile("~/Site.master" AutoEventWireup="true"
CodeBehind="Default.aspx.cs" Inherits="WebAppSingleSubmit._Default" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
    <script type="text/javascript">
        $(document).ready(function () {
            $("input[type=submit]").click(function (e) {
                if ((typeof (Page_ClientValidate) == 'function') && (Page_ClientValidate() == false)) {
                    return false;
                }
                if (!confirm("آیا مطمئن هستید؟")) {
                    return false;
                }
                this.disabled = true;
                this.value = '... در حال پردازش اطلاعات';
                __doPostBack($(this).attr('name'), '');
            });
        });
    </script>
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
    <p>
        <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
ControlToValidate="TextBox1"
            ErrorMessage="RequiredFieldValidator"></asp:RequiredFieldValidator>
        <br />
        <asp:Button ID="Button1" runat="server" Text="Button" OnClick="Button1_Click" />
    </p>
</asp:Content>
```

نویسنده: **بهاره**
تاریخ: ۱۹:۴۴ ۱۳۹۲/۱۱/۱۷

سلام

آیا روشی هست که بشه جلوی کلیک کاربر روی دکمه refresh مرورگر (IE) را گرفت؟
امکانش هست که این دکمه را غیرفعال کرد؟

نویسنده: **وحید نصیری**
تاریخ: ۱۹:۵۴ ۱۳۹۲/۱۱/۱۷

« [الگوی PRG در ASP.NET MVC](#) »

نویسنده: **بهاره**
تاریخ: ۱۳:۵ ۱۳۹۲/۱۱/۱۹

ممnon. ولی من از MVC استفاده نمی کنم. پروژه در قالب SharePoint 2007 است. 303 در asp.net Redirect وجود دارد؟

نویسنده: **وحید نصیری**
تاریخ: ۱۳:۲۸ ۱۳۹۲/۱۱/۱۹

- مهم داشتن یک ایده اولیه برای شروع و جستجو هست.

- معادل return RedirectToAction در آن کدها Response.Redirect است در وب فرمها. مابقی اصول الگوی PRG تفاوتی نمی کند. البته Redirect در اینجا از status code=302 استفاده می کند؛ ولی اکثر مرورگرها آنرا همانند 303 پردازش خواهند کرد.

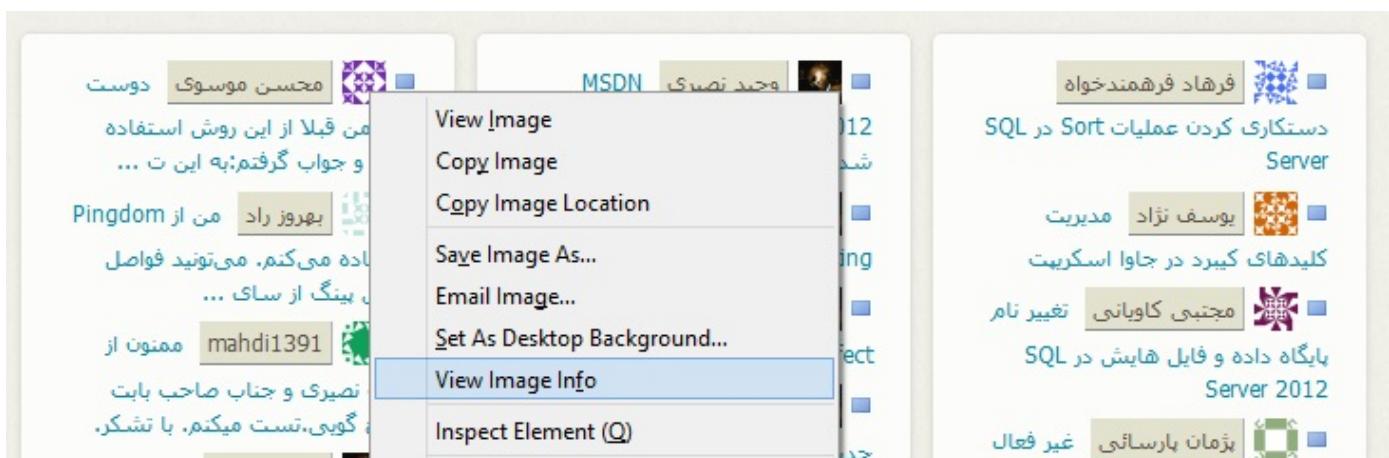
نوبسنده: امیر نوروزیان
تاریخ: ۱۳۹۳/۱۰/۱۹ ۱۳:۳۲

با سلام
در mvc چطوری این روش استفاده کنیم.

نوبسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۱۹ ۱۳:۳۶

نکته‌ی آن در این مطلب بحث شده: « استفاده از دکمه‌های CSS توئیتر در ASP.NET MVC - قسمت دوم »

برای برنامه نویسان همیشه این امکان هست که تصاویری را که از کاربر دریافت می‌کنند تغییر اندازه دهند، مثلا در همین سایت تصاویری از کاربران جهت نمایش در پروفایل آنها دریافت می‌شود، در همین سایت نیز این اتفاق می‌افتد مثلا تصاویر پروفایل کاربران با اندازه‌های متفاوتی نشان داده می‌شود.



برای انجام این کارها می‌توان به دو طریق عمل کرد:

تغییر اندازه تصویر در زمان ذخیره‌سازی
در زمانی که می‌خواهیم تصویر را به بازدید کننده نشان دهیم

در حالت 1 زمانی که تصویری را از کاربر دریافت می‌کنیم با توجه به اینکه تصویر را با چه اندازه‌هایی در نرم افزار نیاز داریم تغییر اندازه داده و تک تک ذخیره می‌کنیم، این روش کل عملیات در زمان ثبت و تنها یکبار اتفاق می‌افتد، این روش جای بیشتری از منابع (مانند هارد دیسک یا دیتابیس) سرور را اشغال می‌کند اما در عوض می‌توان گفت سرعت بالاتری دارد، در روش دوم زمانی که بازدید کننده از سایت (نرم افزار) بازدید می‌کند تصویر اصلی با توجه به نیاز تغییر اندازه داده شده و برای کاربر ارسال می‌شود (در واقع کاربر آن را مشاهده می‌کند)، این روش فضای کمتری از منابع را اشغال می‌کند اما در زمان اجرا عملیات اضافی برای هر کاربری (البته با کش کردن این عملیات کم می‌شود) انجام می‌شود.

در هر دو روش گفته شده در هر صورت ما باید متذکر (توابعی) برای تغییر اندازه تصویر داشته باشیم که در زیر نحوه نوشتمن آن را شرح خواهیم داد.

```

using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Drawing.Imaging;
using System.IO;

namespace PWS
{
  public static class Helpers
  {
    /// <summary>
    /// تغییر اندازه تصویر
    /// </summary>
    /// <param name="imageFile"></param>
    /// <param name="targetSize"></param>
    /// <param name="format"></param>
  }
}
  
```

```

/// <returns></returns>
public static byte[] ResizeImageFile(this byte[] imageFile, Int32 targetSize, ImageFormat
format)
{
    if (imageFile == null)
        throw new Exception("لطفا تصویر اصلی را مشخص نمایید");
    // باز کردن تصویر اصلی به عنوان یک جریان
    using (var oldImage = Image.FromStream(new MemoryStream(imageFile)))
    {
        محاسبه اندازه تصویر خروجی با توجه به اندازه داده شده
        var newSize = CalculateDimensions(oldImage.Size, targetSize);
        // ایجاد تصویر جدید
        using (var newImage = new Bitmap(newSize.Width, newSize.Height,
PixelFormat.Format24bppRgb))
        {
            using (var canvas = Graphics.FromImage(newImage))
            {
                canvas.SmoothingMode = SmoothingMode.AntiAlias;
                canvas.InterpolationMode = InterpolationMode.HighQualityBicubic;
                canvas.PixelOffsetMode = PixelOffsetMode.HighQuality;
                // تغییر اندازه تصویر اصلی و قرار دادن آن در تصویر جدید
                canvas.DrawImage(oldImage, new Rectangle(new Point(0, 0), newSize));
                var m = new MemoryStream();
                // ذخیره تصویر جدید با فرمات وارد شده
                newImage.Save(m, format);
                return m.GetBuffer();
            }
        }
    }
}

/// <summary>
/// محاسبه ابعاد تصویر خروجی
/// </summary>
/// <param name="oldSize">تصویر اصلی</param>
/// <param name="targetSize">اندازه تصویر خروجی</param>
/// <returns></returns>
private static Size CalculateDimensions(Size oldSize, Int32 targetSize)
{
    var newSize = new Size();
    if (oldSize.Height > oldSize.Width)
    {
        newSize.Width = Convert.ToInt32(oldSize.Width * (targetSize / (float)oldSize.Height));
        newSize.Height = targetSize;
    }
    else
    {
        newSize.Width = targetSize;
        newSize.Height = Convert.ToInt32(oldSize.Height * (targetSize / (float)oldSize.Width));
    }
    return newSize;
}
}

```

در متد `ResizeImageFile` تصویر اصلی به عنوان یک جریان باز می‌شود. (سطر 23)

اندازه تصویر خروجی با توجه به اندازه وارد شده توسط متد `CalculateDimensions` تعیین می‌شود؛ روال کار متد `CalculateDimensions` اینگونه است که اندازه عرض و ارتفاع تصویر اصلی بررسی می‌شود و با توجه به اینکه کدام یک از اینها بزرگتر است تغییر اندازه در آن صورت می‌گیرد، مثلاً در صورتی که عکس ارتفاع بیشتری نسبت به عرض تصویر داشته باشد تصویر تغییر اندازه داده شده نیز با توجه به تناسب ارتفاع تغییر داده می‌شود و بالعکس. (سطر 26)

پس از تغییر اندازه تصویر جدیدی در حافظه ایجاد می‌شود. (سطر 28)

سپس تنظیمات گرافیکی لازم برای تصویر جدید اعمال می‌شود. (سطر 30 تا 34)

تصویر اصلی با توجه به اندازه جدید تغییر کرده و در تصویر جدید قرار می‌گیرد. (سطر 36)

در نهایت تصویر جدید با فرمت وارد شده در متذخیره شده و به عنوان خروجی متذبرگشت داده میشود. (سطر 37 تا 40) خروجی این متذنیز آرایه بایتی میباشد که به سادگی میتوانید از آن برای ذخیره تصاویر در دیتابیس استفاده نمایید.

نحوه استفاده از این تابع در ASP.NET میتواند به صورت زیر باشد :

```
byte[] oldImage = FileUploadImage.FileBytes;  
byte[] target = oldImage.ResizeImageFile(100, ImageFormat.Jpeg);
```

در واقع فراخوانی مذکور تصویر ورودی را به اندازه 100 پیکسل تغییر داده و در ارایه بایتی به نام target ذخیره میکند.
در بخش بعد در زمان نمایش تصویر به کاربر آن را تغییر اندازه خواهیم داد.
لازم به ذکر است که کدهای تغییر اندازه StarterKit های میکروسافت کپی برداری شده است.

نظرات خوانندگان

نویسنده: محسن.د
تاریخ: ۱۳۹۱/۰۹/۱۴ ۱۳:۲۴

بسیار عالی

در فضای [asp.net MVC](#) کلاس [System.Web.Helpers](#) [webImage](#) این امکانات رو در اختیار توسعه دهنده قرار میده.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۱/۰۹/۱۴ ۱۵:۰

بله دیدم جالب بود واسم، اما تغییر اندازه ندیدم کدوم متده؟

نویسنده: محسن.د
تاریخ: ۱۳۹۱/۰۹/۱۴ ۱۸:۴

[WebImage.Resize](#)

نویسنده: پ.م
تاریخ: ۱۳۹۳/۰۲/۲۷ ۱۴:۳۶

لطفا در مورد اینکه در زمان واکنش تصاویر از دیتابیس (کش کنیم) توضیح دهید
چون خیلی سرعت کار با سایت پایین میاد
مثلا هر بار کاربر که بخواهد از صفحه ای به صفحه ای دیگه بره تصاویر از نو بارگزاری میشن و این سرعت سایت رو کند میکنه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۲۷ ۱۴:۳۹

به مطلب و مأخذ «[مدیریت درخواست‌های شرطی در ASP.NET MVC](#)» مراجعه کنید. نکات کلی در بحث عنوان شده. در مأخذ آن
پیاده سازی‌های مخصوص وب فرم‌ها هم هست.

نویسنده: رضایی
تاریخ: ۱۳۹۳/۰۵/۳۱ ۱۹:۱۷

سلام؛ موقع تغییر سایز یک تصویر png به png پس زمینه رو مشکی میکنه. آیا راهی هست در صورتی که تصویر اولیه png باشه
تصویر نهایی هم بدون پس زمینه باشه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۲۰ ۱۳:۵

[Resize PNG Image and Keep Transparent Background](#)

نویسنده: میثم شریفی
تاریخ: ۱۳۹۳/۱۲/۰۳ ۱۹:۲۵

با تشکر مقاله بسیار خوبی بود.
یک کتابخانه متن باز هست که امکانات کاملی ارائه کرده و [helper](#) خوبی هم برای [mvc](#) دارد [ImageResizer](#)

در کنار کتابخانه [elmah](#) که وظیفه ثبت تمامی خطاهای برنامه را دارد کتابخانه [MiniProfiler](#) امکان یافتن مشکلات کارایی و تنگناهای وب سایت را در اختیارمان قرار می‌دهد. دو قابلیت عمده که این ابزار فراهم می‌نماید امکان مشاهده و بررسی کوئری‌های خام ADO.NET از قبیل LINQ-to-SQL و SQL Server, Oracle و ... و ... نمایش زمان اجرای عملی صفحات

برای استفاده از این ابزار کافیست تا آن را از nuget دریافت نمایید

PM> Install-Package MiniProfiler

در صفحه _Layout در ASP.NET MVC در body تابع RenderIncludes را مانند زیر صدا بزنید تا در همه صفحات نمایش داده شود

```
@using StackExchange.Profiling;
<head>
  ...
</head>
<body>
  ...
  @MiniProfiler.RenderIncludes()
</body>
```

در کلاس global کد زیر را برای اجرای MiniProfiler اضافه نمایید

```
protected void Application_BeginRequest()
{
  if (Request.IsLocal)
  {
    MiniProfiler.Start();
  }
}

protected void Application_EndRequest()
{
  MiniProfiler.Stop();
}
```

برای پیکربندی MiniProfiler در web.config کد زیر را اضافه نمایید

```
<system.webServer>
  ...
  <handlers>
    <add name="MiniProfiler" path="mini-profiler-resources/*" verb="*"
      type="System.Web.Routing.UrlRoutingModule"
      resourceType="Unspecified"
      preCondition="integratedMode" />
  </handlers>
</system.webServer>
```

یا کتابخانه MiniProfiler.MVC را از nuget دریافت نمایید

PM> Install-Package MiniProfiler.MVC

با اضافه شدن این کتابخانه همه پیکربندی بصورت صورت خودکار انجام می‌گیرد. حال وب سایت را اجرا کنید در بالای صفحه

مانند شکل زیر مدت زمان بارگذاری صفحه نمایش داده می‌شود که با کلیک بر روی آن اطلاعات بیشتری را مشاهده می‌نمایید

localhost:25804

7.8 ms

Home/Index X220V2 on Tue, 11 Sep 2012 03:43:42 GMT

	duration (ms)	from start (ms)
http://localhost:25804/	2.3	+0.0
Render : Index	2.1	+3.0

[show time with children](#)

	client event	duration (ms)	from start (ms)
Response		13.0	+11.0
Dom Content Loaded Event		4.0	+198.0
Dom Complete			+202.0

[share](#) [show trivial](#)

اگر در اکشن اجرا شده کوئری اجرا شد باشد ستونی به نام `query times` نمایش داده می‌شود که تعداد کوئری‌ها و مدت زمان آن را نمایش می‌دهد

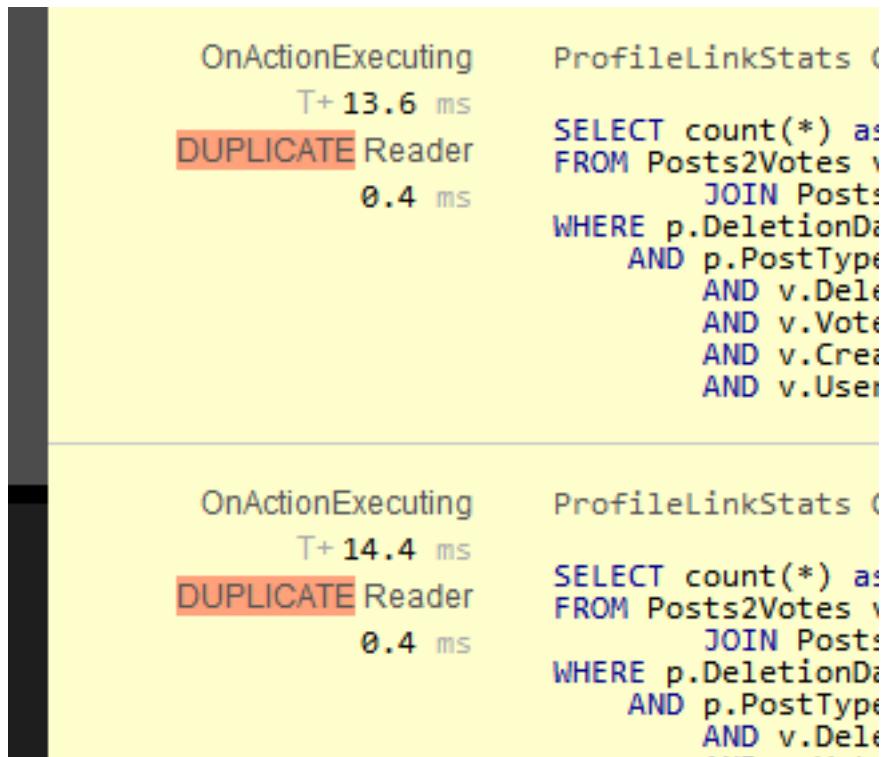
chat | meta | faq | search

Users/ProfileLinkStats NY-WEB03 at 2011-06-09 00:01:27Z **103.4 ms**

	duration (ms)	from start (ms)	query time (ms)	
OnActionExecuting	110.5	+ 1.3	19 sql 35.9	
TrySet	2.9	+ 5.1	1 sql 0.6	
Refresh	14.9	+ 8.1	5 sql 3.3	
Mutex_Clear	16.5	+ 57.7	1 sql 15.2	

[share](#) [show trivial](#) [show time with children](#) 37.5 % in sql

حال بر روی گزینه `sql` کلیک کنید که صفحه دیگری باز شود و کوئری خام آن را مشاهد نمایید اگر کوئری تکرار شده باشد در کنار آن با `DUPLICATE` متمایز شده است



برای مشاهده کوئری‌های MiniProfiler.EF کتابخانه Entity Framework/First Code را اضافه نمایید

PM> Install-Package MiniProfiler.EF

اگر بصورت دستی MiniProfiler را پیکربندی کرده باشد می‌بایست در Application_Start دستور زیر را اجرا نماید

```
protected void Application_BeginRequest()
{
    if (Request.IsLocal)
    {
        MiniProfiler.Start();
        MiniProfilerEF.Initialize();
    }
}
```

در حالت پیشرفته‌تر اگر قصد داشته باشد زمان یک قطعه کد را جداگانه محاسبه نماید بصورت زیر عمل نماید

```
public ActionResult Index()
{
    var profiler = MiniProfiler.Current;
    using (profiler.Step("Step 1"))
    {
        //code 1
    }
    using (profiler.Step("Step 2"))
    {
        //code 2
    }
    return View();
}
```

با این کار زمان هر step را بصورت جداگانه محاسبه می‌نماید. در ASP.NET Webforms دقیقاً به همین صورت استفاده می‌شود فقط

کافیست در masterpage اصلی یا اگر از masterpage استفاده نمی‌کنیم در صفحه مورد نظر تابع RenderIncludes را بصورت زیر صدا بزنیم

```
<%= StackExchange.Profiling.MiniProfiler.RenderIncludes() %>
```

امیدوارم مفید واقع شده باشد.

نظرات خوانندگان

نویسنده: سعید
تاریخ: ۱۳۹۱/۱۱/۱۵ ۱۸:۳

ممنون از شما. آخرین باری که miniprofiler.ef را تست کردم با ef5 کار نمیکرد و خطای داد. شما تست کردید با آخرین نگارش `ef code first`

نویسنده: مجتبی کاویانی
تاریخ: ۱۳۹۱/۱۱/۱۵ ۱۸:۱۴

نه هیچ مشکلی ندارد من خودم با ۵ ef code first تست کردم

نویسنده: Alireza Godazchian
تاریخ: ۱۳۹۱/۱۱/۱۸ ۸:۴۶

با سلام؛ آقای مهندس خیلی عالی توضیح دادید و بنده هم تست کردم و جواب گرفتم.
با تشکر فراوان

نویسنده: امیر
تاریخ: ۱۳۹۱/۱۱/۱۸ ۲۰:۵۲

سلام چرا تو دانست ای اس پی کار نمیکنه فقط تو ام وی سی جواب میده
نمیشناسه MiniProfiler

نویسنده: مجتبی کاویانی
تاریخ: ۱۳۹۱/۱۱/۱۸ ۲۲:۸

در ASP.NET هم بخوبی جواب میدهد آیا Handler را در Web.config اضافه نموده اید؟

نویسنده: هیمن صادقی
تاریخ: ۱۳۹۳/۰۴/۲۷ ۱۶:۵۳

با سلام و درود بی کران
من کار که فرمودید انجام دادم اما زمانی که اجرا میکنم خطای "No Entity Framework provider found for the ADO.NET provider with invariant name 'System.Data.Odbc'. Make sure the provider is registered in the 'entityFramework' section of the application config file. See <http://go.microsoft.com/fwlink/?LinkId=260882> for more information" نمایم

از entity framework6 , mvc4 استفاده میکنیم
در صورت امکان روش رفع خطای میدانید بیان نماید تا بتوانم از این ابزار استفاده کنم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۲۷ ۱۸:۲۲

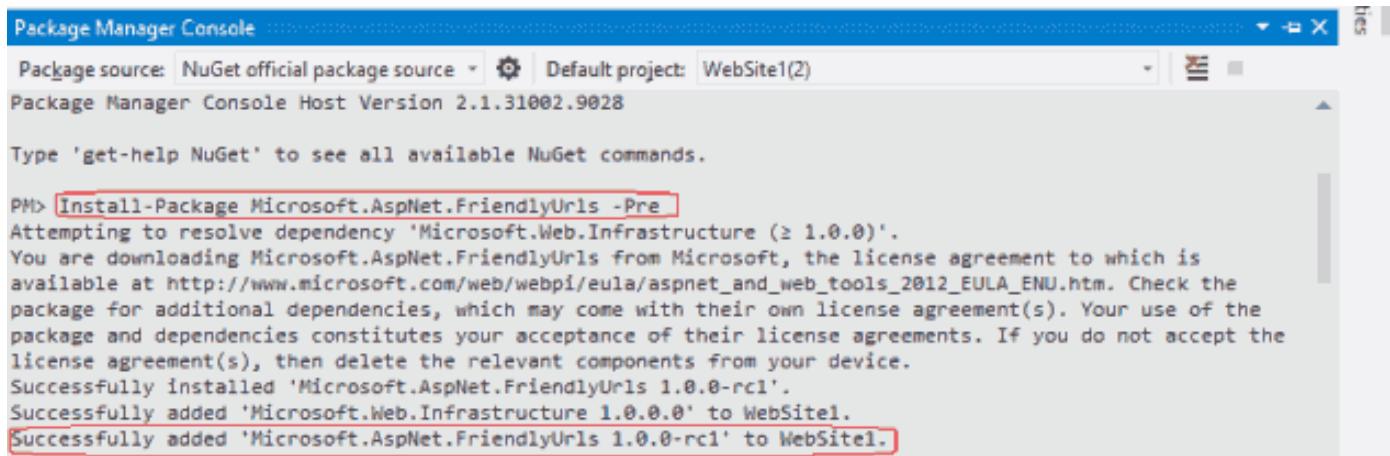
برای EF 6 یک بسته‌ی نیوگت جداگانه دارد. [اطلاعات بیشتر](#)

احتمالاً به اهمیت بحث [SEO](#) و آدرس‌های تمیز آن آشنا هستید. ASP.NET MVC از همان ابتدا از قابلیت URL Routing پشتیبانی می‌کرد و ASP.NET Web Forms هم از نسخه .NET 4.0 آن را در خود اضافه نمود. با این حال نوشتن Url Routing برای وب سایت‌ها کمی مشکل و در ASP.NET کمتر استفاده می‌شود. علاوه بر آن تغییر آدرس‌های قدیمی با Url Routing بسیار وقت گیر می‌باشد. به همین خاطر تیم توسعه ASP.NET کتابخانه ای با نام [Asp.NET FriendlyUrls](#) که فعلاً نسخه قبل از نهایی آن منتشر شده است را در اختیار توسعه دهندگان وب سایت‌ها قرار داده‌اند که با حداقل کد نویسی و بدون اضافه نمودن مسیریابی آدرس‌ها، تمام آدرس‌های قدیم و جدید را مطابق استاندارد SEO تغییر می‌دهد.

دریافت و نصب Asp.NET FriendlyUrls

در Visual Studio 2012.2 در صورتی که یک پروژه جدید از نوع ASP.NET Web Forms ایجاد نمایید بصورت پیشفرض این کتابخانه نصب شده است. اما می‌توانید با دستور زیر نیز آن را از Nuget دریافت نمایید:

```
Install-Package Microsoft.AspNet.FriendlyUrls -Pre
```



```

Package Manager Console
Package source: NuGet official package source  Default project: WebSite1(2)
Package Manager Console Host Version 2.1.31002.9028

Type 'get-help NuGet' to see all available NuGet commands.

PM> Install-Package Microsoft.AspNet.FriendlyUrls -Pre
Attempting to resolve dependency 'Microsoft.Web.Infrastructure (>= 1.0.0)'.
You are downloading Microsoft.AspNet.FriendlyUrls from Microsoft, the license agreement to which is
available at http://www.microsoft.com/web/webpi/eula/aspNet_and_web_tools_2012_EULA_ENU.htm. Check the
package for additional dependencies, which may come with their own license agreement(s). Your use of the
package and dependencies constitutes your acceptance of their license agreements. If you do not accept the
license agreement(s), then delete the relevant components from your device.
Successfully installed 'Microsoft.AspNet.FriendlyUrls 1.0.0-rc1'.
Successfully added 'Microsoft.Web.Infrastructure 1.0.0.0' to WebSite1.
Successfully added 'Microsoft.AspNet.FriendlyUrls 1.0.0-rc1' to WebSite1.

```

پس از نصب، در کلاس Global می‌بایست کتابخانه را فعال نمایید:

```

public static class RouteConfig
{
    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.EnableFriendlyUrls();
    }
}

```

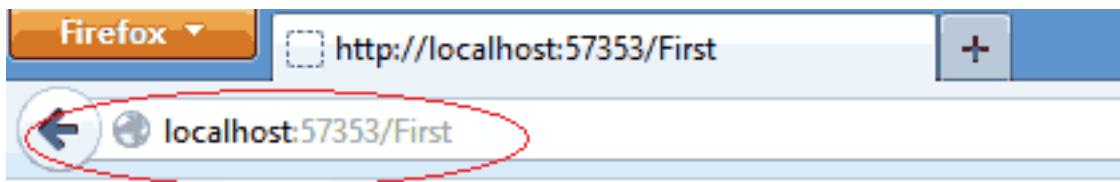
و همچنین در Application_Start

```

void Application_Start(object sender, EventArgs e)
{
    RouteConfig.RegisterRoutes(RouteTable.Routes);
}

```

به همین راحتی؛ حال دو صفحه First.aspx و Second.aspx را به پروژه اضافه نمائید و وب سایت را اجرا کنید. بدون نوشتند پسوند aspx به صفحات دسترسی خواهد داشت:



First Page

از درون برنامه هم به راحتی می‌توانید صفحات را به همان صورت قبل بدون پسوند صدا بزنید:

```
Response.Redirect("~/Second");
```

دسترسی به پارامترها نیز به صورت قبل امکان پذیر می‌باشد:

```
Response.Redirect("~/First?Id=1001");
string Id = Request.QueryString["Id"] as string;
```

برای آدرس‌های پیشرفته‌تر، 3 متود الحاقی اضافه گردیده است. مثلاً اگر آدرس زیر را داشته باشیم

```
http://localhost:57353/First/Products/NewProduct
```

Request.GetFriendlyUrlFileVirtualPath()

مسیر مجازی فایل را بر می‌گرداند

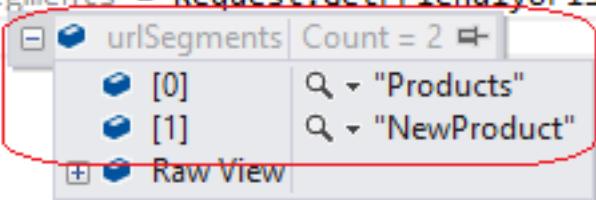
```
string virtualPath = Request.GetFriendlyUrlFileVirtualPath();
```

virtualPath | Q ~ /First.aspx |

Request.GetFriendlyUrlSegments()

بخش‌های آدرس را بر می‌گرداند

```
protected void Page_Load(object sender, EventArgs e)
{
    IList<string> urlSegments = Request.GetFriendlyUrlSegments();
```



Request.GetFriendlyUrlFileExtension()

پسوند فایل آدرس را بر می‌گرداند

```
string extension = Request.GetFriendlyUrlFileExtension();
```



کلاس FriendlyUrl دو متود استاتیک و یک خاصیت دارد

FriendlyUrl.Segments

شبیه() مجموعه بخش‌های آدرس صفحه جاری را بر می‌گرداند

FriendlyUrl.Resolve

آدرس صفحه جاری را بر می‌گرداند

FriendlyUrl.Href

امکان ساخت یک آدرس را با استفاده از مسیر مجازی، بخش‌ها و پارامترها، فراهم می‌نماید

```
string url = FriendlyUrl.Href("~/First", "Product", 1001);
```



در آخر، فضای نام Microsoft.AspNet.FriendlyUrls می‌باشد.

نظرات خوانندگان

نویسنده: M.Q

تاریخ: ۱۳۹۱/۱۱/۳۰ ۱۹:۵۰

خیلی عالی بود ممنون

نویسنده: صابر فتح الهی

تاریخ: ۱۳۹۱/۱۲/۰۲ ۸:۴۸

دستت درد نکنه مهندس عالی بود، فکر کنم نسخه نهاییش اومده در این [لینک](#)

نویسنده: Rahmani

تاریخ: ۱۳۹۱/۱۲/۰۵ ۱۴:۱۶

وقتی با vs سایت رو run می‌کنم این قابلیت کار می‌کنه، اما وقتی می‌فرستم روی iis رو سرور واقعی، خطای 404 میده.
سرچیدم ولی جواب خاصی پیدا نکردم. کمک لطفاً

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۱۲/۰۵ ۱۴:۲۹

از چه IIS ای استفاده می‌کنید؟ اگر 6 IIS است، [این مطلب](#) را مطالعه کنید.
به صورت خلاصه 6 IIS برنامه‌های ASP.NET و صفحات آنرا بر اساس پسوند آنها پردازش می‌کند. اگر پسوندی ندارند باید IIS را طوری تنظیم کرد که تمام درخواست‌های رسیده را به موتور ASP.NET هدایت کند.

نویسنده: Rahmani

تاریخ: ۱۳۹۱/۱۲/۰۵ ۱۵:۲۲

ببخشید. باید همونجا می‌گفتم.
Windows Server 2008 روی IIS 7.5

نویسنده: وحید نصیری

تاریخ: ۱۳۹۱/۱۲/۰۵ ۱۷:۵

- بررسی کنید آیا برنامه شما در IIS7 در حالت classic اجرا می‌شود یا integrated mode . اگر حالت کلاسیک است، همان روش IIS6 که عنوان شد باید طی شود.
- تنظیم زیر را به web.config برنامه اضافه کنید:

```
<system.webServer>
  <modules runAllManagedModulesForAllRequests="true" />
</system.webServer>
```

- این هات فیکس را نصب کنید: ([^](#)). البته اگر سرویس پک یک 2008 R2 نصب شده، نیازی به آن نیست.

نویسنده: Rahmani

تاریخ: ۱۳۹۱/۱۲/۰۶ ۱۳:۰

با نصب اون هات فیکس مشکل رفع شد.
ممنون آقای نصیری

نویسنده: پریسا
تاریخ: ۱۳۹۱/۱۲/۰۹ ۱۸:۵

چرا وقتی از FriendlyUrl استفاده می‌کنم به ازای هر style یا فایل.js صفحه اجرا می‌شود؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۰۹ ۱۹:۱۸

می‌توانید مسیرهایی/فایل‌هایی را که نباید وارد سیستم routing شوند از لیست حذف کنید. در همان متدهای RegisterRoutes تعريف زیر اضافه شوند (اگر متدهای IgnoreRoute در دسترس نبودند همین کار را انجام می‌دهد):

```
routes.IgnoreRoute("*.html|js|css|gif|jpg|jpeg|png|swf");
```

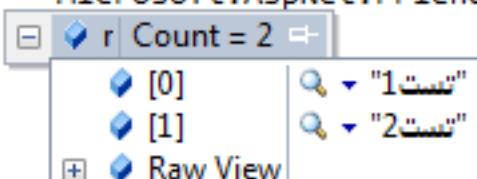
نویسنده: پرسا
تاریخ: ۱۳۹۱/۱۲/۱۰ ۹:۳۴

سلام کدهایی که فرمودین اضافه کردم

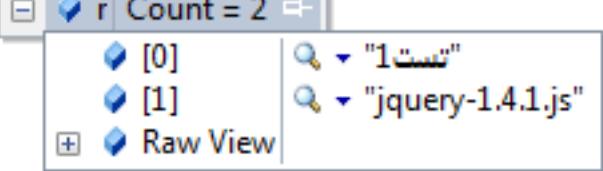
```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.EnableFriendlyUrls();
    routes.Ignore("*.html|js|css|gif|jpg|jpeg|png|swf");
}
```

اما باز نشد با اجرای صفحه همچنان به ازای هر فایل.js صفحه اجرا می‌شود page load

```
public partial class Test : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            IList<string> r = Microsoft.AspNet.FriendlyUrls.FriendlyUrl.Segment
        }
    }
}
```



```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        IList<string> r = Microsoft.AspNet.FriendlyUrls.FriendlyUrl.Segments;
    }
}
```



نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۱۰ ۱۰:۲۵

- برای تعریف صحیح اسکریپت‌ها یا از متد ResolveUrl استفاده کنید:

```
<script src='<%= ResolveUrl("~/Scripts/jquery-1.4.1.js")%>' type="text/javascript" />
```

و یا روش بهتر استفاده از [اسکریپت منیجر](#) است.

نویسنده: پرسا
تاریخ: ۱۳۹۱/۱۲/۱۰ ۱۱:۵۳

خیلی لطف کردن فقط یه سوال دیگه
فایل‌های `css` اضافه کردم به اسکریپت منیجر درست شد
اما فایل `css` را چی کار کنم

نویسنده: محسن
تاریخ: ۱۳۹۱/۱۲/۱۲ ۰:۱۹

تست کنید آیا این سیستم به `css` ای که داخل ریشه روت سایت هم هست ایراد می‌گیره؟ اگه الان با `css` مشکل ندارید علت این است که سیستم فشرده سازی و یکی کردن اون‌ها رو به صورت یک زیر پوشه ارائه نمی‌ده.

نویسنده: پرسا
تاریخ: ۱۳۹۱/۱۲/۱۲ ۸:۴۸

بله من اگر آدرس `Css` در پروژه به صورت زیر باشد

```
<link href="css/StyleSheet.css" rel="stylesheet" type="text/css" />
```

باز برای این `pageload` صفحه را اجرا می‌کند

نویسنده: پریسا
تاریخ: ۱۳۹۲/۰۲/۲۱ ۱۷:۱۲

سلام

زمانی که از این `friendly url` استفاده می‌کنم `a.test.ir/gallery` که `subdomain` دارم مثلا `a.test.ir/gallery` صفحه‌ی گالری

را پیدا نمی‌کنند در صورتی که بدون subdomain کار می‌کند

نویسنده: ایمان صفری
تاریخ: ۱۳۹۲/۰۵/۰۹ ۲۰:۲۰

با تشکر از شما من وقتی از Resolve استفاده می‌کنم مشکلی ندارم اما زمانیکه وارد یک Route تعریف شده میشم هم CSS و فایلهای jquery که استفاده کردم به هم می‌ریزه و هم اینکه Url ای که وارد شدم با کلیک روی یک لینک دیگه به آدرس مورد جستجو اضافه میشه از کدهایی که هم آقای نصیری گفتند استفاده کردم ولی نتیجه ای نداد

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۵/۱۰ ۰:۱۸

برای خطایابی و یافتن مشکل مطلب [نحوه استفاده از افزونه Firebug برای دیاگ برنامه‌های ASP.NET](#) مبتنی بر [jQuery](#) می‌توانه مفید باشد.

نویسنده: حسین
تاریخ: ۱۳۹۲/۱۱/۰۸ ۱۶:۲۷

با سلام و تشکر از شما: این افزونه با medium trust کار نمیکنه راهی وجود داره که با medium Trust کار کنه؟ یه درد و دل! اینکه چرا بعضی هاستها تنظیمات medium trust رو اعمال میکنن برام عجیبه! میگن به دلیل امنیتی! خب سایت هایی که هستن چجوری مشکل امنیت پیدا نمیکنن! مثل همین موضوع من از asp.net identity هم نمیتونم استفاده کنم چون به owin گیر میده (به همون دلیل امنیتی!)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۸ ۱۶:۵۸

یکی از اعضای تیم ASP.NET [در اینجا عنوان کرد](#) که Medium Trust به صورت رسمی منسوز شده درنظر گرفته می‌شود و در محصولات آتی ما به هیچ عنوان مدنظر نخواهد بود و هاستها باید به روش [OS-level isolation](#) کوچ کنند.

نویسنده: سارا
تاریخ: ۱۳۹۲/۱۱/۲۰ ۱۰:۴۸

ببخشید من دقیقاً منظورتون رو از کلاس global متوجه نمی‌شم... باید چنین کلاسی ایجاد کنیم یا باید با نصب به وجود بیاد؟... لطفاً یه توضیح مختصر در این مورد بدین

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

منظورشان همان فایل global.asax.cs معروف است (Add New Item -> Global Application Class).

نویسنده: مرتین
تاریخ: ۱۳۹۲/۱۱/۲۰ ۱۱:۵۸

سلام: من کتابخانه رو به پروژه اضافه کردم. اما برای فعال سازیش به مشکل برمی‌خوردم. EnableFriendlyUrls رو نمیشناسه و خطای میده

```
System.Web.Routing.RouteCollection' does not contain a definition for 'EnableFriendlyUrls'
and no extension method 'EnableFriendlyUrls' accepting a first argument of type
'System.Web.Routing.RouteCollection' could be found (are you missing a using directive or an assembly
reference?)
```

استفاده هم زمان این روش با url routing که مشکلی به وجود نمی‌آید؟

نویسنده: وحید نصیری
تاریخ: ۱۲:۲۴ ۱۳۹۲/۱۱/۲۰

متد الحقی است و نیاز به تعریف فضای نام مرتبط با آن در کلاس مورد استفاده می‌باشد (using Microsoft.AspNet.FriendlyUrls).

نویسنده: رضا
تاریخ: ۱۰:۴۵ ۱۳۹۲/۱۲/۰۳

من این رو اجرا می‌کنم تا صفحه ای به اسم viewSwitcher ارور میده هیچ کدام از کلمات این خط رو نمیشناسه

```
<%: CurrentView %> view | <a href="<%: SwitchUrl %>" data-ajax="false">Switch to <%: AlternateView %></a>
```

مشکلش چیه؟

نویسنده: وحید نصیری
تاریخ: ۱۱:۴۸ ۱۳۹۲/۱۲/۰۳

مشکلی نداره. یک نمونه برای آزمایش: [FriendlyUrlsTests.rar](#)

مايكروسافت با افزایش سرعت به روز رسانی توسعه پروژه های سورس باز خود جهت پاسخ دادن به نیاز توسعه دهندگان و توسعه و پژوهش استادیو مطابق با آخرین تکنولوژی های تولید وب سایت، می کوشد تعداد بیشتری از توسعه دهندگان را به سمت استفاده از تکنولوژی های خود سوق دهد.

سالها است که برنامه نویسان خبره با توجه به روش کاری خود از امکانات Code Generator ها برای تولید کدهای لایه های Data Access ، Logic و یا حتی User Interface استفاده می نمایند. پس از عرضه Entity Framework و تولید خودکار کدهای لایه های ASP.NET MVC در ASP.NET Web Forms، این بار این امکان علاوه بر خسته کننده و تکراری، کدهای لایه رابط کاربر (CRUD) را نیز تولید نماییم.

شروع کار با ASP.NET Scaffolding

پیش نیاز این کار استفاده از 2012.2 Visual Studio به همراه Web Tools می باشد.

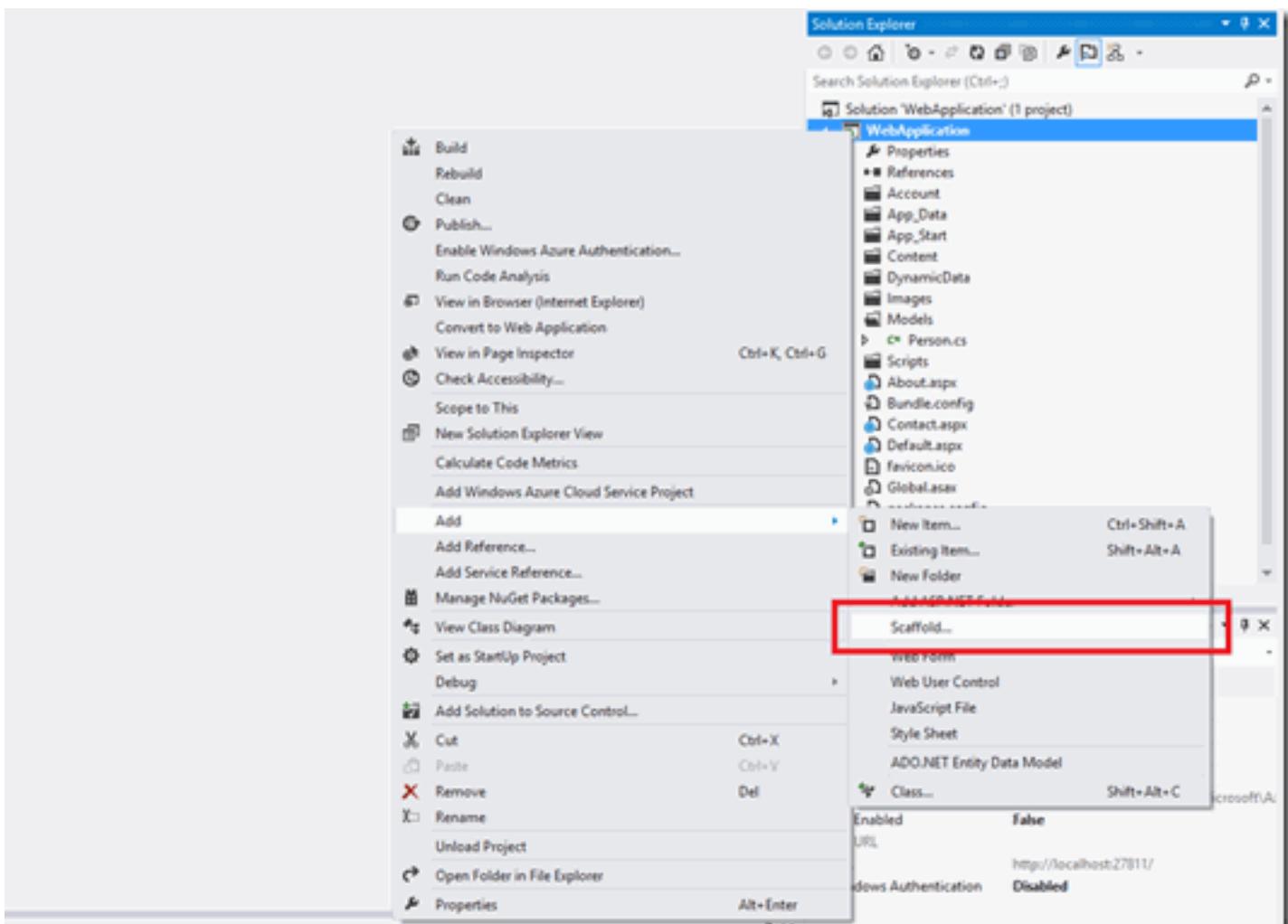
اول، ابزار Microsoft ASP.NET Scaffolding را از منوی Tools گزینه Extensions and Updates دریافت و نصب نمایید.

دوم پروژه جدیدی از نوع Visual C# ASP.NET Web Forms Application با فریم ورک 4.5 ایجاد نمایید.
از پنجره ASP.NET Web Forms Scaffold Generator install کتابخانه NuGet Package manager را دریافت نمایید
`install-package Microsoft.AspNet.Scaffolding.WebForms -pre`

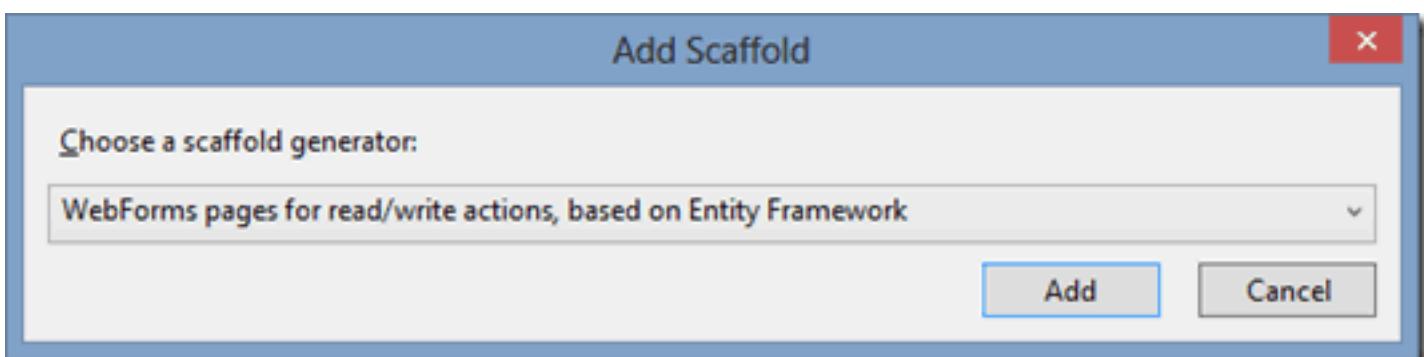
کلاس Person را مانند زیر در فolder Models ایجاد نمایید

```
public class Person
{
    [ScaffoldColumn(false)]
    public int ID { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
}
```

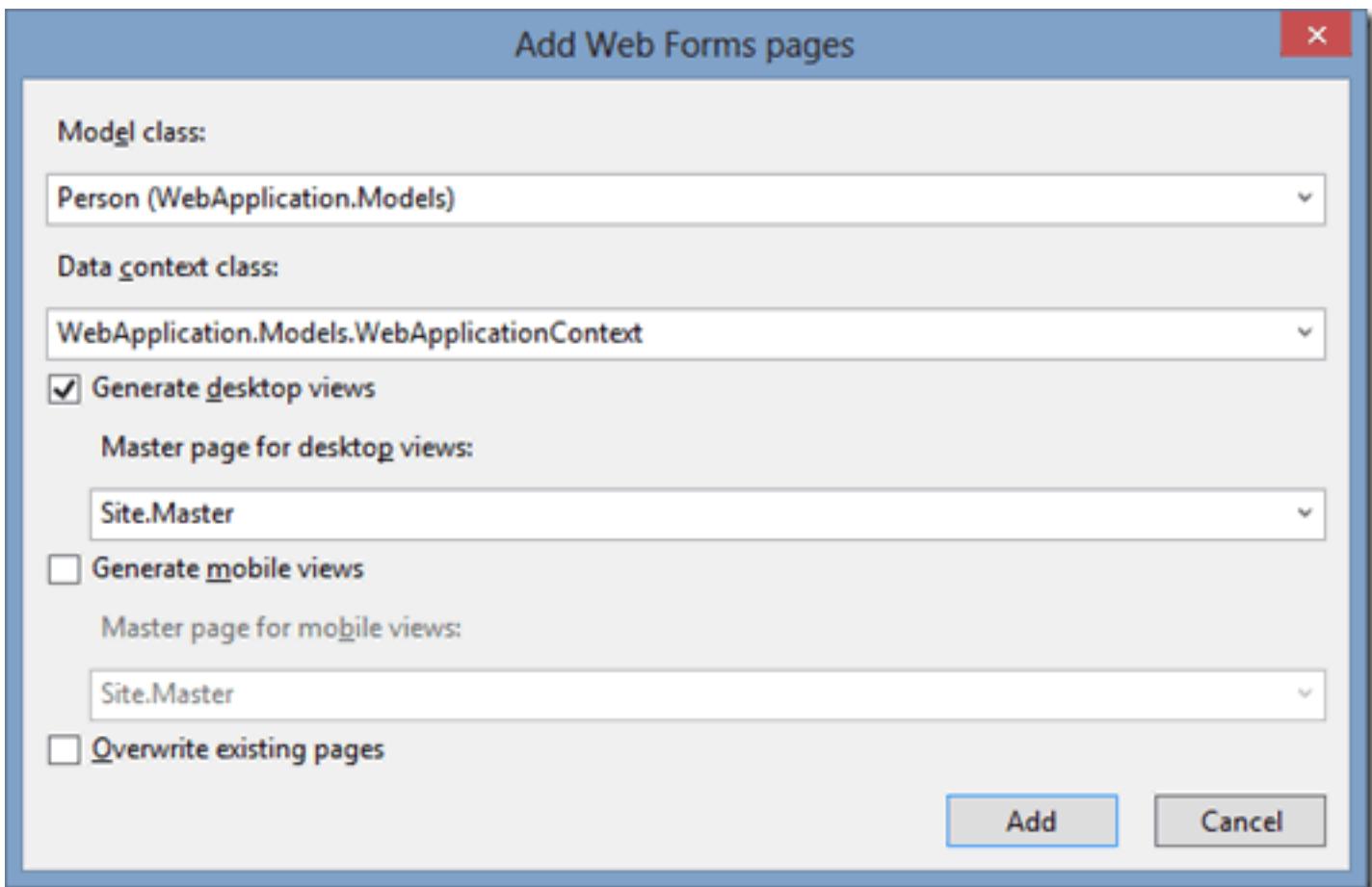
ویژگی ScaffoldColumn را برای ID، برابر false قرار دهید تا از ایجاد این ستون جلوگیری نمایید.
پروژه را Build نمایید.
بر روی پروژه راست کلیک و از گزینه Add، گزینه ... Scaffold را انتخاب نمایید.



از پنجره Add Scaffold باز شده بر روی گزینه Add کلیک کنید.

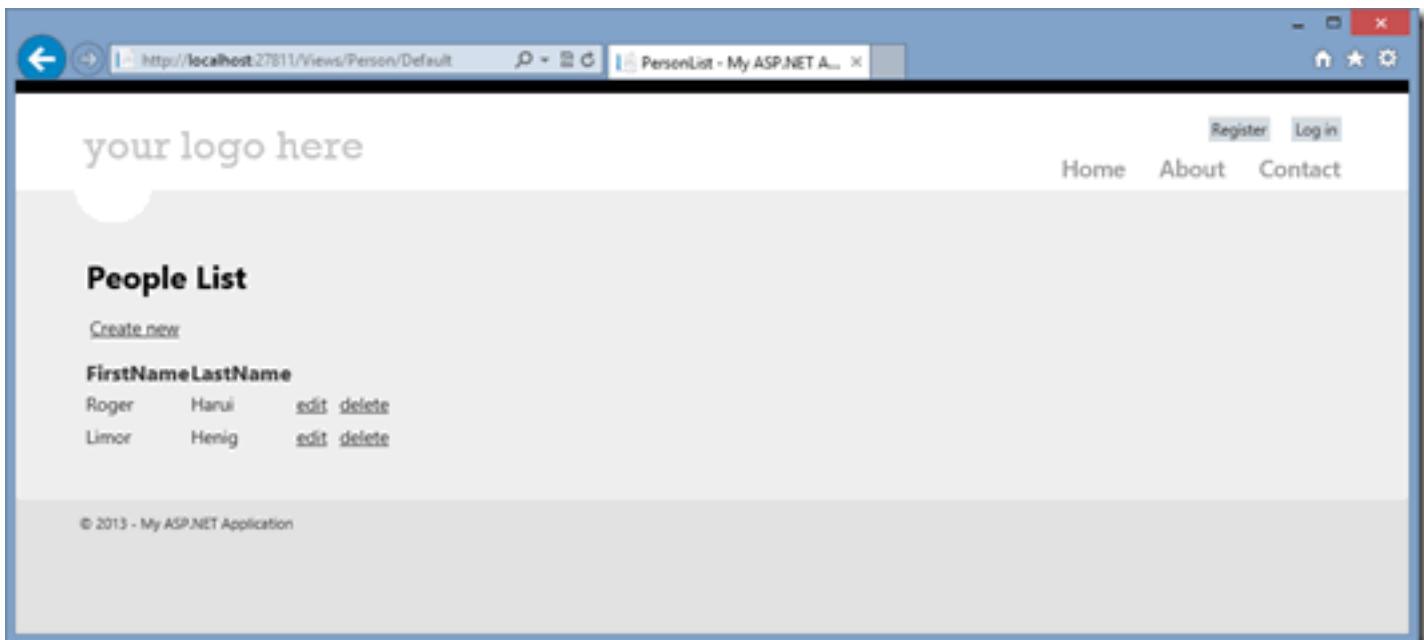


پنجره Add Web Forms Pages مانند زیر باز می‌شود که امکان انتخاب کلاس، MasterPage و Data Context فراهم می‌باشد.



از گزینه Data Context class گزینه New Data Context را انتخاب نمایید. صفحات مورد نیاز را در فolder Views/Person ایجاد می‌نمایید.

کدهای تولید شده را می‌توانید بازبینی نمایید پروژه را اجرا تا خروجی کار را مشاهده نمایید.



نظرات خوانندگان

نویسنده: محمد باقر سیف اللهی
تاریخ: ۹:۳۸ ۱۳۹۱/۱۲/۰۴

سلام . ممنون بابت معرفی این Extensions . چه تفاوتی بین این Dynamic Data و پروژه‌های Extensions که در ASP هست ، وجود داره ؟ و یک سوال دیگه ، هدف خاصی برای استفاده از دات نت ۴.۵ وجود داره ؟

نویسنده: علی قمشلوی
تاریخ: ۹:۴۵ ۱۳۹۱/۱۲/۰۴

خیلی عالی بود

نویسنده: مجتبی کاویانی
تاریخ: ۱۰:۲۶ ۱۳۹۱/۱۲/۰۴

این ابزار با استفاده از Dynamic Data عملیات تولید کدها را انجام می‌دهد و پیش نیاز آن دات نت ۴.۵ حتماً بخارط پشتیانی بهتر از Entity Framework First Code دارد.

نویسنده: محمد باقر سیف اللهی
تاریخ: ۱۰:۴۷ ۱۳۹۱/۱۲/۰۴

تشکر

نویسنده: فاتحی
تاریخ: ۱۱:۳۹ ۱۳۹۱/۱۲/۰۷

سلام

من ۲۰۱۲.۲ Web Tools را نصب کردم و تمام مراحلی را که توضیح دادید رو هم انجام دادم.
اما نمی‌تونم گزینه scaffold رو تو Add پیدا کنم.
به نظرتون مشکل از کجاست؟

نویسنده: وحید نصیری
تاریخ: ۲۱:۱۵ ۱۳۹۱/۱۲/۰۷

[لیست جامع به روز رسانی‌های ۲۰۱۲](#) VS را بررسی کنید. شاید چیزی از قلم افتاده این بین.

نویسنده: مجتبی کاویانی
تاریخ: ۲۱:۵۲ ۱۳۹۱/۱۲/۰۷

برای نمایش گزینه Scaffold می‌بایست ابزار [Microsoft ASP.NET Scaffolding](#) را نصب نمایید

نویسنده: امیر
تاریخ: ۱:۳۱ ۱۳۹۱/۱۲/۰۸

سلام بعد از انجام تمام مراحل این خطاب را ظاهر می‌شه

.Server Error in '/' Application

The resource cannot be found. Version Information: Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.0.30319.17929

نویسنده: محسن
تاریخ: ۱۳۹۱/۱۲/۰۸ ۱۲:۱۷

برای کار با scaffolding نیاز هست که به EF Code first کاملا مسلط باشد و گرنه نمی‌توانید برنامه رو دیباگ کنید یا از خطاهاش سر دربیارید.

برای پروژه‌ای نیاز بود تا از sub-domain استفاده شود. نحوه کار بدین صورت بود که اطلاعات باید بر اساس شهرها بارگذاری می‌شد؛ برای مثال اگر کاربر آدرس tehran.mysite.com را وارد کرد، اطلاعات مربوط به شهر تهران بارگذاری شود و برای سایر شهرها هم به همین ترتیب. در سرور که روش کار و ساخت ساب دومین کاملاً مشخص است. اما برای لوکال باید به چه ترتیب عمل شود؟

مسلمان برنامه پیش از آپلود حتماً باید روی لوکال تست شود. برای ساخت ساب دومین روی لوکال هاست کافیست به مسیر زیر رفته و فایل hosts را با دسترسی administrator باز کنید.

```
[windows drive]:\Windows\System32\drivers\etc
```

سپس به شیوه زیر ساب دومین‌های خود را در این فایل وارد کنید.

```
# localhost name resolution is handled within DNS itself.  
127.0.0.1 localhost  
127.0.0.1 tehran.localhost  
127.0.0.1 mashhad.localhost  
127.0.0.1 sabzevar.localhost
```

خوب، از این به بعد در صورتیکه کاربر هر کدام از ساب دومین‌های وارد شده در این فایل را در مرورگر خود وارد کند، به راحتی می‌توان در رویداد Application_BeginRequest در فایل Global.asax مسیر url را بدست آورد و کاربر را به مسیر دلخواه هدایت کرد.

نکته مهمی که باید حتماً توجه شود استفاده از دستور Request.Url.ToString(); است. هنگامیکه برنامه روی هاست در حال اجرای است، قادر به برگرداندن مسیر به همراه با نام ساب دومین می‌باشد، اما این دستور روی لوکال، نام ساب دومین را برنمی‌گرداند! برای این کار روی لوکال استفاده از روش زیر استفاده کنید:

```
host = Request.Headers["HOST"].ToLower();
```

اکنون بعد از به دست آوردن مسیر کاربر می‌توانیم او را به مسیر دلخواه بفرستیم. مثلاً من ابتدا با به دست آوردن نام شهر از روی url کاربر، کد متناظر با شهر را پیدا کرده و کد را به همراه آدرس به صفحه مورد نظر فرستاده و در آن صفحه با دریافت کد شهر، اطلاعات لازم را از دیتابیس فراخوانی می‌کنم:

```
void Application_BeginRequest(object sender, EventArgs e)
{
    string host, url, newUrl;
    host = Request.Headers["HOST"].ToLower(); //for localhost
    //host = Request.Url.ToString(); // for server
    url = Request.Url.PathAndQuery.ToLower();

    if (host.EndsWith(".localhost:8001") && (url == "" || url == "/"))
    {
        int cityId = City.GetCityIdByName(host.Replace(".localhost:8001", ""));
        newUrl = string.Format("~/postlist/{0}/", cityId);
        HttpContext.Current.RewritePath(newUrl);
        return;
    }
}
```

نکته: در صورتی که در حال استفاده از vpn باشد قادر به دسترسی به subdomains خود در حالت لوکال نخواهد بود. پس برای تست برنامه، حتماً این مورد را مد نظر داشته باشد.

موفق باشید.

نظرات خوانندگان

نویسنده: فواد عبدالله
تاریخ: ۱۳۹۱/۱۲/۱۱ ۱۲:۴۴

ممنون از آموزشتون ولی یه سوال برام پیش اومد و اونم اینکه شما تو فایل hosts به صورت استاتیک اسمی رو تعریف کردین ولی در عمل بیشتر نیاز به داینامیک بودن این مطلب هست.

نویسنده: محسن
تاریخ: ۱۳۹۱/۱۲/۱۱ ۱۳:۱۹

IIS مازولی به نام URL Rewrite برای اینکار داره. با برنامه نویسی هم قابل کنترل است.

نویسنده: ناصر فرجی
تاریخ: ۱۳۹۱/۱۲/۱۱ ۱۳:۲۴

بله. روی لوکال هدف تست برنامه ست. برای هاست هم در صورتی که بخاید به صورت بی نهایت ساب دومین داشته باشد (شبیه سیستم هایی مثل بلاگفا) باید یک ip اختصاصی داشته باشد و تنظیماتی رو در DNS خودتون اعمال کنید. به این صورت کاربر هر زیر دامنه ای رو روی دامین شما باز کنه به صورت اتوماتیک به وب سایت شما هدایت میشه. در صورتی هم که تعداد ساب دومین هاتون محدوده که میتوانید خودتون از طریق کنترل پنل دامنه اونها رو ایجاد کنید.

نویسنده: ایمان اسلامی
تاریخ: ۱۳۹۲/۰۹/۲۰ ۲۰:۵۷

با سلام و تشکر از مطالب شما میخواستم بدونم میشه یه پروژه از نوع mvc و یه پروژه از نوع web form رو به صورت subdomain رو iis نصب کرد؟ در واقع من میخوام cms سایت رو با mvc انجام بدم و خود سایت رو با web from و به صورت ساب دومین مثل:
`cms.mysite.com`
`mysite.com`

ممنون.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۲۰ ۲۱:۱۲

مشکلی نداره. برای هر کدام یک application pool جدا درست کنید تا تداخل نداشته باشند.

نویسنده: میثم فغفوری
تاریخ: ۱۳۹۲/۱۱/۱۲ ۱۹:۲۳

ممنون بابت مطلبتون. متند [Request.Headers["HOST"]] بر روی هاست نام ساب دامین‌ها رو بر میگردونه؟ اگر نه چجوری میشه فهمید برنامه به صورت لوکال اجرا شده یا بر روی سرور؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۲ ۱۹:۲۸

از خاصیت [IsLocal](#) استفاده کنید. پشت صحنه‌ی [Host](#).

نویسنده: رسول
تاریخ: ۱۱:۳۳ ۱۳۹۲/۱۲/۱۴

سلام.
من این روش را توى لوکال بدون مشکل انجام دادم
اما بر روی vps باید چیکار کنم؟
توى dns رفتم و new alias ساختم ولی کار نداد.

نویسنده: میشم سام
تاریخ: ۱۲:۲ ۱۳۹۳/۱۲/۱۹

سلام من نه روی لوکال تونستم جواب بگیرم نه روی سرور
خیلی برام مهم
اگه بتونید یه نمونه از پروژه برام بفرستید خیلی ممنون میشم

نویسنده: ح مراداف
تاریخ: ۱۸:۷ ۱۳۹۴/۰۲/۰۵

برای من ارور 404 میده و اصلا `BeginRequest` اجرا نمیشه...
(مثل حالته که WildCardDNS روی سرور تنظیم نشده باشد و درخواست‌ها به سایت اصلی بر نگردید)

به نظر "وقتی برنامه رو با `development server` خود `vs` اجرا میکنم ساب دومین‌ها کار میکنه ... با `iis express` که تست میکنم
هم خطای "bad request 400" میده

دقیقا فکر کنم مشکل من هم همین باشد که با `iis express` دارم برنامه رو اجرا می‌کنم. میشه دوستان درباره `development server` خود `vs` توضیح بدند.

نویسنده: مهدی مقدسی
تاریخ: ۱۵:۸ ۱۳۹۴/۰۳/۲۱

من میخواستم این کارو در هاست انجام بدم. میشه مراحل و منابع برای معرفی کنید.

نویسنده: وحید نصیری
تاریخ: ۱۵:۳۳ ۱۳۹۴/۰۳/۲۱

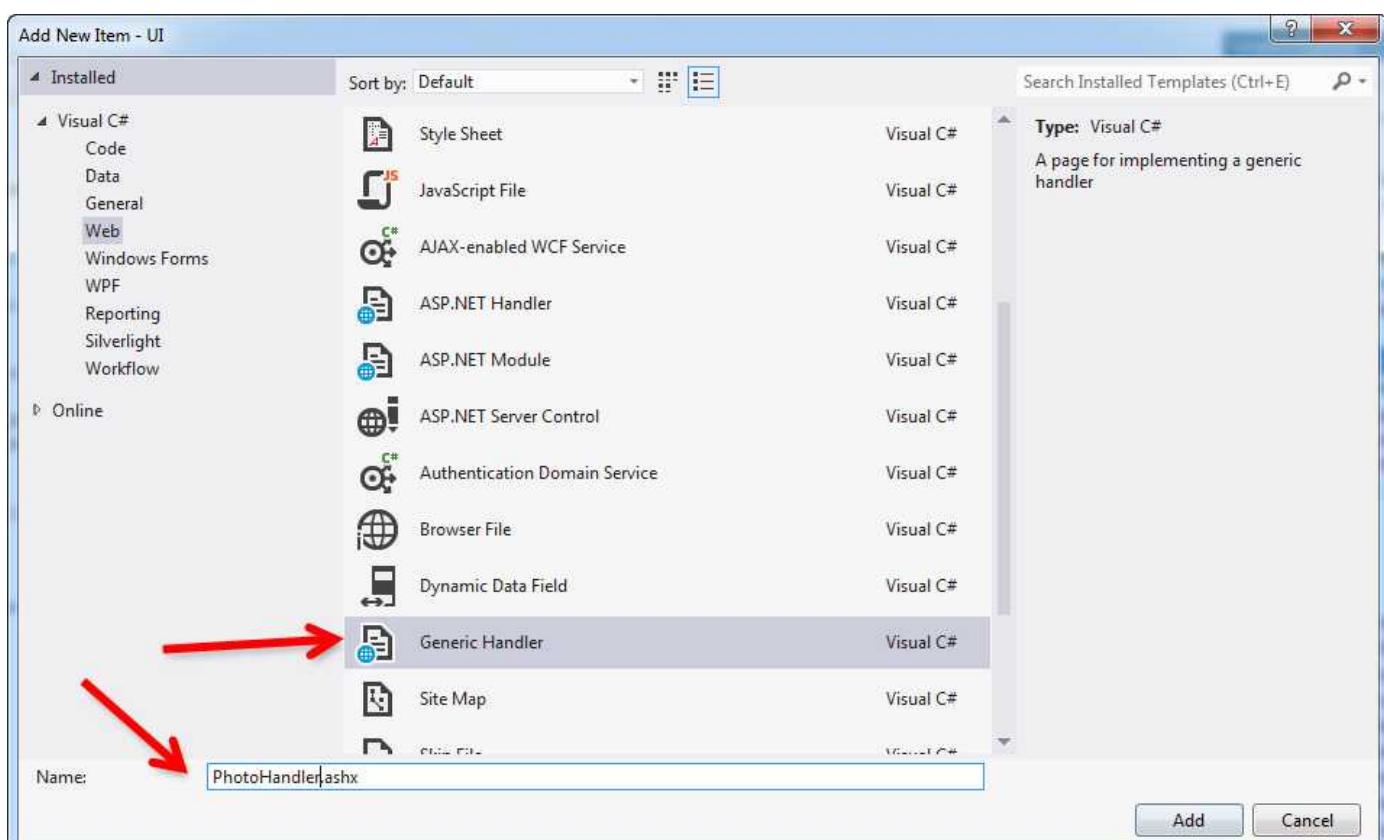
به کنترل پنل دومینی که ثبت کردید وارد شوید. در آنجا امکان ثبت subdomain ۱۳۹۴/۰۳/۲۱ هست. چیزی مثل مطلب «[بلاگر و دومین](#) [سفارشی](#)». ظاهر و اینترفیس این عملیات در هاست‌های مختلف، متفاوت است؛ ولی معاهیم اصلی آن خیر.

در ادامه مطلب [تغییر اندازه تصاویر #1](#)، در این پست می‌خواهیم نحوه تغییر اندازه تصاویر را در زمان درخواست کاربر بررسی کنیم.

در پست قبلی بررسی کردیم که کاربر می‌تواند در دو حالت تصاویر دریافتی از کاربران سایت را تغییر اندازه دهد، یکی در زمان ذخیره سازی تصاویر بود و دیگری در زمانی که کاربر درخواست نمایش یک تصویر را دارد.

خوب ابتدا فرض می‌کنیم برای نمایش تصاویر چند حالت داریم مثلاً کوچک، متوسط، بزرگ و حالت واقعی (اندازه اصلی). البته دقت نمایید که این طبقه بندی فرضی بوده و ممکن است برای پروژه‌های مختلف این طبقه بندی متفاوت باشد. (در این پست قصد فقط اشنازی با تغییر اندازه تصاویر است و شاید کد به درستی refactor نشده باشد).

برای تغییر اندازه تصاویر در زمان اجرا یکی از روش‌ها، می‌تواند استفاده از [Handler](#) باشد. خوب برای ایجاد Handler ابتدا در پروژه وب خود بروی پروژه راست کلیک کرده، و گزینه New Item را برگزینید، و در پنجره ظاهر شده مانند تصویر زیر گزینه Generic Handler را انتخاب نمایید.



پس از ایجاد هندرلر، فایل کد آن مانند زیر خواهد بود، ما باید کدهای خود را در متد `ProcessRequest` بنویسیم.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
```

```
namespace PWS.UI.Handler
{
    /// <summary>
    /// Summary description for PhotoHandler
    /// </summary>
    public class PhotoHandler : IHttpHandler
    {

        public void ProcessRequest(HttpContext context)
        {
            context.Response.ContentType = "text/plain";
            context.Response.Write("Hello World");
        }

        public bool IsReusable
        {
            get
            {
                return false;
            }
        }
    }
}
```

خوب برای نوشتن کد در این مرحله ما باید چند کار انجام دهیم.

1- گرفتن پارامترهای ورودی کاربر جهت تغییر سایز از طریق روش‌های انتقال مقادیر بین صفحات (در اینجا استفاده از [Query](#) .([String](#)

2- بازیابی تصویر از دیتابیس یا از دیسک به صورت یک آرایه بایتی.

3- تغییر اندازه تصویر مرحله 2 و ارسال تصویر به خروجی.

```
using System;
using System.Data.SqlClient;
using System.Diagnostics;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Drawing.Imaging;
using System.Globalization;
using System.IO;
using System.Web;

namespace PWS.UI.Handler
{
    /// <summary>
    /// Summary description for PhotoHandler
    /// </summary>
    public class PhotoHandler : IHttpHandler
    {

        /// <summary>
        /// بازیابی تصویر اصلی از بانک اطلاعاتی
        /// </summary>
        /// <param name="photoId">کد تصویر</param>
        /// <returns></returns>
        private byte[] GetImageFromDatabase(int photoId)
        {
            using (var connection = new SqlConnection("ConnectionString"))
            {
                using (var command = new SqlCommand("Select Photo From tblPhotos Where Id = @PhotoId",
connection))
                {
                    command.Parameters.Add(new SqlParameter("@PhotoId", photoId));
                    connection.Open();
                    var result = command.ExecuteScalar();
                    return ((byte[])result);
                }
            }
        }

        /// <summary>
        /// بازیابی فایل از دیسک
        /// </summary>
        /// <param name="photoId">با فرض اینکه نام فایل این است</param>
        /// <returns></returns>
```

```

private byte[] GetImageFromDisk(string photoId /* or something */)
{
    using (var sourceStream = new FileStream("Original File Path + id", FileMode.Open,
FileAccess.Read))
    {
        return StreamToByteArray(sourceStream);
    }
}

/// <summary>
/// Streams to byte array.
/// </summary>
/// <param name="inputStream">The input stream.</param>
/// <returns></returns>
/// <exception cref="System.ArgumentException"></exception>
static byte[] StreamToByteArray(Stream inputStream)
{
    if (!inputStream.CanRead)
    {
        throw new ArgumentException();
    }

    // This is optional
    if (inputStream.CanSeek)
    {
        inputStream.Seek(0, SeekOrigin.Begin);
    }

    var output = new byte[inputStream.Length];
    int bytesRead = inputStream.Read(output, 0, output.Length);
    Debug.Assert(bytesRead == output.Length, "Bytes read from stream matches stream length");
    return output;
}

/// <summary>
/// Enables processing of HTTP Web requests by a custom HttpHandler that implements the <see
 cref="T:System.Web.IHttpHandler" /> interface.
/// </summary>
/// <param name="context">An <see cref="T:System.Web.HttpContext" /> object that provides
references to the intrinsic server objects (for example, Request, Response, Session, and Server) used
to service HTTP requests.</param>
public void ProcessRequest(HttpContext context)
{
    // Set up the response settings
    context.Response.ContentType = "image/jpeg";
    context.Response.Cache.SetCacheability(HttpCacheability.Public);
    context.Response.BufferOutput = false;

    مرحله اول
    int size = 0;
    switch (context.Request.QueryString["Size"])
    {
        case "S":
            size = 100; //100px
            break;
        case "M":
            size = 198; //198px
            break;
        case "L":
            size = 500; //500px
            break;
    }
    byte[] changedImage;
    var id = Convert.ToInt32(context.Request.QueryString["PhotoId"]);
    byte[] sourceImage = GetImageFromDatabase(id);
    // پا
    //byte[] sourceImage = GetImageFromDisk(id.ToString(CultureInfo.InvariantCulture));

    // مرحله 2
    if (size != 0) // غير از حالت واقعی تصویر
    {
        changedImage = Helpers.ResizeImageFile(sourceImage, size, ImageFormat.Jpeg);
    }
    else
    {
        changedImage = (byte[])sourceImage.Clone();
    }

    // مرحله 3
    if (changedImage == null) return;
    context.Response.AddHeader("Content-Length",

```

```
changedImage.Length.ToString(CultureInfo.InvariantCulture));
    context.Response.BinaryWrite(changedImage);
}

public bool IsReusable
{
    get
    {
        return false;
    }
}
}
```

در این هندر ما چند متد اضافه کردیم.

1- متد **GetImageFromDatabase** : این متد یک کد تصویر را گرفته و آن را از بانک اطلاعاتی بازیابی میکند. (در صورتی که تصویر در بانک ذخیره شده باشد)

2- متد **GetImageFromDisk** : این متد نام تصویر (با فرض اینکه یک عدد می‌باشد) را به عنوان پارامتر گرفته و آنرا بازیابی می‌کند (در صورتی که تصویر در دیسک ذخیره شده باشد).

3- متد **StreamToByteArray** : زمانی که تصویر از فایل خوانده می‌شود به صورت Stream است این متد یک Stream را گرفته و تبدیل به یک آرایه بایتی می‌کند.

در نهایت در متد **ProcessRequest** تصویر خوانده شده با توجه به پارامترهای ورودی تغییر اندازه داده شده و در نهایت به خروجی نوشته می‌شود.

برای استفاده این هندر، کافی است در توصیر خود به عنوان مسیر رشته‌ای شبیه زیر وارد نمایید:

```
PhotoHandler.ashx?PhotoId=10&Size=S  
مانند  
<img src='PhotoHandler.ashx?PhotoId=10&Size=S' alt=' تصویر از مایشی' />
```

پ.ن : هر چند می‌توانستیم کد هارا بهبود داده و خیلی بینه‌تر بنویسیم اما هدف فقط اشنایی با عمل تغییر اندازه تصویر در زمان اجرا بود، امیدوارم اساتید من ببخشن.

نظرات اقای موسوی تا حدودی اعمال شد و در پست تغییراتی انجام شد.
موفق و موید باشید

نظرات خوانندگان

نویسنده: مهدی موسوی
تاریخ: ۱۳۹۱/۱۲/۱۱ ۱۳:۳۴

سلام.

چند نکته جهت بهبود کیفیت کد نوشته شده:

شما رو که `changedImage` هستش دارید، به چه دلیل اونو به `Stream` تبدیلش می‌کنید؟ چرا با استفاده از `Context.Response.BinaryWrite` همون آرایه `changedImage` رو بهش نمی‌دید؟ اینطوری دیگه به اون حلقه و همچنین تبدیل `Stream` به `byte[]` و مجدداً خوندن از `Stream` نیازی نخواهد بود.

رو بهتر نیست یک بار یک جا تعریف کنید و از اون در طول `Function` استفاده کنید؟ یا حتی `Property` ای برای اینکار تعریف کنید و هر جا لازم بود اونو `Call` کنید؟

شدن `Stream Dispose` کار صحیحی نیست وقتی قرار نیست کاری با اون `Exception` انجام بشه. از اون بدتر، در .NET 4.5 به `Catch` کردن `Exception` دیگه ای رو میتوانه در پی داشته باشه.

در پیاده سازی `StreamToArray`، برخی از کدهایی که نوشته اید باید حقیقتاً `ASSERT` باشن، نه اینکه در `Runtime` اونها رو چک کنید و ...

موفق باشید.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۱/۱۲/۱۱ ۱۷:۵۴

- 1- بله درسته، حواسم به چنین متدی نبود، در کد پست اعمال کردم.
- 2- بله می‌توان اینکار را کرد و حتی بهبودهای بیشتری روی آن داد. اما قصد اموزش یک مبحث بود والا می‌توانستیم کل `if`ها و `switch`ها را حذف کنیم.
- 3- بله درسته حذفش می‌کنم.
- 4- این متد نوشته من نیست و من ارجاع دادم به منبع اصلی.

موفق باشید

نویسنده: دانشجو
تاریخ: ۱۳۹۲/۰۱/۱۹ ۹:۱

دلیل استفاده از `Handler` برای تغییر اندازه تصویر چیه؟ در [روش قبلی](#) شما با یک تابع این کار رو انجام دادین، آیا تفاوتی بین این دو روش وجود داره؟ کدام بهتر استفاده بشه؟ اگه ممکنه در مورد `Handler` توضیحی بدین. با تشکر

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۱/۱۹ ۱۳:۱۴

سلام

- دوست گلم توی پستی که [شما ارجاع دادین](#) گفته شده که برای تغییر اندازه تصاویر دو راه میشه در نظر گرفت:
- 1- در زمان ثبت، تصویر تغییر اندازه داده شود.
 - 2- در زمان نیاز تغییر اندازه داده شود.

که در مورد هر کدام توضیحاتی داده شده است، توی این پست با استفاده از هندر در زمان اجرا (و استفاده از تابع تغییر اندازه تصویر) تصویر مورد نظر تغییر اندازه داده شده است.

در مورد هندر هم لینک داده بودم توی پست.
موفق و موید باشید.

نویسنده: شاهین

تاریخ: ۲:۵۵ ۱۳۹۲/۰۲/۰۵

دوست عزیز یه مثال میذاری که چطور ازش استفاده کنم؟
هر کاری می کنم ارور میده میکه آدرس فایل اشتباهه

نویسنده: صابر فتح الهی

تاریخ: ۱۰:۱۵ ۱۳۹۲/۰۲/۰۵

سلام

دوست من مشکلتون کجاست، لطفا کدهاتون بذارین که بهتر بتونم شمارو راهنمایی کنم.
کاری که باید بکنین

- 1- نوشتن متده `ResizeImage`
- 2- ایجاد یک هندر مانند بالا
- 3- استفاده در نمایش تصاویر که مثالش آخر پست هست.

نویسنده: شاهین

تاریخ: ۱۵:۳۷ ۱۳۹۲/۰۲/۰۵

مرسی بابت جواب دادنتون این خط کد رو چطوری باید بنویسم؟
`Original File Path + id`

نویسنده: صابر فتح الهی

تاریخ: ۱۶:۵۵ ۱۳۹۲/۰۲/۰۵

شما می تونین در پارامتر ورودی متده `nam` فایل و مسیر اون پاس بدین.

در خط 103 از برنامه بالا شما می توانید به جای کد زیر

```
byte[] sourceImage = GetImageFromDatabase(id);
```

از کدی شبیه به این استفاده نمایید

```
byte[] sourceImage = GetImageFromDisk(  
    Path.Combine(context.Server.MapPath("~/uploads"), Path.GetFileName(id.ToString())));
```

که مسیر `uploads` یک مسیر دلخواه است که فایل های شما در ان قرار گرفته است. نحوه فراخوانی در این حالت به شکل زیر خواهد بود

```
PhotoHandler.ashx?PhotoId=test.jpg&Size=S
```

مانند

```
<img src='PhotoHandler.ashx?PhotoId=test.jpg&Size=S' alt='تصویر ازمايشي' />
```

البته باید تبدیل نوع ورودی در خط 102 تغییر کند(نام فایل از نوع رشته ای است)، هر چند می‌توان کدها را بهتر کرد.

پ.ن: نام فایل می‌تواند test.jpg یا هر نامی باشد که در فولدر مورد نظر وجود دارد. البته برای این کار باید تمیز کاری روی ورودی انجام شود که می‌توانید از این [پست](#) بهره بگیرید.
موفق و موید باشید

نویسنده: رضا گرمارودی
تاریخ: ۱۳۹۲/۱۲/۱۲ ۱۳:۳۳

سلام؛ هر دو مطلب شما را مطالعه کردم. همچنین لینک هندر شما که به مایکروسافت و MSDN ارجاع شده بود و در مشکلی ندارم اما در MVC هندر را نمی‌شناسد. قسمت زیر را هم به وبکانفیگ اضافه کردم ولی موثر واقع نشد. لطفا راهنمایی بفرمایید برای استفاده از هندر همانند هندر فوق در MVC چکار باید انجام داد.

```
<system.webServer>
    <handlers>
        <add name="ImageHandler" path="ImageHandler.ashx" verb="*" type="ImageHandler" />
    </handlers>
</system.webServer>
```

نسخه مورد استفاده: mvc5 , visulstudio2013

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۱۲ ۱۳:۳۹

هندر را می‌توان تبدیل به یک فیلتر و یا حتی یک ActionResult کرد. برای مثال در مورد تصاویر «[اضافه کردن Watermark به تصاویر یک برنامه ASP.NET MVC در صورت لینک شدن در سایتی دیگر](#)»

در تکمیل [این مطلب](#) برای حذف هدرهای مربوط به وب سرور در برنامه‌های ASP.NET MVC از روش زیر می‌توانیم استفاده کنیم.

در حالت پیش فرض تمام پاسخهای که به سمت سرور ارسال می‌شوند به همراه خود یک سری جزئیات را نیز منتقل می‌کنند.

Transformer | Headers | TextView | ImageView | HexView | WebView | Auth | Caching | Privacy

Response Headers

HTTP/1.0 200 OK

- Cache
 - ... Cache-Control: private
 - ... Date: Sun, 28 Oct 2012 21:33:46 GMT
 - ... X-Cache: MISS from none
- Entity
 - ... Content-Length: 15929
 - ... Content-Type: text/html; charset=utf-8
- Miscellaneous
 - ... Server: Microsoft-IIS/6.0
 - ... X-AspNetMvc-Version: 3.0
 - ... X-AspNet-Version: 4.0.30319
 - ... X-Cache-Lookup: MISS from none:80
 - ... X-Powered-By: ASP.NET
- Transport
 - ... Connection: keep-alive
 - ... Via: 1.0 none (squid)

برای یک وب اپلیکیشن ASP.NET MVC این هدرها را داریم :

Server: که توسط IIS اضافه می‌شود.

X-AspNet-Version: که در زمان Flush در httpResponse اضافه می‌شود.

X-AspNetMvc-Version: که توسط System.Web.dll در MvcHandler اضافه می‌شود.

X-Powered-By: این مورد نیز توسط IIS اضافه می‌شود.

هکرها از اینکه فریم ورک مورد استفاده چه چیزی است خوشحال خواهند شد: اگر سرور شما برای مدتی Update نشده باشد و یک آسیب پذیری امنیتی بزرگ برای ورژن فریم ورکی که استفاده می‌کنید پیدا شود در نتیجه به هکرها برای رسیدن به هدفشان

کمک کرده اید.

به علاوه این هدرها فضایی را برای تمام پاسخها در نظر میگیرند (البته در حد چندین بایت ولی در اینجا بحث برروی Optimization است).

برای حذف این هدرها باید مراحل زیر را انجام دهیم:

حذف کردن هدر Server : به Global.asax.cs رفته و رویداد Application_PresendRequestHeaders با کد زیر را به آن اضافه کنید :

```
protected void Application_PresendRequestHeaders(object sender, EventArgs e)
{
    var app = sender as HttpApplication;
    if (app == null || !app.Request.IsLocal || app.Context == null)
        return;
    var headers = app.Context.Response.Headers;
    headers.Remove("Server");
}
```

حذف کردن هدر X-AspNetMvc-Version : در فایل Application_Start به رویداد Global.asax.cs این کد زیر را اضافه کنید :

```
protected void Application_Start()
{
    ...
    MvcHandler.DisableMvcResponseHeader = true;
    ...
}
```

حذف کردن هدر X-AspNet-Version : به فایل Web.Config مراجعه کرده و این المتن را در داخل system.web اضافه کنید:

```
<system.web>
    ...
    <httpRuntime enableVersionHeader="false" />
    ...
</system.web>
```

حذف کردن هدر X-Powered-By : در داخل فایل Web.Config در داخل system.webServer این خطوط را اضافه کنید:

```
<system.webServer>
    ...
    <httpProtocol>
        <customHeaders>
            <remove name="X-Powered-By" />
        </customHeaders>
    </httpProtocol>
    ...
</system.webServer>
```

با انجام مراحل فوق پاسخهای سرور سبکتر شده و در نهایت حاوی اطلاعات مهم در مورد ورزش فریم ورک نمیباشد.

نظرات خوانندگان

نویسنده: sysman | تاریخ: ۱۳۹۲/۰۱/۱۴ | ساعت: ۲۲:۰۰

ممنون از این مطلب مفید ولی من یک مشکلی دارم کاری که گفتید را انجام دادم و همه چیز خوب انجام شد ولی در بخش modernizr-2.5.3.js باز هم اطلاعات سرور رو به من میدهد

The screenshot shows the Network tab of a browser's developer tools. It lists three requests:

- GET site.css (status 30)
- GET modernizr-2.5.3.js (status 30)
- GET jquery-1.9.1.js (status 20)

For the GET modernizr-2.5.3.js request, the Headers tab is selected, showing the following response headers:

Header	Value
Accept-Ranges	bytes
Date	Tue, 02 Apr 2013 19:47:23 GMT
Etag	"f73f4a0b72fce1:0"
Server	Microsoft-IIS/8.0
X-SourceFiles	=?UTF-8?B?RDpcVmlzdWFsIFN0dWRpb?=

The Response tab is also visible below the Headers tab.

نویسنده: علی | تاریخ: ۱۳۹۲/۰۱/۱۴ | ساعت: ۳۶:۰۰

فایل‌های استاتیک رو هم باید به موتور ASP.NET مپ کنید. تا زمانیکه مپ نباشند مستقیماً توسط IIS سرو می‌شن و تنظیمات روی اون‌ها تاثیری نداره.

نویسنده: sysman | تاریخ: ۱۳۹۲/۰۱/۱۴ | ساعت: ۱۰:۴۰

نمی دونم این کاری که گفتین رو دقیقاً چطور انجام بدhem ولی این کار باعث ایجاد کندی نمی‌شه؟ اگر در خود IIS تنظیمات [این پست](#) رو اعمال کنم به نتیجه میرسم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۱۴ ۱۱:۱۳

- برای فایل‌های جاوا اسکریپت توصیه من این است:
- (الف) اگر از Web forms استفاده می‌کنید: استفاده از Script manager (^ و ^ Script manager)
- (ب) اگر از MVC استفاده می‌کنید: استفاده از Bundling & minification در هر دو حالت نحوه ارائه اسکریپت‌ها تحت کنترل برنامه ASP.NET در خواهد آمد و مستقیماً و بدون دخالت IIS در سمت سرور فعال شود، این مساله بار اضافه‌ای را به سرور تحمیل نخواهد کرد.
- برای می‌پ کردن فایل‌های استاتیک به موتور ASP.NET می‌شود از StaticFileHandler استفاده کرد. اگر کش کردن اطلاعات استاتیک در سمت سرور فعال شود، این مساله بار اضافه‌ای را به سرور تحمیل نخواهد کرد.

```
<system.web>
  <httpHandlers>
    <add path="*.js" verb="*" type="System.Web.StaticFileHandler" />
  </httpHandlers>
```

نویسنده: یونس دوست
تاریخ: ۱۳۹۲/۰۶/۰۲ ۱۱:۳۴

سلام. من از این استفاده کردم ولی موقع حذف Header مربوط به سرور ارور زیر را میدم:
This operation requires IIS integrated pipeline mode
ضمناً من از iis 7.5 استفاده می‌کنم و توی Application Pools هم Classic .Net AppPool رو در حالت pipeline قرار دادم هم DefaultAppPool ولی همچنان ارور رو دارم. مشکل از کجاست؟
ممnon.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۶/۰۲ ۱۱:۵۴

- محل تست کردن واقعی این کدها بر روی ویندوز سرور است و نه دیباگر و وب سرور آزمایشی ویژوال استودیو.
- شما اگر تا این حد دسترسی به IIS دارید، اصلاً نیازی به کدنویسی برای حذف هدرهای وب سرور نخواهید داشت. به قسمت URL کنسول مدیریتی مراجعه و مداخل موجود را حذف یا ویرایش کنید.
- + حذف هدر مربوط به نام Server کار ساده‌ای نیست. خیلی‌ها از روش HTTP module هم جواب نگرفته‌اند، اما با استفاده از URL Scan خود مایکروسافت قابل حذف است (این برنامه روی ویندوز‌های سرور 2003 به بعد قابل نصب است). بعد از نصب به فایل C:\Windows\System32\inetsrv\UrlScan\UrlScan.ini RemoveServerHeader را با 1 مقدار دهی کنید. ضمناً قبل از نصب URLScan تغییر زیر را هم امتحان کنید (بجای Remove از Set استفاده شده):

```
void OnPreSendRequestHeaders(object sender, EventArgs e)
{
  HttpContext.Current.Response.Headers.Set("Server", "CERN httpd");
}
```

نویسنده: پیام دات نت
تاریخ: ۱۳۹۳/۰۳/۱۵ ۲۰:۳۵

سلام؛ ممنون از مطلب خوبتون. من تو IIS 8 Local اجرا کردم درست کار می‌کنه و بسیار عالیه. اما تو اجرای نهایی روی سرور هدر Server حذف نشد و از سمت سرور ارسال می‌شد. بعد از کلی جستجو به نتیجه زیر رسیدم.

```
<system.webServer>
  <rewrite>
    <outboundRules>
      <rule name="Remove RESPONSE_Server" >
        <match serverVariable="RESPONSE_Server" pattern=".+" />
        <action type="Rewrite" value="" />
      </rule>
    </outboundRules>
  </rewrite>
</system.webServer>
```

نوبنده: دانش پژوه
تاریخ: ۱۳۹۳/۱۰/۰۲ ۱۱:۳۹

ممنون از مطلبتون

فقط یک نکته رو میخواستم بگم نام کوکی سشن بطور پیشفرض ASP.NET_SessionId هست و میشه از این نام فهمید که سرور asp.net هستش. از طریق کد زیر توی وب کافیگ میتوانید تغییرش بدید:

```
<system.web>
  <sessionState cookieName="foo" />
</system.web>
```

همان طور که می‌دانید در css امکان استفاده از فونت‌های فارسی مهیا است. برای این کار کافیست با دستور زیر فونت را در فایل css خود تعریف کنیم و در صورتیکه فونت روی سیستم کاربر موجود نباشد ابتدا فونت روی سیستم دانلود شده و سپس نمایش داده می‌شود. استفاده از سه پسوند مختلف نیز برای مرورگرهای مختلف در نظر گرفته شده است تا خروجی در تمامی مرورگرها به درستی نمایش داده شود.

```
@font-face {
    src: url('Font/BYekan.eot?#') format('eot'), /* IE6-8 */
         url('Font/BYekan.woff') format('woff'), /* FF3.6+, IE9, Chrome6+, Saf5.1+*/
         url('Font/BYekan.ttf') format('truetype'); /* Saf3-5, Chrome4+, FF3.5, Opera 10+ */
}
```

در یک پروژه با مشکل عجیبی روبرو شدم . پروژه روی لوکال به خوبی کار می‌کرد اما بعد از آپلود روی سرور فونت‌ها بسیار دیر لود می‌شد. یعنی ابتدا تمامی اجزای صفحه لود شده و با تأخیری چند ثانیه ای فونت نمایش داده می‌شد. مرورگری که در حال کار روی آن برای تست پروژه بودم باید از پسوند woff مربوط به آن فونت استفاده می‌کرد. بعد از تست پروژه با فایرباگ متوجه شدم که متأسفانه اصلاً فونت روی سیستم دانلود نشده و خروجی 404 که مربوط به File Not Found است برگشت داده می‌شد. سعی کردم با دادن آدرس فونت در برنامه download manager فونت را دانلود کنم. اما باز هم فونت دانلود نمی‌شد! (یعنی فایل روی سرور موجود بود ولی هیچ دسترسی به آن ممکن نبود) برای دو پسوند دیگر یعنی eot و ttf هم این کار را تست کردم و متوجه شدم روی این دو پسوند مشکل خاصی وجود ندارد. با تماس با پشتیبانی هاست متوجه شدم اصلاً این پسوند در Mime Type های سرور تعریف نشده است . بنابراین به صورت دستی Mime Type مورد نظر را روی سرور تعریف کردم و مشکل حل شد:

Extension	MIME-Type	
.woff	application/x-font-woff	

همچنین لیست کامل Mime Types ها در آدرس زیر موجود است:

[لیست کامل Mime Types](#)

نظرات خوانندگان

نویسنده: آموزش مجازی
تاریخ: ۱۳۹۱/۱۲/۲۹ ۱۵:۲۷

شما اگر این تنظیمات رو توی پروژه خودت انجام بدی ، احتیاجی نیست برای IISMimeType تعريف کنی. البته فقط قسمتهایی از این تنظیمات در مورد مساله ای که گفتی لازم هست اما من برای ایده گرفتن در مورد تنظیمات 7 IIS و به بالا در web.config که کلا مهجور مونده ، تنظیماتی که خودم معمولاً میذارم رو گذاشتم اینجا.

```
<system.webServer>
  <urlCompression doStaticCompression="true" doDynamicCompression="true" />
<staticContent>
  <remove fileExtension=".js"/>
  <mimeType fileExtension=".js" mimeType="text/javascript" />
  <mimeType fileExtension=".mp4" mimeType="video/mp4" />
  <mimeType fileExtension=".m4v" mimeType="video/m4v" />
  <mimeType fileExtension=".ogg" mimeType="video/ogg" />
  <mimeType fileExtension=".ogv" mimeType="video/ogg" />
  <mimeType fileExtension=".webm" mimeType="video/webm" />
  <mimeType fileExtension=".oga" mimeType="audio/ogg" />
  <mimeType fileExtension=".spx" mimeType="audio/ogg" />
  <mimeType fileExtension=".svg" mimeType="images/svg+xml" />
  <mimeType fileExtension=".svgz" mimeType="images/svg+xml" />
  <mimeType fileExtension=".eot" mimeType="application/vnd.ms-fontobject" />
  <mimeType fileExtension=".otf" mimeType="font/otf" />
  <mimeType fileExtension=".woff" mimeType="font/x-woff" />
</staticContent>
```

نویسنده: آرمان فرقانی
تاریخ: ۱۳۹۱/۱۲/۲۹ ۱۶:۳۹

البته با توجه به پردازش سلسله مراتبی فایل های config در وب سرور احتمال دارد در برخی میزبان های اشتراکی برخی از قابلیت ها توسط مدیر سرور Lock شود.

نویسنده: ناصر فرجی
تاریخ: ۱۳۹۲/۰۱/۰۱ ۱۵:۴۱

بله. تمامی تنظیمات iis رو به نوعی میشه از طریق web.config هم انجام داد ولی برای اعمال سراسری تنظیمات بهتره از iis استفاده بشه.

عموما در برنامه های وب برای نمایش فایل های پویای باینری تولید شده، یا ابتدا آن ها را بر روی سخت دیسک ذخیره کرده و مسیر نهایی را به نحوی به کاربر نمایش می دهند و یا فایل را بدون ذخیره سازی، در مرورگر کاربر اصطلاحا `Flush` می کنند. حالت `Flush` سبب نمایش صفحه دیالوگ ذخیره سازی فایل گردیده و در همینجا `Response` خاتمه خواهد یافت.

برای نمونه در اینجا توسط متده `inMemoryFile` در حافظه تشکیل شده و سپس به صورت یک `Byte Array` بازگشت داده می شود. در ادامه کار، این اطلاعات در مرورگر کاربر `Flush` خواهد شد:

```
using System.IO;
using System.Net.Mime;
using System.Web;

namespace WebApplication
{
    public class PdfHandler : IHttpHandler
    {
        private static byte[] inMemoryFile()
        {
            تولید پویای فایل در حافظه و یا حتی خواندن از یک نمونه موجود//
            return File.ReadAllBytes(@"D:\path\DynamicCrosstabSampleRpt.pdf");
        }

        public void ProcessRequest(HttpContext context)
        {
            var pdf = inMemoryFile();

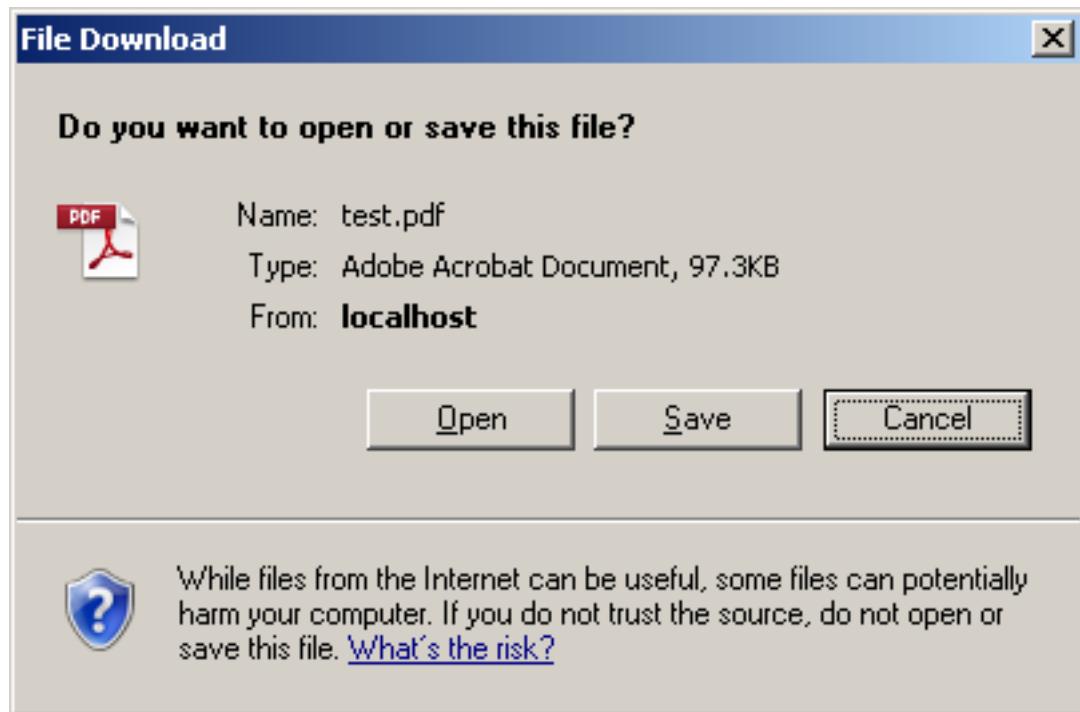
            context.Response.Cache.SetCacheability(HttpCacheability.NoCache);
            context.Response.ContentType = MediaTypeNames.Application.Pdf;
            context.Response.AddHeader("Content-Length", pdf.Length.ToString());
            context.Response.AddHeader("content-disposition", "attachment;filename=test.pdf");
            context.Response.Buffer = true;
            context.Response.Clear();
            context.Response.OutputStream.Write(pdf, 0, pdf.Length);
            context.Response.OutputStream.Flush();
            context.Response.OutputStream.Close();
            context.Response.End();
        }

        public bool IsReusable
        {
            get { return false; }
        }
    }
}
```

و برای نمایش آن در یک `iframe` در صفحه:

```
<iframe width="100%" src="PdfHandler.ashx" height="200px"></iframe>
```

نتیجه کار، نمایش صفحه دیالوگ ذخیره سازی فایل به کاربر است:



سؤال: فرض کنید Adobe reader بر روی سیستم نصب است و مرورگر با استفاده از Active-X آن می‌تواند این نوع فایل‌ها را نمایش دهد. آیا راهی وجود دارد تا بجای نمایش save popup dialog، این فایل توسط مرورگر نمایش داده شود؟
پاسخ: بلی، در کدهای فوق تنها کافی است یک سطر آن تغییر کند:

```
Response.AddHeader("content-disposition", "inline;filename=test.pdf");
```

در اینجا تنها نحوه مقدار دهی content-disposition تقاویت کرده است. حالت attachment سبب نمایش save popup dialog می‌شود و مقدار inline، فایل را در مرورگر نمایش خواهد داد.
اینبار اگر برنامه را اجرا کنیم، PdfHandler.ashx اشاره می‌کند، فایل PDF را در صفحه نمایش می‌دهد.

نظرات خوانندگان

نوبتند: آرمان
تاریخ: ۱۳۹۱/۱۲/۲۸ ۱۱:۴۸

عالی بود مهندس. در صورتی که inline باشد ولی برنامه ای برای استفاده از پی دی اف نباشد یا مرورگر مجاز به استفاده نباشد باز صفحه دانلود دیده میشه؟

نوبتند: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۲۸ ۱۲:۱۲

بله. صفحه file download فوق باز می شود.

نوبتند: امیر هاشم زاده
تاریخ: ۱۳۹۳/۰۵/۲۰ ۲۲:۱۴

آیا امکان عدم دریافت فایل مذکور توسط دانلود منیجرها وجود دارد؟

نوبتند: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۲۰ ۲۲:۱۹

با یک پسوند دیگر که نمی‌شناسند تست کنید؛ مثلا filename=test.abc

نوبتند: افتتابی
تاریخ: ۱۳۹۳/۰۵/۳۱ ۰:۱۶

متشرک؛ خوب اگه من بخواهم با استفاده از PDFReport گزارش درست کنم و توی خود view نمایش بدم ، این کدها رو توی کدوم کلاس بگذارم؟

نوبتند: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۳۱ ۰:۲۴

به صورت توکار لحاظ شده:

```
.Generate(data =>
{
    fileName = HttpUtility.UrlEncode(fileName, Encoding.UTF8);
    data.FlushInBrowser(fileName, FlushType.Inline);
}); // creating an in-memory PDF file
```

آن همان مطلب جاری است.

طراحی یک معماری خوب و مناسب یکی از عوامل مهم تولید یک برنامه کاربردی موفق می‌باشد. بنابراین انتخاب یک ساختار مناسب به منظور تولید برنامه کاربردی بسیار مهم و تا حدودی نیز سخت است. در اینجا یاد خواهیم گرفت که چگونه یک طراحی مناسب را انتخاب نماییم. همچنین روش‌های مختلف تولید برنامه‌های کاربردی را که مطمئناً شما هم از برخی از این روشها استفاده نمودید را بررسی می‌نماییم و مزایا و معایب آن را نیز به چالش می‌کشیم.

ضد الگو (Antipattern) - رابط کاربری هوشمند (Smart UI)

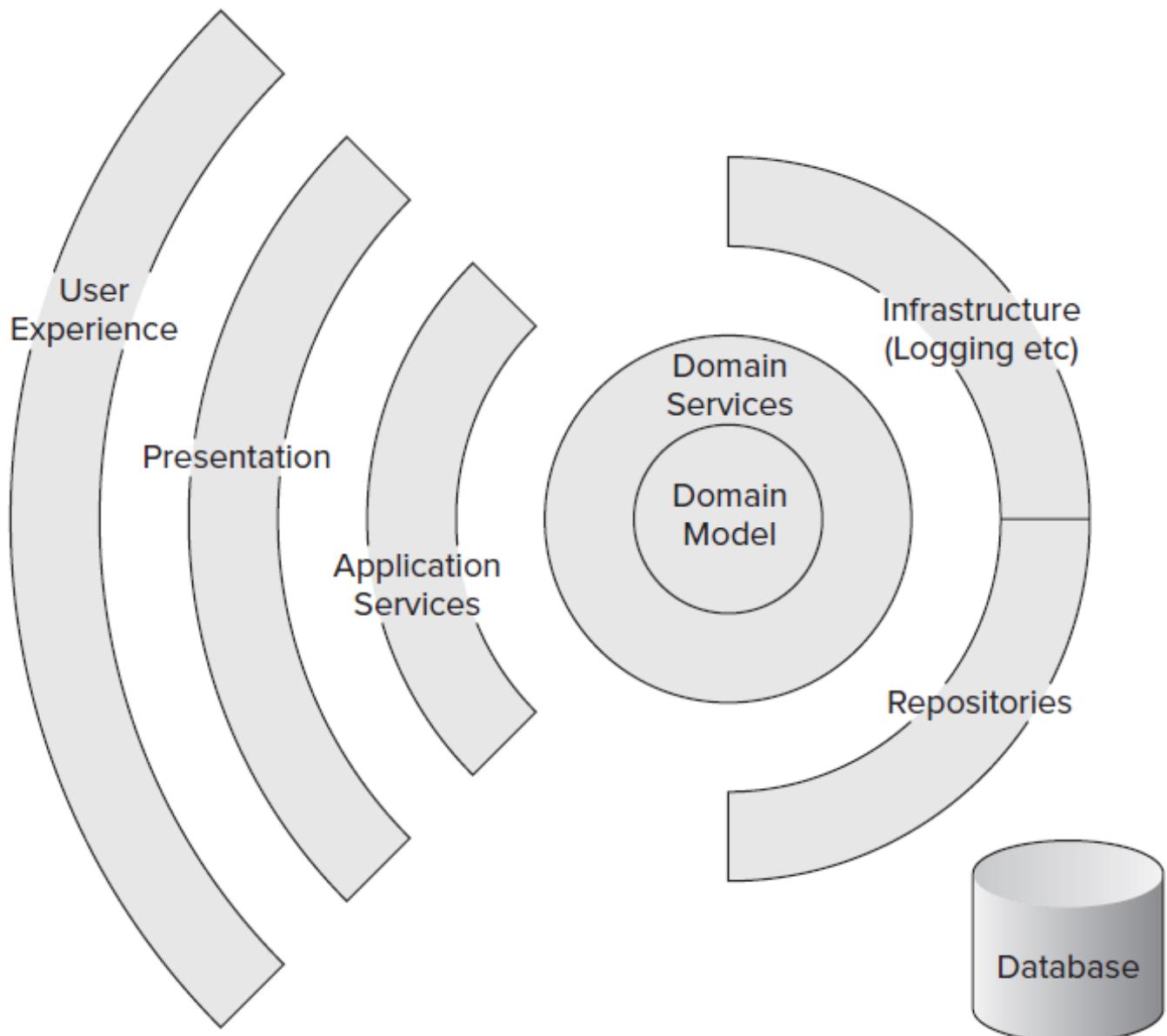
با استفاده از Visual Studio یا به اختصار VS، می‌توانید برنامه‌های کاربردی را به راحتی تولید نمایید. طراحی رابط کاربری به آسانی عمل کشیدن و رها کردن (Drag & Drop) کنترل‌ها بر روی رابط کاربری قابل انجام است. همچنین در پشت رابط کاربری (Code Behind) تمامی عملیات مربوط به مدیریت رویدادها، دسترسی به داده‌ها، منطق تجاری و سایر نیازهای برنامه کاربردی، کد نویسی خواهد شد. مشکل این نوع کدنویسی بدین شرح است که تمامی نیازهای برنامه در پشت رابط کاربری قرار می‌گیرند و موجب تولید کدهای تکراری، غیر قابل تست، پیچیدگی کدنویسی و کاهش قابلیت استفاده مجدد از کد می‌گردد.

به این روش کد نویسی Smart UI می‌گویند که موجب تسهیل تولید برنامه‌های کاربردی می‌گردد. اما یکی از مشکلات عمدی این روش، کاهش قابلیت نگهداری و پشتیبانی و عمر کوتاه برنامه‌های کاربردی می‌باشد که در برنامه‌های بزرگ به خوبی این مشکلات را حس خواهید کرد.

از آنجایی که تمامی برنامه نویسان مبتدی و تازه کار، از جمله من و شما در روزهای اول برنامه نویسی، به همین روش کدنویسی می‌کردیم، لزومی به ارائه مثال در رابطه با این نوع کدنویسی نمی‌بینم.

تفکیک و جدا سازی اجزای برنامه کاربردی (Separating Your Concern)

راه حل رفع مشکل Smart UI، لایه بندی یا تفکیک اجزای برنامه از یکدیگر می‌باشد. لایه بندی برنامه می‌تواند به شکل‌های مختلفی صورت بگیرد. این کار می‌تواند توسط تفکیک کدها از طریق فضای نام (Namespace)، پوشه بندی فایلهای حاوی کد و یا جداسازی کدها در پروژه‌های متفاوت انجام شود. در شکل زیر نمونه‌ای از معماری لایه بندی را برای یک برنامه کاربردی بزرگ می‌بینید.



به منظور پیاده سازی یک برنامه کاربردی لایه بندی شده و تفکیک اجزای برنامه از یکدیگر، مثالی را پیاده سازی خواهیم کرد. ممکن است در این مثال با مسائل جدید و شیوه های پیاده سازی جدیدی مواجه شویم که این نوع پیاده سازی برای شما قابل درک نباشد. اگر کمی صبر پیش نمایید و این مجموعه ای آموزشی را پیگیری کنید، تمامی مسائل نامانوس با جزئیات بیان خواهند شد و درک آن برای شما ساده خواهد گشت. قبل از شروع این موضوع را هم به عرض برسانم که علت اصلی این نوع پیاده سازی، انعطاف پذیری بالای برنامه کاربردی، پشتیبانی و نگهداری آسان، قابلیت تست پذیری با استفاده از ابزارهای تست، پیاده سازی پروژه بصورت تیمی و تقسیم بخش های مختلف برنامه بین اعضای تیم و سایر مزایای فوق العاده آن می باشد.

1- Visual Studio را باز کنید و یک Solution خالی با نام SoCPatterns.Layered ایجاد نمایید.

2- جهت ایجاد Solution خالی، پس از انتخاب New Project ، از سمت چپ گزینه Other Project Types و سپس Solutions را انتخاب نمایید. از سمت راست گزینه Blank Solution را انتخاب کنید.

3- بر روی Solution کلیک راست نموده و از گزینه Add > New Project یک پروژه Class Library با نام SoCPatterns.Layered.Repository ایجاد کنید.

3- با استفاده از روش فوق سه پروژه Class Library دیگر با نامهای زیر را به Solution اضافه کنید:

SoCPatterns.Layered.Model

SoCPatterns.Layered.Service

SoCPatterns.Layered.Presentation

4- با توجه به نیاز خود یک پروژه دیگر را باید به Solution اضافه نمایید. نوع و نام پروژه در زیر لیست شده است که شما باید با توجه به نیاز خود یکی از پروژه‌های موجود در لیست را به Solution اضافه کنید.

(Windows Forms Application (SoCPatterns.Layered.WinUI

(WPF Application (SoCPatterns.Layered.WpfUI

(ASP.NET Empty Web Application (SoCPatterns.Layered.WebUI

(ASP.NET MVC 4 Web Application (SoCPatterns.Layered.MvcUI

5- بر روی پروژه SoCPatterns.Layered.Repository کلیک راست نمایید و با انتخاب گزینه Add Reference به پروژه‌ی SoCPatterns.Layered.Model ارجاع دهید.

6- بر روی پروژه SoCPatterns.Layered.Service کلیک راست نمایید و با انتخاب گزینه Add Reference به پروژه‌های SoCPatterns.Layered.Repository و SoCPatterns.Layered.Model ارجاع دهید.

7- بر روی پروژه SoCPatterns.Layered.Presentation کلیک راست نمایید و با انتخاب گزینه Add Reference به پروژه‌های SoCPatterns.Layered.Service و SoCPatterns.Layered.Model ارجاع دهید.

8- بر روی پروژه UI خود به عنوان مثال SoCPatterns.Layered.WebUI کلیک راست نمایید و با انتخاب گزینه Add Reference به پروژه‌های SoCPatterns.Layered.Model ، SoCPatterns.Layered.Repository ، SoCPatterns.Layered.Service و SoCPatterns.Layered.Presentation ارجاع دهید.

9- بر روی پروژه UI خود به عنوان مثال SoCPatterns.Layered.WebUI کلیک راست نمایید و با انتخاب گزینه Set as StartUp Project پروژه‌ی اجرایی را مشخص کنید.

10- بر روی Solution کلیک راست نمایید و با انتخاب گزینه Add > New Solution Folder پوشه‌های زیر را اضافه نموده و پروژه‌های مرتبط را با عمل Drag & Drop در داخل پوشه‌ی مورد نظر قرار دهید.

UI .1

SoCPatterns.Layered.WebUI §

Presentation Layer .2
SoCPatterns.Layered.Presentation §

Service Layer .3
SoCPatterns.Layered.Service §

Domain Layer .4

SoCPatterns.Layered.Model §

Data Layer .5

SoCPatterns.Layered.Repository §

توجه داشته باشید که پوشه بندی برای مرتب سازی لایه‌ها و دسترسی راحت‌تر به آنها می‌باشد.

پیاده سازی ساختار لایه بندی برنامه به صورت کامل انجام شد. حال به پیاده سازی کدهای مربوط به هر یک از لایه‌ها و بخش‌ها می‌پردازیم و از لایه Domain شروع خواهیم کرد.

نظرات خوانندگان

نویسنده: آرمان فرقانی
تاریخ: ۲۰:۲ ۱۳۹۱/۱۲/۲۸

مباحثی از این دست بسیار مفید و ضروری است و به شدت استقبال می‌کنم از شروع این سری مقالات. البته پیش‌تر هم مطالبی از این دست در سایت ارائه شده است که امیدوارم این سری مقالات بتونه تا حدی پراکنده‌گی مطالب مربوطه را از بین ببرد. فقط لطف بفرمایید در این سری مقالات مرز بندی مشخصی برای برخی مفاهیم در نظر داشته باشید. به عنوان مثال گاهی در یک مقاله مفهوم Repository معادل مفهوم لایه سرویس در مقاله دیگر است. یا Domain Model مرز مشخصی با View Model داشته باشد. همچنین بحث‌های خوبی مهندس نصیری عزیز در مورد عدم نیاز به ایجاد Repository در مفهوم متداول در هنگام استفاده از EF داشتند که در رفرنس‌های معتبر دیگر هم مشاهده می‌شود. لطفاً در این مورد نیز بحث بیشتری با مرز بندی مشخص داشته باشید.

نویسنده: حسن
تاریخ: ۲۲:۵ ۱۳۹۱/۱۲/۲۸

آیا صرفاً تعریف چند مژول مختلف برنامه را لایه بندی می‌کند و ضمانتی است بر لایه بندی صحیح، یا اینکه استفاده از الگوهای MVC و MVVM می‌توانند ضمانتی باشند بر جدا سازی حداقل لایه نمایشی برنامه از لایه‌های دیگر، حتی اگر تمام اجزای یک برنامه داخل یک اسمبلی اصلی قرار گرفته باشند؟

نویسنده: میثم خوشبخت
تاریخ: ۰:۵۳ ۱۳۹۱/۱۲/۲۹

این سری مقالات جمع بندی کامل معماری لایه بندی نرم افزار است. پس از پایان مقالات یک پروژه کامل رو با معماری منتخب پیاده سازی می‌کنم تا تمامی شک و شباهات برطرف بشه. در مورد مرزبندی لایه‌ها هم صحبت می‌کنم و مفهوم هر کدام را دقیقاً توضیح میدم.

نویسنده: میثم خوشبخت
تاریخ: ۰:۵۹ ۱۳۹۱/۱۲/۲۹

اگر مقاله فوق رو با دقت بخونید متوجه می‌شید که MVC و MVVM در لایه UI پیاده سازی می‌شن. البته در لایه Domain Model در برخی مواقع لایه Controller رو در لایه Presentation قرار میدن. در Domain Model در MVVM نیز لایه Service رو در Lایه View Model نیز در لایه Repository قرار می‌گیره. همچنین View Modelها نیز در لایه Model قرار می‌گیرن.

در مورد مژول بندی هم اگر در مقاله خونده باشید می‌توانید لایه‌ها رو از طریق پوشش‌ها، فضای نام و یا پروژه‌ها از هم جدا کنید.

نویسنده: حسن
تاریخ: ۱۰:۱۴ ۱۳۹۱/۱۲/۲۹

شما در مطلبتون با ضدالگو شروع کردید و عنوان کردید که روش code behind یک سری مشکلاتی رو داره. سؤال من هم این بود که آیا صرفاً تعریف چند مژول جدید می‌تواند ضمانتی باشد بر رفع مشکل code behind یا اینکه با این مژول‌ها هم نهایتاً همان مشکل قبل پابرجا است یا می‌تواند پابرجا باشد.

ضمن اینکه تعریف شما از لایه دقیقاً چی هست؟ به نظر فقط تعریف یک اسمبلی در اینجا لایه نام گرفته.

نویسنده: آرمان فرقانی
تاریخ: ۱۱:۵۸ ۱۳۹۱/۱۲/۲۹

صحبت شما کاملاً صحیح است و صرفاً با مازولار کردن به معماری چند لایه نمی‌رسیم. اما نویسنده مقاله نیز چنین نگفته و در پایان مقاله بحث پایان ساختار چند لایه است و نه پایان پروژه. این قسمت اول این سری مقالات است و قطعاً در هنگام پیاده سازی کدهای هر لایه مباحثی مطرح خواهد شد تا تضمین مفهوم مورد نظر شما باشد.

نویسنده: میثم خوشبخت
تاریخ: ۱۳۹۱/۱۲/۲۹ ۱۲:۳۸

با تشکر از دوست عزیزم جناب آقای آرمان فرقانی با توضیحی که دادند.
یکی از دلایل این شیوه که نویسی امکان تست نویسی برای هر یک از لایه‌ها و همچنین استقلال لایه‌ها از هم دیگه هست که هر لایه بتونه بدون وجود لایه‌ی دیگه تست بشه. مازولار کردنه ممکنه مشکل UI Smart رو حل کنه و ممکنه حل نکنه. بستگی به شیوه که نویسی داره.

نویسنده: بهروز
تاریخ: ۱۳۹۱/۱۲/۲۹ ۱۳:۱۱

وقتی نظرات زیر مطلب شما رو میخونم میفهمم که نیاز به این سری آموزشی که دارید ارائه میدید چقدر زیاد احساس میشه فقط میخواستم بگم بر سر این مبحثی که دارید ارائه می‌دید اختلاف بین علما زیاد است! (حتی در عمل و در شرکت‌های نرم افزاری که تا به حال دیدم چه برسد در سطح آموزش...)
امیدوارم این حساسیت رو در نظر بگیرید و همه ما پس از مطالعه این سری آموزشی به فهم مشترک و یکسانی در مورد مفاهیم موجود برسیم
فکر می‌کنم وجود یک پروژه برای دست یافتن به این هدف هم ضروری باشد
باز هم تشکر

نویسنده: میثم خوشبخت
تاریخ: ۱۳۹۱/۱۲/۲۹ ۱۳:۵۹

من هم وقتی کار بر روی این معماری رو شروع کردم با مشکلات زیادی روبرو بودم و خیلی از مسائل برای من هم نامانوس و غیر قابل هضم بود. ولی بعد از اینکه چند پروژه نرم افزاری رو با این معماری پیاده سازی کردم فهم بیشتری نسبت به اون پیدا کردم و خیلی از مشکلات موجود رو با دقت بالا و با در نظر گرفتن تمامی الگوها رفع کردم. امیدوارم این حس مشترک بوجود بیاد. ولی دلیل اصلی ایجاد تکنولوژی‌ها و معماری‌های جدید اختلاف نظر بین علماست. این اختلاف نظر در اکثر موقع میتوانه مفید باشد.
ممnon دوست عزیز

نویسنده: مسعود مشهدی
تاریخ: ۱۳۹۲/۰۱/۰۴ ۱۸:۳۳

با سلام
بابت مطالبتون سپاسگزارم
همون طور که خودتون گفتید نظرات و شیوه‌های متفاوتی در نوع لایه بندی‌ها وجود داره.
در مقام مقایسه لایه بندی زیر چه وجه اشتراک و تفاوتی با لایه بندی شما داره.

Application.Web

Application.Manager

Application.DAL

Application.DTO

Application.Core

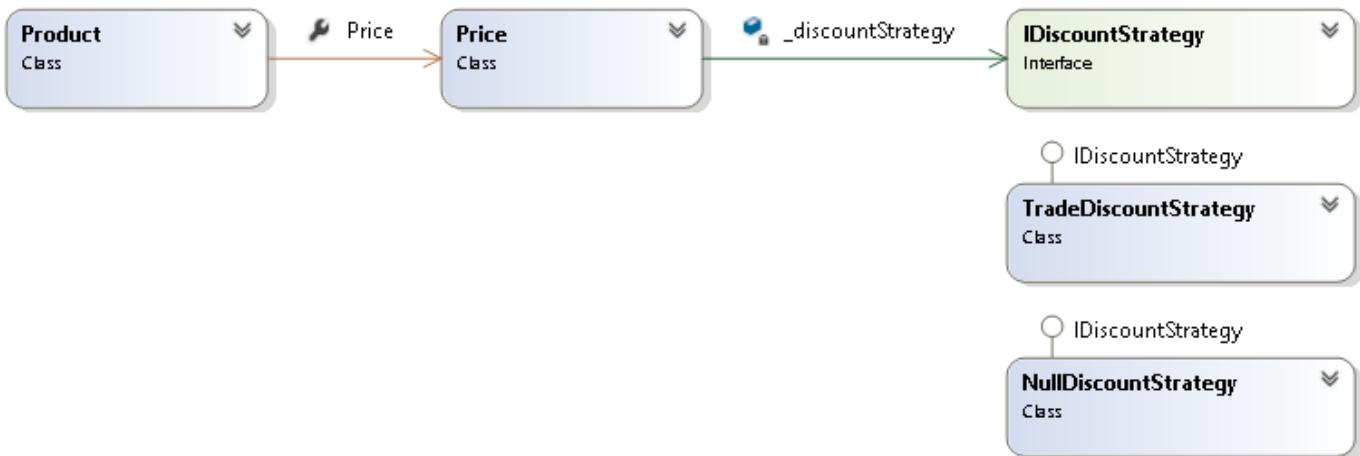
با تشکر

Business Layer یا Domain Model

پیاده سازی را از منطق تجاری یا Business Logic آغاز می کنیم. در روش کد نویسی Smart UI، منطق تجاری در قرار می گرفت اما در روش لایه بندی، منطق تجاری و روابط بین داده ها در Domain Model طراحی و پیاده سازی می شوند. در مطالب بعدی راجع به Domain Model و الگوهای پیاده سازی آن بیشتر صحبت خواهیم کرد اما بصورت خلاصه این لایه یک مدل مفهومی از سیستم می باشد که شامل تمامی موجودیت ها و روابط بین آنهاست.

الگوی Domain Model جهت سازماندهی پیچیدگی های موجود در منطق تجاری و روابط بین موجودیت ها طراحی شده است.

شكل زیر مدلی را نشان می دهد که می خواهیم آن را پیاده سازی نماییم. کلاس Product موجودیتی برای ارائه محصولات یک فروشگاه می باشد. کلاس Price جهت تشخیص قیمت محصول، میزان سود و تخفیف محصول و همچنین استراتژی های تخفیف با توجه به منطق تجاری سیستم می باشد. در این استراتژی همکاران تجاری از مشتریان عادی تقسیک شده اند.



را در پروژه SoCPatterns.Layered.Model به نام Domain Model پیاده سازی می کنیم. بنابراین به این پروژه یک Interface به نام IDiscountStrategy را با کد زیر اضافه نمایید:

```

public interface IDiscountStrategy
{
    decimal ApplyExtraDiscountsTo(decimal originalSalePrice);
}
  
```

علت این نوع نامگذاری Interface فوق، انطباق آن با الگوی Strategy Design Pattern می باشد که در مطالب بعدی در مورد این الگو بیشتر صحبت خواهیم کرد. استفاده از این الگو نیز به این دلیل بود که این الگو مختص الگوریتم هایی است که در زمان اجرا قابل انتخاب و تغییر خواهند بود.

توجه داشته باشید که معمولاً نام Design Pattern انتخاب شده برای پیاده سازی کلاس را بصورت پسوند در انتهای نام کلاس ذکر می‌کنند تا با یک نگاه، برنامه نویس بتواند الگوی مورد نظر را تشخیص دهد و مجبور به بررسی کد نباشد. البته به دلیل تشابه برخی از الگوهای امکان تشخیص الگو، در پاره ای از موارد وجود ندارد و یا به سختی امکان پذیر است.

الگوی Strategy یک الگوریتم را قادر می‌سازد تا در داخل یک کلاس کپسوله شود و در زمان اجرا به منظور تغییر رفتار شی، بین رفتارهای مختلف سوئیچ شود.

حال باید دو کلاس به منظور پیاده سازی روال تخفیف ایجاد کنیم. ابتدا کلاسی با نام TradeDiscountStrategy را با کد زیر به پروژه SoCPatterns.Layered.Model اضافه کنید:

```
public class TradeDiscountStrategy : IDiscountStrategy
{
    public decimal ApplyExtraDiscountsTo(decimal originalSalePrice)
    {
        return originalSalePrice * 0.95M;
    }
}
```

سپس با توجه به الگوی Null Object کلاسی با نام NullDiscountStrategy را با کد زیر به پروژه SoCPatterns.Layered.Model اضافه کنید:

```
public class NullDiscountStrategy : IDiscountStrategy
{
    public decimal ApplyExtraDiscountsTo(decimal originalSalePrice)
    {
        return originalSalePrice;
    }
}
```

از الگوی Null Object زمانی استفاده می‌شود که نمی‌خواهید و یا در برخی مواقع نمی‌توانید یک نمونه (Instance) معتبر را برای یک کلاس ایجاد نمایید و همچنین مایل نیستید که مقدار Null را برای یک نمونه از کلاس برگردانید. در مباحث بعدی با جزئیات بیشتری در مورد الگوها صحبت خواهم کرد.

با توجه به استراتژی‌های تخفیف کلاس Price را ایجاد کنید. کلاسی با نام Price را با کد زیر به پروژه SoCPatterns.Layered.Model اضافه کنید:

```
public class Price
{
    private IDiscountStrategy _discountStrategy = new NullDiscountStrategy();
    private decimal _rrp;
    private decimal _sellingPrice;
    public Price(decimal rrp, decimal sellingPrice)
    {
        _rrp = rrp;
        _sellingPrice = sellingPrice;
    }
    public void SetDiscountStrategyTo(IDiscountStrategy discountStrategy)
    {
        _discountStrategy = discountStrategy;
    }
    public decimal SellingPrice
    {
        get { return _discountStrategy.ApplyExtraDiscountsTo(_sellingPrice); }
    }
}
```

```

    }
    public decimal Rrp
    {
        get { return _rrp; }
    }
    public decimal Discount
    {
        get {
            if (Rrp > SellingPrice)
                return (Rrp - SellingPrice);
            else
                return 0;
        }
    }
    public decimal Savings
    {
        get{
            if (Rrp > SellingPrice)
                return 1 - (SellingPrice / Rrp);
            else
                return 0;
        }
    }
}

```

کلاس Price از نوعی Dependency Injection به نام SetDiscountStrategyTo استفاده نموده است که استراتژی تخفیف را برای کالا مشخص می‌نماید. نوع دیگری از Dependency Injection با نام Constructor Injection وجود دارد که در مباحث بعدی در مورد آن بیشتر صحبت خواهم کرد.

جهت تکمیل لایه Model ، کلاس Product را با کد زیر به پروژه SoCPatterns.Layered.Model اضافه کنید:

```

public class Product
{
    public int Id {get; set;}
    public string Name { get; set; }
    public Price Price { get; set; }
}

```

موجودیت‌های تجاری ایجاد شدن اما باید روشی اتخاذ نمایید تا لایه Model نسبت به منبع داده ای بصورت مستقل عمل نماید. به سرویسی نیاز دارید که به کلاینت‌ها اجازه بدهد تا با لایه مدل در ابطاباشند و محصولات مورد نظر خود را با توجه به تخفیف اعمال شده برای رابط کاربری برگردانند. برای اینکه کلاینت‌ها قادر باشند تا نوع تخفیف را مشخص نمایند، باید یک نوع شمارشی ایجاد کنید که به عنوان پارامتر ورودی متد سرویس استفاده شود. بنابراین نوع شمارشی CustomerType را با کد زیر به پروژه SoCPatterns.Layered.Model اضافه کنید:

```

public enum CustomerType
{
    Standard = 0,
    Trade = 1
}

```

برای اینکه تشخیص دهیم کدام یک از استراتژی‌های تخفیف باید بر روی قیمت محصول اعمال گردد، نیاز داریم کلاسی را ایجاد کنیم تا با توجه به CustomerType تخفیف مورد نظر را اعمال نماید. کلاسی با نام DiscountFactory را با کد زیر ایجاد نمایید:

```

public static class DiscountFactory
{
}

```

```
public static IDiscountStrategy GetDiscountStrategyFor
    (CustomerType customerType)
{
    switch (customerType)
    {
        case CustomerType.Trade:
            return new TradeDiscountStrategy();
        default:
            return new NullDiscountStrategy();
    }
}
```

در طراحی کلاس فوق از الگوی Factory استفاده شده است. این الگو یک کلاس را قادر می‌سازد تا با توجه به شرایط، یک شی معتبر را از یک کلاس ایجاد نماید. همانند الگوهای قبلی، در مورد این الگو نیز در مباحث بعدی بیشتر صحبت خواهم کرد.

لایهی سرویس با برقراری ارتباط با منبع داده ای، داده‌های مورد نیاز خود را بر می‌گرداند. برای این منظور از الگوی Repository استفاده می‌کنیم. از آنجایی که لایه Model باید مستقل از منبع داده ای عمل کند و نیازی به شناسایی نوع منبع داده ای ندارد، جهت پیاده سازی الگوی Repository از Interface استفاده می‌شود. یک Interface به نام IRepository را با کد زیر به پروژه SoCPatterns.Layered.Model اضافه کنید:

```
public interface IRepository
{
    IList<Product> FindAll();
}
```

الگوی Repository به عنوان یک مجموعه‌ی در حافظه (In-Memory Collection) یا انباره‌ی از موجودیت‌های تجاری عمل می‌کند که نسبت به زیر بنای ساختاری منبع داده ای کاملاً مستقل می‌باشد.

کلاس سرویس باید بتواند استراتژی تخفیف را بر روی مجموعه‌ی از محصولات اعمال نماید. برای این منظور باید یک Collection سفارشی ایجاد نماییم. اما من ترجیح می‌دهم از Extension Methods برای اعمال تخفیف بر روی محصولات استفاده کنم. بنابراین کلاسی به نام ProductListExtensionMethods را با کد زیر به پروژه SoCPatterns.Layered.Model اضافه کنید:

```
public static class ProductListExtensionMethods
{
    public static void Apply(this IList<Product> products,
                           IDiscountStrategy discountStrategy)
    {
        foreach (Product p in products)
        {
            p.Price.SetDiscountStrategyTo(discountStrategy);
        }
    }
}
```

الگوی Separated Interface تضمین می‌کند که کلاینت از پیاده سازی واقعی کاملاً نامطلع می‌باشد و می‌تواند برنامه نویس را به سمت Dependency Inversion و Abstraction بگذارد.

حال باید کلاس Service را ایجاد کنیم تا از طریق این کلاس، کلاینت با لایه Model در ارتباط باشد. کلاسی به نام ProductService را با کد زیر به پروژه SoCPatterns.Layered.Model اضافه کنید:

```

public class ProductService
{
    private IProductRepository _productRepository;
    public ProductService(IProductRepository productRepository)
    {
        _productRepository = productRepository;
    }
    public IList<Product> GetAllProductsFor(CustomerType customerType)
    {
        IDiscountStrategy discountStrategy =
            DiscountFactory.GetDiscountStrategyFor(customerType);
        IList<Product> products = _productRepository.FindAll();
        products.Apply(discountStrategy);
        return products;
    }
}

```

در اینجا کدنویسی منطق تجاری در Domain Model به پایان رسیده است. همانطور که گفته شد، لایه‌ی Business یا همان Model به هیچ منبع داده‌ای خاصی وابسته نیست و به جای پیاده سازی کدهای منبع داده‌ای، از Interface‌ها به منظور برقراری ارتباط با پایگاه داده استفاده شده است. پیاده سازی کدهای منبع داده‌ای را به لایه‌ی Repository واگذار نمودیم که در بخش‌های بعدی نحوه پیاده سازی آن را مشاهده خواهید کرد. این امر موجب می‌شود تا لایه Model درگیر پیچیدگی‌ها و کدنویسی‌های منبع داده‌ای نشود و بتواند به صورت مستقل و فارغ از بخش‌های مختلف برنامه تست شود. لایه بعدی که می‌خواهیم کدنویسی آن را آغاز کنیم، لایه‌ی Service می‌باشد.

در کدنویسی‌های فوق از الگوهای طراحی (Design Patterns) متعددی استفاده شده است که به صورت مختصر در مورد آنها صحبت کردم. اصلاً جای نگرانی نیست، چون در مباحثت بعدی به صورت مفصل در مورد آنها صحبت خواهیم کرد. در ضمن، ممکن است روال یادگیری و آموزش بسیار نامفهوم باشد که برای فهم بیشتر موضوع، باید کدها را بصورت کامل تست نموده و مثالهایی را پیاده سازی نمایید.

نظرات خوانندگان

نویسنده: سینا کردی
تاریخ: ۱۳۹۱/۱۲/۳۰ ۴:۱۰

سلام
ممnon از شما این بخش هم کامل و زیبا بود
ولی کمی فشرده بود
لطفا اگر ممکن هست در مورد معماRیها و الگوها و بهترین های آنها کمی توضیح دهید یا منبعی معرفی کنید تا این الگوها و معماRی برای ما بیشتر مفهوم بشو
من در این زمینه تازه کارم و از شما میخواهم که من رو راهنمایی کنید که چه مقدماتی در این زمینه ها نیاز دارم
باز هم ممنون.

نویسنده: علی
تاریخ: ۱۳۹۱/۱۲/۳۰ ۹:۱۲

در همین سایت مباحثت الگوهای طراحی و [Refactoring](#) مفید هستند.

و یا الگوهای طراحی Agile رو هم در [اینجا](#) می‌توانید پیگیری کنید.

نویسنده: میثم خوشبخت
تاریخ: ۱۳۹۱/۱۲/۳۰ ۱۱:۳۸

فشردگی این مباحثت بخارط این بود که میخواستم فعلا یک نمونه پروژه رو آموزش بودم تا یک شمای کلی از کاری که می‌خواهیم انجام بدیم رو ببینید. در مباحثت بعدی این مباحثت رو بازتر می‌کنم. خود من برای مطالعه و جمع بندی این مباحثت منابع زیادی رو مطالعه کردم. واقعا برای بعضی مباحثت نمیشه به یک منبع اکتفا کرد.

نویسنده: محسن.د
تاریخ: ۱۳۹۱/۱۲/۳۰ ۱۷:۱

بسیار عالی

آیا فراخوانی مستقیم تابع `SetDiscountStrategyTo` کلاس `Price` در تابع `Apply` از نظر کپسوله سازی مورد اشکال نیست ؟ بهتر نیست که برای خود کلاس `Product` یک تابع پیاده سازی کنیم که در درون خودش تابع `Price.SetDiscountStrategyTo` فراخوانی کند و به این شکل کلاس های بیرونی رو از تغییرات درونی کلاس `Product` مستقل کنیم ؟

نویسنده: میثم خوشبخت
تاریخ: ۱۳۹۱/۱۲/۳۰ ۱۸:۱

دوست عزیزم، متده `Apply` یک `Extension Method` برای `IList<Product>` است. اگر این متده تعریف نمی‌شد شما باید در کلاس `foreach` رو قرار می‌دادید. البته با این حال در قسمت هایی از طراحی کلاسها که الگوهای طراحی را زیر سوال نمی‌برد و تست پذیری را دچار مشکل نمی‌کند، طراحی سلیقه ای است. مقاله من هم آیهی نازل شده نیست که دستخوش تغییرات نشود. شما می‌توانید با سلیقه و دید فنی خود تغییرات مورد نظر رو اعمال کنید. ولی اگر نظر من را بخواهید این طراحی مناسب‌تر است.

نویسنده: رضا عرب
تاریخ: ۱۳۹۲/۰۱/۰۹ ۱۴:۴۵

خسته نباشید، واقعاً ممنونم آقای خوشبخت، لطفاً به نگارش این دست مطالب مرتبط با طراحی ادامه دهید، زمینه بکریه که کمتر عملی به آن پرداخته شده و این نوع نگارش شما فراتر از یک معرفیه که واقعاً جای تشکر دارد.

نویسنده: f.tahan36
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۷:۱۰

با سلام

تفاوت design factory با factory چیست؟ (با مثال کد)
و virtual کردن یک تابع معمولی با virtual کردن تابع سازنده چه تفاوتی دارد؟

با تشکر

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۳۰ ۰:۴۰

از همون رندهایی هستی که تمرين کلاسیت رو آورده اینجا؟! :)

Service Layer

نقش لایه‌ی سرویس این است که به عنوان یک مدخل ورودی به برنامه کاربردی عمل کند. در برخی مواقع این لایه را به عنوان لایه‌ی Facade نیز می‌شناسند. این لایه، داده‌ها را در قالب یک نوع داده‌ای قوی (Strongly Typed) به نام View Model به نام View Model می‌فرماید. کلاس Strongly Typed View Model یک محسوب می‌شود که نماهای خاصی از داده‌ها را که متفاوت از دید یا نمای تجاری آن است، بصورت بهینه ارائه می‌نماید. در مورد الگوی View Model در مباحث بعدی بیشتر صحبت خواهیم کرد.

الگوی Facade یک Interface را به منظور کنترل دسترسی به مجموعه‌ای از Interface‌ها و زیر سیستم‌های پیچیده ارائه می‌کند. در مباحث بعدی در مورد آن بیشتر صحبت خواهیم کرد.

کلاسی با نام ProductViewModel را با کد زیر به پروژه SoCPatterns.Layered.Service اضافه کنید:

```
public class ProductViewModel
{
    public int ProductId { get; set; }
    public string Name { get; set; }
    public string Rrp { get; set; }
    public string SellingPrice { get; set; }
    public string Discount { get; set; }
    public string Savings { get; set; }
}
```

برای اینکه کلاینت با لایه‌ی سرویس در تعامل باشد باید از الگوی Request/Response Message استفاده کنیم. بخش توسط کلاینت تغذیه می‌شود و پارامترهای مورد نیاز را فراهم می‌کند. کلاسی با نام ProductListRequest را با کد زیر به پروژه SoCPatterns.Layered.Service اضافه کنید:

```
using SoCPatterns.Layered.Model;

namespace SoCPatterns.Layered.Service
{
    public class ProductListRequest
    {
        public CustomerType CustomerType { get; set; }
    }
}
```

در شی Response نیز بررسی می‌کنیم که درخواست به درستی انجام شده باشد، داده‌های مورد نیاز را برای کلاینت فراهم می‌کنیم و همچنین در صورت عدم اجرای صحیح درخواست، پیام مناسب را به کلاینت ارسال می‌نماییم. کلاسی با نام ProductListResponse را با کد زیر به پروژه SoCPatterns.Layered.Service اضافه کنید:

```
public class ProductListResponse
{
    public bool Success { get; set; }
```

```
public string Message { get; set; }
public IList<ProductViewModel> Products { get; set; }
}
```

به منظور تبدیل موجودیت Product به ProductViewModel، یک برای تبدیل یک Product و دیگری برای تبدیل لیستی از Product. شما می‌توانید این دو متده را به کلاس Product موجود در Domain Model اضافه نمایید، اما این متدها نیاز واقعی منطق تجاری نمی‌باشند. بنابراین بهترین انتخاب، استفاده از Extension Method ها می‌باشد که باید برای کلاس Product و در لایه‌ی سرویس ایجاد نمایید. کلاسی با نام ProductMapperExtensionMethods را با کد زیر به پروژه SoCPatterns.Layered.Service اضافه کنید:

```
public static class ProductMapperExtensionMethods
{
    public static ProductViewModel ConvertToProductViewModel(this Model.Product product)
    {
        ProductViewModel productViewModel = new ProductViewModel();
        productViewModel.ProductId = product.Id;
        productViewModel.Name = product.Name;
        productViewModel.RRP = String.Format("{0:C}", product.Price.RRP);
        productViewModel.SellingPrice = String.Format("{0:C}", product.Price.SellingPrice);
        if (product.Price.Discount > 0)
            productViewModel.Discount = String.Format("{0:C}", product.Price.Discount);
        if (product.Price.Savings < 1 && product.Price.Savings > 0)
            productViewModel.Savings = product.Price.Savings.ToString("#%");
        return productViewModel;
    }
    public static IList<ProductViewModel> ConvertToProductListViewModel(
        this IList<Model.Product> products)
    {
        IList<ProductViewModel> productViewModels = new List<ProductViewModel>();
        foreach(Model.Product p in products)
        {
            productViewModels.Add(p.ConvertToProductViewModel());
        }
        return productViewModels;
    }
}
```

حال کلاس ProductService را جهت تعامل با کلاس سرویس موجود در Domain Model و به منظور برگرداندن لیستی از محصولات و تبدیل آن به لیستی از ProductViewModel، ایجاد نماییم. کلاسی با نام ProductService را با کد زیر به پروژه SoCPatterns.Layered.Service اضافه کنید:

```
public class ProductService
{
    private Model.ProductService _ productService;
    public ProductService(Model.ProductService ProductService)
    {
        _ productService = ProductService;
    }
    public ProductListResponse GetAllProductsFor(
        ProductListRequest productListRequest)
    {
        ProductListResponse productListResponse = new ProductListResponse();
        try
        {
            IList<Model.Product> productEntities =
                _ productService.GetAllProductsFor(productListRequest.CustomerType);
            productListResponse.Products = productEntities.ConvertToProductListViewModel();
            productListResponse.Success = true;
        }
        catch (Exception ex)
        {
            // Log the exception...
            productListResponse.Success = false;
            // Return a friendly error message
        }
    }
}
```

```
        productListResponse.Message = ex.Message;
    }
    return productListResponse;
}
}
```

کلاس Service تمامی خطاهای را دریافت نموده و پس از مدیریت خطا، پیغامی مناسب را به کلاینت ارسال می‌کند. همچنین این لایه محل مناسبی برای Log کردن خطاهای می‌باشد. در اینجا کد نویسی لایه سرویس به پایان رسید و در ادامه به کدنویسی Data Layer می‌پردازیم.

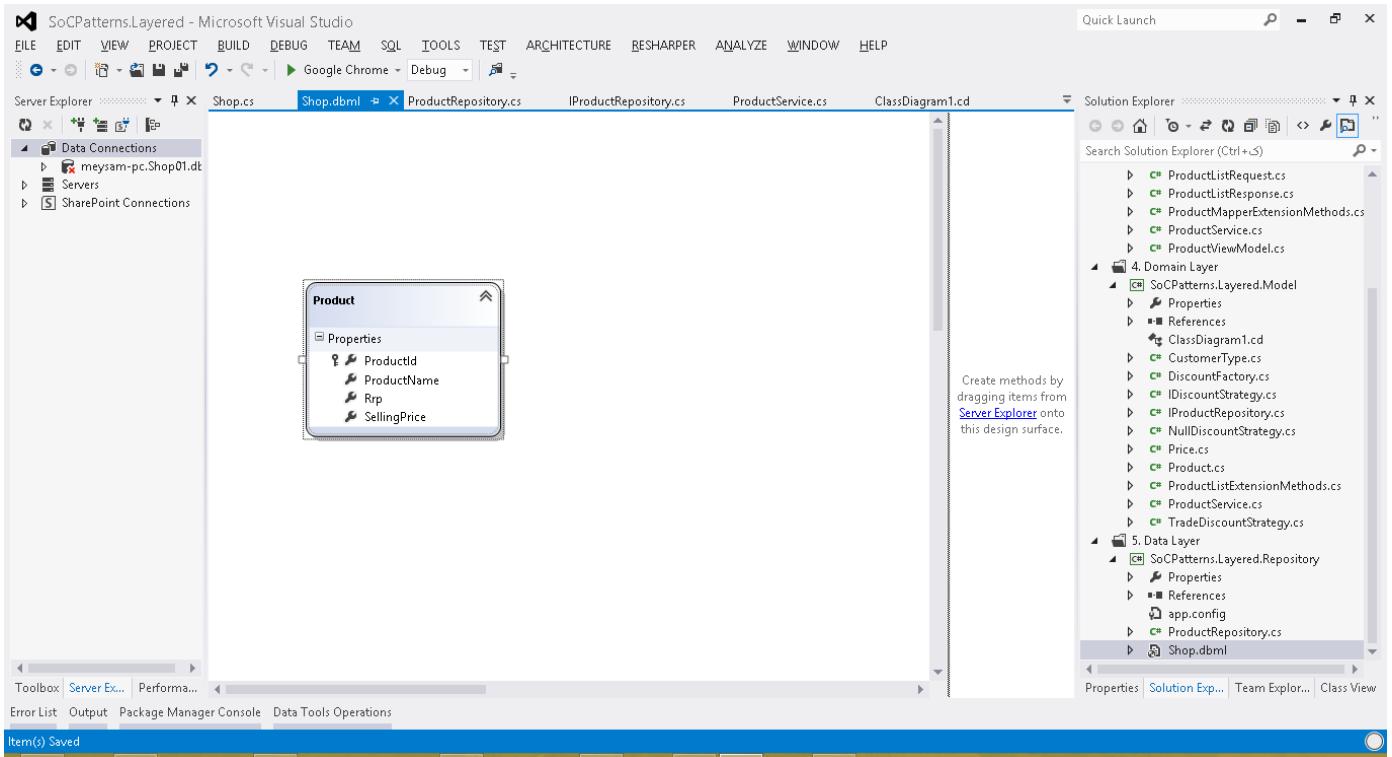
Data Layer

برای اذکاره سازی، مخصوصاً لات، یک بانک اطلاعات، با نام Shop01 اتحاد کنید که شاما، حدول، به نام Product با ساختار زیر باشد:

Column Name	Data Type	Allow Nulls
ProductId	int	<input type="checkbox"/>
ProductName	nvarchar(50)	<input type="checkbox"/>
Rrp	smallmoney	<input type="checkbox"/>
SellingPrice	smallmoney	<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>

برای اینکه کدهای بانک اطلاعاتی را سریعتر تولید کنیم از روش Linq to SQL در Data Layer استفاده می‌کنم. برای این منظور یک Data Context برای Linq to SQL به این لایه اضافه می‌کنیم. بر روی پروژه روزه SoCPatterns.Layered.Repository کلیک راست نمایید و گزینه Add > New Item را انتخاب کنید. در پنجره ظاهر شده و از سمت چپ گزینه Data و سپس از سمت راست گزینه Linq to SQL Classes را انتخاب نموده و نام آن را Shop.dbml تعیین نمایید.

از طریق پنجره Server Explorer به پایگاه داده مورد نظر متصل شوید و با عمل Drag & Drop جدول Product را به بخش Design کشیده و رها نمایید.



اگر به یاد داشته باشید، در لایه Model برای برقراری ارتباط با پایگاه داده از یک Interface به نام `IProductRepository` استفاده نمودیم. حال باید این Interface را پیاده سازی نماییم. کلاسی با نام `ProductRepository` را با کد زیر به پروژه SoCPatterns.Layered.Repository اضافه کنید:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using SoCPatterns.Layered.Model;

namespace SoCPatterns.Layered.Repository
{
    public class ProductRepository : IProductRepository
    {
        public IList<Model.Product> FindAll()
        {
            var products = from p in new ShopDataContext().Products
                           select new Model.Product
                           {
                               Id = p.ProductId,
                               Name = p.ProductName,
                               Price = new Model.Price(p.Rrp, p.SellingPrice)
                           };
            return products.ToList();
        }
    }
}
```

در متod `FindAll` ، با استفاده از دستورات Linq to SQL ، لیست تمامی محصولات را برگرداندیم. کدنویسی لایه Data هم به پایان رسید و در ادامه به کدنویسی لایه Presentation و UI می پردازیم.

Presentation Layer

به منظور جداسازی منطق نمایش (Presentation) از رابط کاربری (User Interface) ، از الگوی Model View Presenter استفاده می کنیم که در مباحث بعدی با جزئیات بیشتری در مورد آن صحبت خواهیم کرد. یک Interface با نام IPresenter اضافه کنید:

```
using SoCPatterns.Layered.Service;  
  
public interface IPresenter  
{  
    void Display(IList<ProductViewModel> Products);  
    Model.CustomerType CustomerType { get; }  
    string ErrorMessage { set; }  
}
```

این Interface توسط Web های ASP.NET و یا Win Form ها باید پیاده سازی شوند. کار با Interface ها موجب می شود تا تست View ها به راحتی انجام شوند. کلاسی با نام ProductListPresenter را با کد زیر به پروژه SoCPatterns.Layered.Presentation اضافه کنید:

```
using SoCPatterns.Layered.Service;  
  
namespace SoCPatterns.Layered.Presentation  
{  
    public class ProductListPresenter  
    {  
        private IPresenter _presenter;  
        private Service.ProductService _productService;  
        public ProductListPresenter(IPresenter presenter,  
                                    Service.ProductService productService)  
        {  
            _presenter = presenter;  
            _productService = productService;  
        }  
        public void Display()  
        {  
            ProductListRequest productListRequest = new ProductListRequest();  
            productListRequest.CustomerType = _presenter.CustomerType;  
            ProductListResponse productListResponse =  
                _productService.GetAllProductsFor(productListRequest);  
            if (productListResponse.Success)  
            {  
                _presenter.Display(productListResponse.Products);  
            }  
            else  
            {  
                _presenter.ErrorMessage = productListResponse.Message;  
            }  
        }  
    }  
}
```

کلاس Presenter وظیفه‌ی واکشی داده‌ها، مدیریت رویدادها و بروزرسانی UI را دارد. در اینجا کدنویسی لایه‌ی Presentation به پایان رسیده است. از مزایای وجود لایه‌ی Presentation این است که تست نویسی مربوط به نمایش داده‌ها و تعامل بین کاربر و سیستم به سهولت انجام می‌شود بدون آنکه نگران دشواری Unit Test Web Form نویسی داشته باشد. حال می‌توانید کد نویسی مربوط به UI را انجام دهید که در ادامه به کد نویسی در Win Forms و Web Forms خواهیم پرداخت.

نظرات خوانندگان

نویسنده: محسن

تاریخ: ۱۳۹۲/۰۱/۰۲ ۱۸:۲۹

ممnon از زحمات شما.

چند سؤال و نظر:

- با تعریف الگوی مخزن به چه مزیتی دست پیدا کردید؟ برای مثال آیا هدف این است که کدهای پیاده سازی آن، با توجه به وجود اینترفیس تعریف شده، شاید روزی با مثلا NHibernate تعویض شود؟ در عمل متاسفانه حتی پیاده سازی LINQ اینها هم متفاوت است و من تابحال در عمل ندیدم که ORM یک پروژه بزرگ رو عوض کنند. یعنی تا آخر و تا روزی که پروژه زنده است با همان انتخاب اول سر می کنند. یعنی شاید بهتر باشه قسمت مخزن و همچنین سرویس یکی بشن.
- چرا لایه سرویس تعریف شده از یک یا چند اینترفیس مشتق نمی شود؟ اینطوری تهیه تست برای اون ساده تر میشه. همچنین پیاده سازی ها هم وابسته به یک کلاس خاص نمی شن چون از اینترفیس دارن استفاده می کنند.
- این اشیاء Request و Response هم در عمل به نظر نوعی ViewModel هستند. درسته؟ اگر اینطوره بهتر یک مفهوم کلی دنبال بشه تا سردرگمی ها رو کمتر کنه.

یک سری نکته جانبی هم هست که می تونه برای تکمیل بحث جالب باشه:

- مثلا الگوی Context per request بجای نوشتن new ShopDataContext بهتر استفاده بشه تا برنامه در طی یک تراکنش و اتصال کار کنه.
- در مورد try/catch و استفاده از اون بحث زیاد هست. خیلی ها توصیه می کنن که یا اصلا استفاده نکنید یا استفاده از اون ها رو به بالاترین لایه برنامه موکول کنید تا این وسط کرش یک قسمت و بروز استثناء در اون، از ادامه انتشار صدمه به قسمت های بعدی جلوگیری کنه.

نویسنده: میثم خوشیخت

تاریخ: ۱۳۹۲/۰۱/۰۲ ۲۳:۳۵

محسن عزیز. از شما ممنونم که به نکته های ظرفی اشاره کردید.

در سری مقالات اولیه فقط دارم یک دید کلی به کسایی میدم که تازه دارن با این مفاهیم آشنا میشن. این پروژه اولیه دستخوش تغییرات زیادی میشه. در واقع محصول نهایی این مجموعه مقالات بر پایه همین نوع لایه بندی ولی بادید و طراحی مناسب تر خواهد بود.

در مورد ORM هم من با چند Application سروکار داشتم که در روال توسعه بخش های جدید رو بنا به دلایلی با DB متفاوتی توسعه داده اند. غیر از این موضوع، حتی بخش هایی از مدل، سرویس و یا مخزن رو در پروژه های دیگری استفاده کرده اند. همچنین برخی از نکات مربوط به تفکیک لایه ها به منظور تست پذیری راحت تر رو هم در نظر بگیرید.

در مورد اشیا Request و Response هم باید خدمت ان عرض کنم که برای درخواست و پاسخ به درخواست استفاده می شوند که چون پروژه ای که مثال زدم کوچک بوده ممکنه کاملا درکش نکرده باشد. ما کلاس های Request و Response متعددی در پروژه داریم که ممکنه خیلی از اونها فقط از یک ViewModel استفاده کنن ولی پارامتر های ارسالی یا دریافتی آنها متفاوت باشد.

در مورد try...catch هم من با شما کاملا موافقم. به دلیل هزینه ای که دارد باید در آخرین سطح قرار بگیرد. در این مورد ما میتوانیم اونو به Presentation و یا در MVC به Controller منتقل کنیم.

در مورد DbContext هم هنوز الگویی رو معرفی نکردم. در واقع هنوز وارد جزئیات لایه Data نشدم. در مورد اون اگه اجازه بدی بعدا صحبت میکنم.

نویسنده: ایلیا

تاریخ: ۱۳۹۲/۰۱/۰۳ ۰:۴۳

آقای خوشیخت خداقوت.

مرسى از مطالب خوبیتون.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۰۳ ۰:۴۸

لطفا برای اینکه نظرات حالت فنی تر و غنای بیشتری پیدا کنند، از ارسال پیام‌های تشکر خودداری کنید. برای ابراز احساسات و همچنین تشکر، لطفا از گزینه رای دادن به هر مطلب که ذیل آن قرار دارد استفاده کنید.
این مطلب تا این لحظه 76 بار دیده شده، اما فقط 4 رای دارد. لطفا برای ابراز تشکر، امتیاز بدهید. ممنون.

نویسنده: محسن
تاریخ: ۱۳۹۲/۰۱/۰۳ ۱:۰

- من در عمل تفاوتی بین لایه مخزن و سرویس شما مشاهده نمی‌کنم. یعنی لایه مخزن داره GetAll می‌کنه، بعد لایه سرویس هم داره همون رو به یک شکل دیگری بر می‌گردونه. این تکرار کد نیست؟ این دو یکی نیستند؟

عموما در منابع لایه مخزن رو به صورت روکشی برای دستورات مثل LINQ to SQL یا EF معرفی می‌کنند. فرضشون هم این است که این روش ما رو از تماس مستقیم با ORM بر حذر می‌داره (شاید فکر می‌کنند ایدز می‌گیرند اگر مستقیم کار کنند!). ولی عرض کردم این روکش در واقعیت فقط شاید با EF یا L2S قابل تعویض باشه نه باORM‌های دیگر با روش‌های مختلف و بیشتر یک تصور واهم هست که جنبه عملی نداره. بیشتر تئوری هست بدون پایه تجربه دنیای واقعی. ضمن اینکه این روکش باعث می‌شده نتوانید از خیلی از امکانات ORM مورد استفاده درست درست استفاده کنید. مثلا ترکیب کوئری‌ها یا روش‌های به تاخیر افتاده و امثال این.

- پس در عمل شما Request ViewModel و Response ViewModel تعریف کردید.

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۲/۰۱/۰۳ ۱۲:۲۷

سپاس از سری مطالبی که منتشر می‌کنید.
- پیشنهادی که من دارم اینه که لایه‌ی Repository حذف شود، همانطور که در مطالب قبلی ذکر شده DbSet در Entity Framework همان پیاده سازی الگوی مخزن هست و ایجاد Repository جدید روی آن یک Abstraction اضافه هست. در نتیجه اگر Repository حذف شود همه‌ی منطق‌ها مانند GetBlaBla به Service منتقل می‌شود.
- یک پیشنهاد دیگر اینکه استفاده از کلمه‌ی New در Presentation Layer را به حداقل رساند و همه جا نیاز مندی‌ها را به صورت وابستگی به کلاس‌های استفاده کننده تزریق شود تا در زمان نوشتن تست‌ها همه‌ی اجزاء قابل تعویض با Mock objects باشند.

نویسنده: افشنین
تاریخ: ۱۳۹۲/۰۱/۰۶ ۱۱:۱۵

لطفا دمو یا سورس برنامه رو هم قرار بدید که یادگیری و آموزش سریعتر انجام بشه.

ممnon

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۱/۱۰ ۰:۱۱

با سلام از کار بزرگی که دارین می‌کنین سپاس
یک سوال؟
جای الگوی Unit Of Work در این پروژه کجا می‌شه؟

در این [پست](#) جناب آقای مهندس نصیری در لایه سرویس الگوی واحد کار را پیاده کرده اند، با توجه به وجود الگوی Repository

در پژوهش شما ممنون میشم شرح بیشتری بدین که جایگاه پیاده سازی الگو واحد کار با توجه به مزایایی که دارد در کدام لایه است؟

نويسنده: رام تاریخ: ۰۱/۱۶/۱۳۹۲، ۲۹:۵

محسن جان، چیزی که من از این الگو در مورد واکشی و نمایش داده‌ها برداشت می‌کنم اینه:

کلاس‌های لایه سرویس آبجکت مدل مربوطه را پر میکنند و به بالا (لایه سرویس) پاس میدند.

بعد

در لایه سرویس نمونه‌ی مدل مربوطه به وی‌اومدل متناظر باهاش تبدیل می‌شود و به لایه بالاتر فرستاده می‌شود.

- کار در "لایه مخزن" روی "مدل ها" انجام میگیره
- کار در "لایه سرویس" روی "وی-مدل ها" انجام میشے

نتیجه: لایه سرویس هدف دیگری را نسبت به لایه مخزن دنبال میکند و این هدف آنقدر بزرگ و مهم هست که برایش یه لایه مجزا در نظر گرفته بشه

نويسنده: رام تاریخ: ۰۱/۱۶/۱۳۹۲

شاهین جان، من با حذف لایه مخزن مخالف هستم. زیرا:

حالا این اطلاعات ممکن است از پایگاه داده یا جاهای دیگه جمع آوری شوند (و الزاماً توسط EF قابل دسترسی و ارائه نباشند)

همچنین گاهی نیاز هست که بر مبنای چند متده است که EF به ما میرسونه (مثلاً چند SP) یک متده کلی‌تر را تعریف کنیم (چند فراخوانی را در یک متده مثلاً متده X در لایه مخزن انجام دهیم) و در لایه بالاتر آن متده را صدا بزنیم (بجای نوشتن و تکرار پاپی همه کدهای نوشته شده در متده X)

علاوه بر این در لایه مخزن میشه چند ORM را هم کنار هم دید (نه فقط EF) که همونطور که آقای خوشبخت در کامنت‌ها نوشتند گاهه، نیاز میشه.

دیباپ

من وجود لایه مخزن را ضریب، صد و نه.

(فراموش نکنیم که هدف از این آموزش تعریف یک الگوی معماری مناسب برای پروژه‌های بزرگ هست و الا بدون خیلی از اینها هم میشه بنامه ساخت. همومنظور، که اکثراً بدون این ساختارها و خیلی ساده‌تر میسازند)

نویسنده: محسن تاریخ: ۱۳۹۲/۰۱/۱۶

- بحث آقای شاهین و من در مورد مثال عینی بود که زده شد. در مورد کار با ORM که کدھاش دقیقاً ارائه شده، این روش قابل نقد و رد است.

شما الان اومدی یک بحث انتزاعی کلی رو شروع کردید. بله. اگر ORM رو کنار بگذارید مثلاً می‌رسید به ADO.NET (یک نمونه که خیلی‌ها در این سایت حداقل یکبار باهش کار کردن). این افراد پیش از اینکه این مباحث مطرح باشن برای خودشون لایه DAL داشتند و تمام جزئیات ADO.NET رو کپسوله کرده بودن در اون. حالا با امدنORM‌ها این لایه DAL کنار رفته چون خود هست که کپسوله کننده ADO.NET است. همین‌ها هم یک لایه دیگر داشتند به نام BLL که از لایه DAL استفاده می‌کرد برای پیاده سازی منطق تجاری برنامه. این لایه الان اسمش شده لایه سرویس.

یعنی تمام مواردی رو که عنوان کردید در مورد ADO.NET صدق می‌کنه. یکی اسمش رو می‌ذاره DAL شما اسمش رو گذشتید Repository. ولی این مباحث ربطی به یک ORM تمام عیار که کپسوله کننده ADO.NET است ندارد.

- ترکیب چند SP در لایه مخزن انجام نمی‌شده. چیزی رو که عنوان کردید یعنی پیاده سازی منطق تجاری و این مورد باید در لایه سرویس باشه. اگر از ADO.NET استفاده می‌کنید، می‌توانیم با استفاده از DAL جزئیات دسترسی به SP رو مخفی و ساده‌تر کنیم با کدی یک دست‌تر در تمام برنامه. اگر از EF استفاده می‌کنیم، باز همین ساده سازی در طی فراخوانی فقط یک متده انجام شده. بنابراین بهتر است وضعیت و سطح لایه‌ای رو که داریم باهش کار می‌کنیم خوب بررسی و درک کنیم.

- می‌توانید در عمل در بین پروژه‌های سورس باز و معتبر موجود فقط یک نمونه رو به من ارائه بدید که در اون از 2 مورد ORM مختلف همزمان استفاده شده باشه؟ این مورد یعنی سوئی مدیریت. یعنی پراکندگی و انجام کاری بسیار مشکل مثلاً یک نمونه: ORM‌ها لایه‌ای دارند به نام سطح اول کش که مثلاً در EF اسمش هست Tracking API. این لایه فقط در حین کار با Context همون کار می‌کنند. اگر دو مورد رو با هم مخلوط کنید، قابل استفاده نیست، ترکیب پذیر نیستند. از این دست باز هم هست مثلاً در مورد نحوه تولید پروکسی‌هایی که برای lazy loading تولید می‌کنند و خیلی از مسایل دیگری از این دست. ضمن اینکه مدیریت چند Context فقط در یک لایه خودش یعنی نقض اصل تک مسئولیتی کلاس‌ها.

نویسنده: محسن
تاریخ: ۹:۱۵ ۱۳۹۲/۰۱/۱۶

سعی نکنید انتزاعی بحث کنید. چون در این حالت این حرف می‌توانه درست باشه یا حتی نباشه. اگر از ADO.NET استفاده می‌کنید، درسته. اگر از EF استفاده می‌کنید غلط هست. لازم هست منطق کار با ADO.NET رو یک سطح کپسوله کنیم. چون از تکرار کد جلوگیری می‌کنیم و نهایتاً به یک کد یک دست خواهیم رسید. لازم نیست اعمال یک ORM رو در لایه‌ای به نام مخزن کپسوله کنیم، چون خودش کپسوله سازی ADO.NET رو به بهترین نحوی انجام داده. برای نمونه در همین مثال عینی بالا به هیچ مزیتی نرسیدیم. فقط یک تکرار کد است. فقط بازی با کدها است.

نویسنده: رام
تاریخ: ۱۶:۴۶ ۱۳۹۲/۰۱/۱۶

من منظور شما را خوب متوجه نمی‌شم ولی حرفام یه بحث انتزاعی نیست چون پروژه عملی زیر دستم دارم که توی اون هم با پر کردن ViewModel کار می‌کنم.

مشکل از اینجا شروع می‌شده که شما فکر می‌کنید همیشه مدل ای که در EF ساختید را باید بدون تغییر در ساختارش به پوسته برنامه برسونید و از پوسته هم دقیقاً نمونه ای از همون را بگیرید و به لایه‌های پایین بفرستید ولی یکی از مهمترین کارهای View Model اینه که این قانون را از این سفتی و سختی در بیاره چون خیلی موقع هست که شما در پوسته برنامه به شکل دیگه ای از داده‌ها (متفاوت با اونچه در Model تعریف کردید و EF باهش کار می‌کنن) نیاز دارید. مثلاً فیلد تاریخ از نوع DateTime در Model و نوع String در پوسته و یا حتی اضافه و کم کردن فیلد‌های یک Model و ایجاد ساختارهای متفاوتی از اون برای عملیات‌های Select, Delete و Update. لذا لایه سرویس قرار نیست فقط همون کار لایه مخزن را تکرار کنه (به قول شما GetAll). بلکه در زمان لزوم تغییرات لازم که نام بردم را هم روش اعمال می‌کنم (که به نظر من آقای خوشبخت هم به خوبی از کلمه Convert در لایه سرویس استفاده کردن).

اما بحث اینکه ما در لایه مخزن روی EF یک سطح کپسوله می‌سازیم جای گفتگو داره هر چند من در اون مورد هم با وجود لایه مخزن بیشتر موافقم تا گفتگوی مستقیم لایه سرویس با چیزی مثل EF

نتیجه: فرقی نمی‌کنه شما از Asp.Net استفاده می‌کنید یا هر ORM مورد نظرتون. کلاس‌های مدل باید در ارتباط با لایه بالاتر خودشون به ویو مدل تبدیل بشند و در این الگو این کار در لایه سرویس انجام می‌شده.

- پیاده سازی الگوی مخزن در عمل (بر اساس بحث فعلی که در مورد کار با ORM‌ها است) به صورت کپسوله سازی ORM در همه جا مطرح میشود و اینکار اساساً اشتباه است. چون هم شما را محروم می‌کنید از قابلیت‌های پیشرفته ORM و هم ارزش افزوده‌ای را به همراه نداره. دست آخر می‌بینید در لایه مخزن GetAll دارید در لایه سرویس هم GetAll دارید. این مساله هیچ مزیتی نداره. یک زمانی در ADO.NET برای GetAll کردن باید کلی کد شبیه به کدهای یک ORM نوشته می‌شد. خود ORM الان اومنده این‌ها را کپسوله کرده و لایه‌ای هست روی اون. اینکه ما مجدداً یک پوسته روی این بکشیم حاصلی نداره بجز تکرار کد. عده‌ای عنوان می‌کنند که حاصل اینکار امکان تعویض ORM را ممکن می‌کنند ولی این‌ها هم بعد از یک مدت تجربه باORM‌های مختلف به این نتیجه می‌رسند که ای بابا! حتی پیاده سازی LINQ این‌ها یکی نیست چه بررسه به قابلیت‌های پیشرفته‌ای که در یکی هست در دو تای دیگر نیست (واقع بینی، بجای بحث تئوری محض).

- اینکه این تبدیلات (پر کردن ViewModel از روی مدل) هم می‌توانه و بهتره که (نه الزاماً) در لایه سرویس انجام بشود، نتیجه مناسبی هست.

در نهایت نوبت به طراحی و کدنویسی UI می‌رسد تا بتوانیم محصولات را به کاربر نمایش دهیم. اما قبل از شروع باید موضوعی را یادآوری کنم. اگر به یاد داشته باشید، در کلاس `ProductService` موجود در لایه `Domain`، از طریق یکی از روش‌های الگوی `Dependency Injection` به نام `Constructor Injection`، فیلدی از نوع `IProductRepository` را مقداردهی نمودیم. حال زمانی که بخواهیم نمونه ای را از `ProductService` ایجاد نماییم، باید به عنوان پارامتر ورودی سازنده، شی ایجاد شده از جنس کلاس `ProductRepository` موجود در لایه `Repository` را به آن ارسال نماییم. اما از آنجایی که می‌خواهیم تفکیک پذیری لایه‌ها از بین نرود و UI بسته به نیاز خود، نمونه مورد نیاز را ایجاد نموده و به این کلاس ارسال کند، از ابزارهایی برای این منظور استفاده می‌کنیم. یکی از این ابزارها `StructureMap` می‌باشد که یک `Inversion of Control Container` یا به اختصار `IoC Container` نامیده می‌شود. با `Inversion of Control` در مباحث بعدی بیشتر آشنا خواهید شد. `StructureMap` ابزاری است که در زمان اجرا، پارامترهای ورودی سازنده کلاس‌هایی را که از الگوی `Dependency Injection` استفاده نموده اند و قطعاً پارامتر ورودی آنها از جنس یک `Interface` می‌باشد را، با ایجاد شی مناسب مقداردهی می‌نماید.

به منظور استفاده از `StructureMap` در 2012 `Visual Studio` باید بر روی پروژه UI خود کلیک راست نموده و گزینه‌ی `Manage NuGet Packages` را انتخاب نمایید. در پنجره ظاهر شده و از سمت چپ گزینه‌ی `Online` و سپس در کادر جستجوی سمت راست و بالای پنجره واژه‌ی `structuremap` را جستجو کنید. توجه داشته باشید که باید به اینترنت متصل باشید تا بتوانید `Package` مورد نظر را نصب نمایید. پس از پایان عمل جستجو، در کادر میانی `structuremap` ظاهر می‌شود که می‌توانید با انتخاب آن و فشردن کلید `Install` آن را بر روی پروژه نصب نمایید.

جهت آشنایی بیشتر با `NuGet` و نصب آن در سایر نسخه‌های `Visual Studio` می‌توانید به لینک‌های زیر رجوع کنید:

1. آشنایی با [NuGet قسمت اول](#)

2. آشنایی با [NuGet قسمت دوم](#)

[Installing .3](#)
[NuGet](#)

کلاسی با نام `BootStrapper` را با کد زیر به پروژه UI خود اضافه کنید:

```

using StructureMap;
using StructureMap.Configuration.DSL;
using SoCPatterns.Layered.Repository;
using SoCPatterns.Layered.Model;

namespace SoCPatterns.Layered.WebUI
{
  public class BootStrapper
  {
    public static void ConfigureStructureMap()
    {
      ObjectFactory.Initialize(x => x.AddRegistry<ProductRegistry>());
    }
  }
}

```

```
public class ProductRegistry : Registry
{
    public ProductRegistry()
    {
        For<IProductRepository>().Use<ProductRepository>();
    }
}
```

ممکن است یک WinUI ایجاد کرده باشد که در این صورت به جای فضای نام SoCPatterns.Layered.WebUI از SoCPatterns.Layered.WinUI استفاده نماید.

هدف کلاس BootStrapper این است که تمامی وابستگی‌ها را توسط StructureMap در سیستم Register نماید. زمانی که کدهای کلاینت می‌خواهند به یک کلاس از طریق StructureMap دسترسی داشته باشند، Structuremap وابستگی‌های آن کلاس را تشخیص داده و بصورت خودکار پیاده سازی واقعی (Concrete Implementation) آن کلاس را، براساس همان چیزی که ما برایش تعیین کردیم، به کلاس تزریق می‌نماید. متدهای ConfigureStructureMap باید در همان لحظه‌ای که Application آغاز به کار می‌کند فراخوانی و اجرا شود. با توجه به نوع UI خود یکی از روالهای زیر را انجام دهید:

در WebUI :

فایل Global.asax را به پروژه اضافه کنید و کد آن را بصورت زیر تغییر دهید:

```
namespace SoCPatterns.Layered.WebUI
{
    public class Global : System.Web.HttpApplication
    {
        protected void Application_Start(object sender, EventArgs e)
        {
            BootStrapper.ConfigureStructureMap();
        }
    }
}
```

در WinUI :

در فایل Program.cs کد زیر را اضافه کنید:

```
namespace SoCPatterns.Layered.WinUI
{
    static class Program
    {
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);

            ;()BootStrapper.ConfigureStructureMap

            Application.Run(new Form1());
        }
    }
}
```

سپس برای طراحی رابط کاربری، با توجه به نوع UI خود یکی از روالهای زیر را انجام دهید:

صفحه Default.aspx را باز نموده و کد زیر را به آن اضافه کنید:

```
<asp:DropDownList AutoPostBack="true" ID="ddlCustomerType" runat="server">
    <asp:ListItem Value="0">Standard</asp:ListItem>
    <asp:ListItem Value="1">Trade</asp:ListItem>
</asp:DropDownList>
<asp:Label ID="lblErrorMessage" runat="server" ></asp:Label>
<asp:Repeater ID="rptProducts" runat="server" >
    <HeaderTemplate>
        <table>
            <tr>
                <td>Name</td>
                <td>RRP</td>
                <td>Selling Price</td>
                <td>Discount</td>
                <td>Savings</td>
            </tr>
            <tr>
                <td colspan="5" style="text-align: center; border-top: 1px solid black; border-bottom: 1px solid black; padding: 5px;"><hr /></td>
            </tr>
    </HeaderTemplate>
    <ItemTemplate>
        <tr>
            <td><%# Eval("Name") %></td>
            <td><%# Eval("RRP")%></td>
            <td><%# Eval("SellingPrice") %></td>
            <td><%# Eval("Discount") %></td>
            <td><%# Eval("Savings") %></td>
        </tr>
    </ItemTemplate>
    <FooterTemplate>
        </table>
    </FooterTemplate>
</asp:Repeater>
```

فایل Form1.Designer.cs را باز نموده و کد آن را بصورت زیر تغییر دهید:

```
#region Windows Form Designer generated code
/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.cmbCustomerType = new System.Windows.Forms.ComboBox();
    this.dgvProducts = new System.Windows.Forms.DataGridView();
    this.colName = new System.Windows.Forms.DataGridViewTextBoxColumn();
    this.colRrp = new System.Windows.Forms.DataGridViewTextBoxColumn();
    this.colSellingPrice = new System.Windows.Forms.DataGridViewTextBoxColumn();
    this.colDiscount = new System.Windows.Forms.DataGridViewTextBoxColumn();
    this.colSavings = new System.Windows.Forms.DataGridViewTextBoxColumn();
    ((System.ComponentModel.ISupportInitialize)(this.dgvProducts)).BeginInit();
    this.SuspendLayout();
    // 
    // cmbCustomerType
    // 
    this.cmbCustomerType.DropDownStyle =
    System.Windows.Forms.ComboBoxStyle.DropDownList;
    this.cmbCustomerType.FormattingEnabled = true;
    this.cmbCustomerType.Items.AddRange(new object[] {
        "Standard",
        "Trade"});
    this.cmbCustomerType.Location = new System.Drawing.Point(12, 90);
```

```

this.cmbCustomerType.Name = "cmbCustomerType";
this.cmbCustomerType.Size = new System.Drawing.Size(121, 21);
this.cmbCustomerType.TabIndex = 3;
//
// dgvProducts
//
this.dgvProducts.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
this.dgvProducts.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] {
this.colName,
this.colRrp,
this.colSellingPrice,
this.colDiscount,
this.colSavings});
this.dgvProducts.Location = new System.Drawing.Point(12, 117);
this.dgvProducts.Name = "dgvProducts";
this.dgvProducts.Size = new System.Drawing.Size(561, 206);
this.dgvProducts.TabIndex = 2;
//
// colName
//
this.colName.DataPropertyName = "Name";
this.colName.HeaderText = "Product Name";
this.colName.Name = "colName";
this.colName.ReadOnly = true;
//
// colRrp
//
this.colRrp.DataPropertyName = "Rrp";
this.colRrp.HeaderText = "RRP";
this.colRrp.Name = "colRrp";
this.colRrp.ReadOnly = true;
//
// colSellingPrice
//
this.colSellingPrice.DataPropertyName = "SellingPrice";
this.colSellingPrice.HeaderText = "Selling Price";
this.colSellingPrice.Name = "colSellingPrice";
this.colSellingPrice.ReadOnly = true;
//
// colDiscount
//
this.colDiscount.DataPropertyName = "Discount";
this.colDiscount.HeaderText = "Discount";
this.colDiscount.Name = "colDiscount";
//
// colSavings
//
this.colSavings.DataPropertyName = "Savings";
this.colSavings.HeaderText = "Savings";
this.colSavings.Name = "colSavings";
this.colSavings.ReadOnly = true;
//
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(589, 338);
this.Controls.Add(this.cmbCustomerType);
this.Controls.Add(this.dgvProducts);
this.Name = "Form1";
this.Text = "Form1";
((System.ComponentModel.ISupportInitialize)(this.dgvProducts)).EndInit();
this.ResumeLayout(false);
}
#endregion
private System.Windows.Forms.ComboBox cmbCustomerType;
private System.Windows.Forms.DataGridView dgvProducts;
private System.Windows.Forms.DataGridViewColumn colName;
private System.Windows.Forms.DataGridViewColumn colRrp;
private System.Windows.Forms.DataGridViewColumn colSellingPrice;
private System.Windows.Forms.DataGridViewColumn colDiscount;
private System.Windows.Forms.DataGridViewColumn colSavings;

```

سپس در `Code Behind` ، با توجه به نوع UI خود یکی از روالهای زیر را انجام دهید:

وارد کد نویسی صفحه Default.aspx شده و کد آن را بصورت زیر تغییر دهید:

```
using System;
using System.Collections.Generic;
using SoCPatterns.Layered.Model;
using SoCPatterns.Layered.Presentation;
using SoCPatterns.Layered.Service;
using StructureMap;

namespace SoCPatterns.Layered.WebUI
{
    public partial class Default : System.Web.UI.Page, IProductListView
    {
        private ProductListPresenter _productListPresenter;
        protected void Page_Init(object sender, EventArgs e)
        {
            _productListPresenter = new
ProductListPresenter(this, ObjectFactory.GetInstance<Service.ProductService>());
            this.ddlCustomerType.SelectedIndexChanged +=
                delegate { _productListPresenter.Display(); };
        }
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!Page.IsPostBack)
                _productListPresenter.Display();
        }
        public void Display(IList<ProductViewModel> products)
        {
            rptProducts.DataSource = products;
            rptProducts.DataBind();
        }
        public CustomerType CustomerType
        {
            get { return (CustomerType) int.Parse(ddlCustomerType.SelectedValue); }
        }
        public string ErrorMessage
        {
            set
            {
                lblErrorMessage.Text =
                    String.Format("<p><strong>Error:</strong><br/>{0}</p>", value);
            }
        }
    }
}
```

وارد کدنویسی Form1 شوید و کد آن را بصورت زیر تغییر دهید:

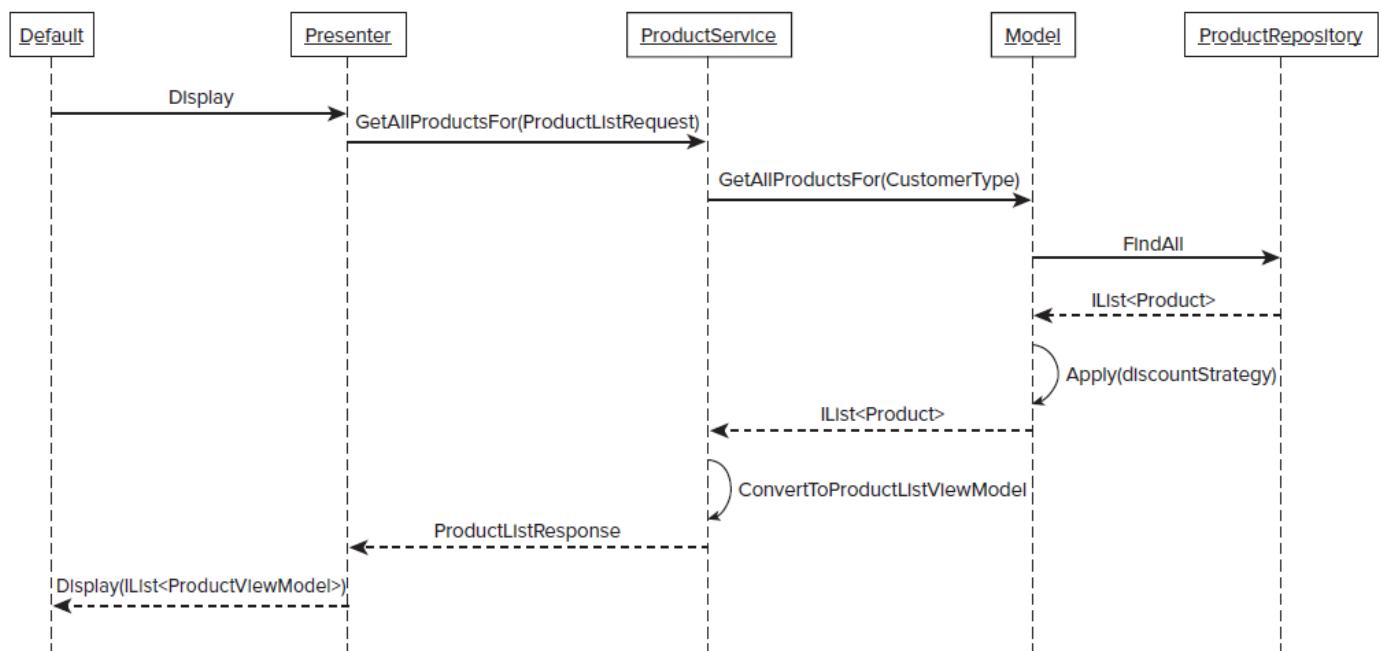
```
using System;
using System.Collections.Generic;
using System.Windows.Forms;
using SoCPatterns.Layered.Model;
using SoCPatterns.Layered.Presentation;
using SoCPatterns.Layered.Service;
using StructureMap;

namespace SoCPatterns.Layered.WinUI
{
    public partial class Form1 : Form, IProductListView
    {
        private ProductListPresenter _productListPresenter;
        public Form1()
        {
            InitializeComponent();
            _productListPresenter =
                new ProductListPresenter(this, ObjectFactory.GetInstance<Service.ProductService>());
            this.cmbCustomerType.SelectedIndexChanged +=
                delegate { _productListPresenter.Display(); };
            dgvProducts.AutoGenerateColumns = false;
        }
    }
}
```

```
        cmbCustomerType.SelectedIndex = 0;
    }
    public void Display(IList<ProductViewModel> products)
    {
        dgvProducts.DataSource = products;
    }
    public CustomerType CustomerType
    {
        get { return (CustomerType)cmbCustomerType.SelectedIndex; }
    }
    public string ErrorMessage
    {
        set
        {
            MessageBox.Show(
                String.Format("Error:{0}{1}", Environment.NewLine, value));
        }
    }
}
```

با توجه به کد فوق، نمونه ای را از کلاس `ProductListPresenter` در لحظه‌ی نمونه سازی اولیه‌ی کلاس UI، ایجاد نمودیم. با استفاده از متدهای `GetInstance` و `GetService` از `StructureMap` مربوط به `ObjectFactory`، نمونه ای از کلاس `ProductService` ایجاد شده است و به سازنده‌ی کلاس `ProductListPresenter` ارسال گردیده است. در مباحث بعدی با جزئیات بیشتری صحبت خواهم کرد. پیاده سازی معماری لایه بندی در اینجا به پایان رسید.

اما اصلا نگران نباشید، شما فقط پرواز کوتاه و مختصری را بر فراز کدهای معماری لایه بندی داشته اید که این فقط یک دید کلی را به شما در مورد این معماری داده است. این معماری هنوز جای زیادی برای کار دارد، اما در حال حاضر شما یک Application با پیوند ضعیف (Loosely Coupled) بین لایه ها دارید که دارای قابلیت تست پذیری قوی، نگهداری و پشتیبانی آسان و تفکیک پذیری قدرتمند بین اجزای آن می باشد. شکل زیر تعامل بین لایه ها و وظایف هر یک از آنها را نمایش می دهد.



نظرات خوانندگان

نویسنده: حامد
تاریخ: ۱۴:۲۹ ۱۳۹۲/۰ ۱/۰۳

ممنون از مقاله خوبتون
به نظر شما امکانش هست برای این معماری یک generator بسازیم به طوری که فقط ما تمام جداول دیتابیس و رابطه‌ی آنها را بسازیم و بعد این generator از روی اون تمام لایه‌ها را بر اساس آن بسازه و بعد ما صرفا جاهایی که نیاز به جزئیات داره را کامل کنیم
آیا نمونه‌ای از این برنامه‌ها هست که این معماری یا معماری‌های مشابه را بسازه؟

نویسنده: شاهین کیاست
تاریخ: ۱۵:۳۱ ۱۳۹۲/۰ ۱/۰۳

اگر با T4 آشنایی داشته باشید بر اساس هر قالبی می‌توانید کد تولید کنید.

نویسنده: حامد
تاریخ: ۱۶:۳۶ ۱۳۹۲/۰ ۱/۰۳

متاسفانه آشنایی ندارم میشه یه توضیح مختصر بدین و یا منبع معرفی کنید

نویسنده: محسن
تاریخ: ۲۳:۵۹ ۱۳۹۲/۰ ۱/۰۳

چون اینجا بحث طراحی مطرح شده یک اصل رو در برنامه‌های وب باید رعایت کرد:

هیچ وقت متن خطای حاصل رو به کاربر نمایش ندید (از لحاظ امنیتی). فقط به ذکر عبارت خطای رخ داده بسنده کنید. خطأ رو مثلاً توسط ELMAH لاغ کنید برای بررسی بعدی برنامه نویس.

نویسنده: شاهین کیاست
تاریخ: ۱:۲۰ ۱۳۹۲/۰ ۱/۰۴

<http://codepanic.blogspot.com/2012/03/t4-enum.html>

نویسنده: M.Q
تاریخ: ۲۲:۱۵ ۱۳۹۲/۰ ۱/۰۴

دوست عزیز غیر از ELMAH ابزار دیگری برای لاغ گیری از خطاهای وجود دارد که قابل اعتماد باشد؟

همچنین اگر ابزاری جهت لاغ گیری از عملیات کاربران (CRUD => حالا R خیلی مهم نیست) می‌شناسید معرفی نمائید.

با سپاس

نویسنده: محسن
تاریخ: ۰:۳۳ ۱۳۹۲/۰ ۱/۰۵

متد auditFields مطرح شده در [مطلوب ردیابی اطلاعات](#) این سایت برای مقصود شما مناسب است.

نویسنده: صابر فتح الهی
تاریخ: ۱۴:۲۳ ۱۳۹۲/۰ ۱/۱۲

سلام با تشکر از شما
من نفهمیدم که توی ASP.NET MVC شما چگونه از الگوی MVP استفاده کردین؟
ظاهراً مثل این قسمت هم توی پست وجود نداره، اگر اشتباه می‌کنم لطفاً تصحیح بفرمایید.

نویسنده: علی
تاریخ: ۱۶:۳ ۱۳۹۲/۰ ۱/۱۲

مثال وب فرم هست. page load و post back داره.

نویسنده: شاهین کیاست
تاریخ: ۱۶:۴ ۱۳۹۲/۰ ۱/۱۲

اگر توجه کنید از الگوی MVP در Web Forms استفاده شده و نه در MVC.

نویسنده: صابر فتح الهی
تاریخ: ۱۸:۳۰ ۱۳۹۲/۰ ۱/۱۲

آقای کیاست و علی آقا
می‌دونم که پروژه چی هست، یکی از UI‌های ما قرار بود MVC باشه خواستم بدونم چطور می‌خوان استفاده کنن، اینجا (در این پست) که می‌دونم Web form ASP.NET هست و در MVC می‌دونم که Page_Load وجود نداره سوال من چیز دیگه بود دوستان

نویسنده: شاهین کیاست
تاریخ: ۱۸:۴۴ ۱۳۹۲/۰ ۱/۱۲

شما گفتید:

سلام با تشکر از شما
من نفهمیدم که توی ASP.NET MVC شما چگونه از الگوی MVP استفاده کردین؟
ظاهراً مثل این قسمت هم توی پست وجود نداره، اگر اشتباه می‌کنم لطفاً تصحیح بفرمایید.
با خواندن کامنت شما برداشت کردم شما تصور کردید کدھای پست جاری مربوط به تکنولوژی ASP.NET MVC هست.

به نظر نویسنده هنوز برای MVC و WPF مثال‌ها را ایجاد نکرده و توضیح نداده اند.
اما برای استفاده از این نوع معماری در MVC کار خاصی لازم نیست انجام شود. همانطور که قبل از مثال‌های آقای نصیری دیده ایم کافی است Controller Service Layer در Model مناسب را تعذیه کند و برای View فراهم کند.

نویسنده: صابر فتح الهی
تاریخ: ۱۹:۲۶ ۱۳۹۲/۰ ۱/۱۲

من هم با توجه به مثال آقای نصیری و استفاده از الگوی کار گیج شدم، این معماری یک لایه Repository دارد، من الگوی کار توی این لایه پیاده کردم، با پیاده سازی در این لایه به نظر می‌اد لایه سرویس کاربردش از دست میده توی پست‌های قبل هم از آقای خوشبخت سوال کردم اما طاهرها هنوز وقت نکردن پاسخ بدند.

مورد دوم اینکه در این پست الگوی کار شرح داده شده و پیاده سازی شده، و در این پست گفته شده " حین استفاده از EF code first، الگوی واحد کار، همان DbContext است و الگوی مخزن، همان DbSet‌ها. ضرورتی به ایجاد یک لایه محافظ اضافی بر روی

این‌ها وجود ندارد. " با توجه به این مسائل کلا مسائل قاطی کردم متاسفانه آقای نصیری هم سرشون شلوغ و درگیر دوره‌ها است، که بحثی بر سر این معماری بشه.

نویسنده: شاهین کیاست
تاریخ: ۲۰:۴۶ ۱۳۹۲/۰۱/۱۲

روشی که در مثال آقای نصیری گفته شده با روش این سری مقالات کمی متفاوت هست.
در آنجا از روکش اضافه برای Repository استفاده نشده همچنین از الگوی واحد کار استفاده شده.
به علاوه این سری مقالات ممکن است هنوز تکمیل نشده باشند.
به نظر من هر کس با توجه به میزان اطلاعاتی که دارد و درکی که از الگوها دارد با مقایسه‌ی روش‌ها و مقالات می‌تواند تصمیم بگیرد چه معماری به کار بگیرد.

نویسنده: صابر فتح الهی
تاریخ: ۲۱:۳ ۱۳۹۲/۰۱/۱۲

حرف شما کاملاً متنین هست

من قبلًا معماری سه لایه کار می‌کردم، که نمونه اون توی همین سایت بخش پیروزه‌ها گذاشتم، اما الان با EF، MVC کمی به مشکل برخوردم و درست نتونستم تا حالا لایه‌های مورد نظر برای خودم در پیروزه‌ها تفکیک کنم، این معماری به نظرم جالب اومد، خواستم که الگوی کار هم توی اون به کار ببرم که به مشکل برخوردم (چون درک درستی از الگوی کار پیدا نکردم یا شاید کلا دارم اشتباه می‌کنم). البته به قول شما شاید این معماری هنوز تکمیل نشده پیروزه اش، در هر صورت از پاسخ‌های شما متشکرم.

نویسنده: شاهین کیاست
تاریخ: ۲۱:۷ ۱۳۹۲/۰۱/۱۲

خواهش می‌کنم.
 فقط جهت یادآوری مثال روش آقای نصیری با پوشش MVC و EF قابل دریافت است.

نویسنده: ابوالفضل روشن ضمیر
تاریخ: ۱:۴۵ ۱۳۹۲/۰۱/۱۷

سلام
با تشکر فروان از شما ...
اگر امکان داره این مثال که در قالی یک پیروزه نوشته شده برای دانلود قرار دهید ... تا بهتر بتوانیم برنامه را تجزیه و تحلیل کنیم
.... ممنون

نویسنده: فرشید علی اکبری
تاریخ: ۱۵:۵۳ ۱۳۹۲/۰۱/۱۹

با سلام و تشکر از زحمات کلیه دوستان
با زحمتی که آقای خوشبخت تا اینجا کشیدن فکر کنم در صورتیکه خودمون مقاله مربوطه به این پیروزه رو قدم به قدم بخونیم و طراحی کینم خیلی بهتر متوجه می‌شیم تا اینکه اونو آماده دانلود کنیم. من با این روش پیش رفتم و برای ایجاد اون با step by step کردن مرا حلش حدود 45 دقیقه وقت گذاشتم ولی درصد یادگیریش خیلی بالاتر بود تا گرفتن فایل آماده...
در ضمن لازمه بگم که بخارط رفع شک و شبکه در سرعت پردازش و بالا امدن اطلاعات، من تست این روش رو با تعداد 155 هزار رکورد انجام دادم که کمتر از سه ثانیه برآم لود شد... باوجودیکه کامپوننت‌های دات نت بار مختلفی رو هم روی فرمم قرار دادم که بیشتر به اهمیت لود اطلاعاتم در پیروزه و فرم‌های واقعی پی ببرم.
سئوال اینکه :

به نظر شما ما می‌تونیم روی این لایه‌ها الگوی واحد کار رو هم ایجاد کنیم یا نه؟ اصلاً ضرورتی داره؟

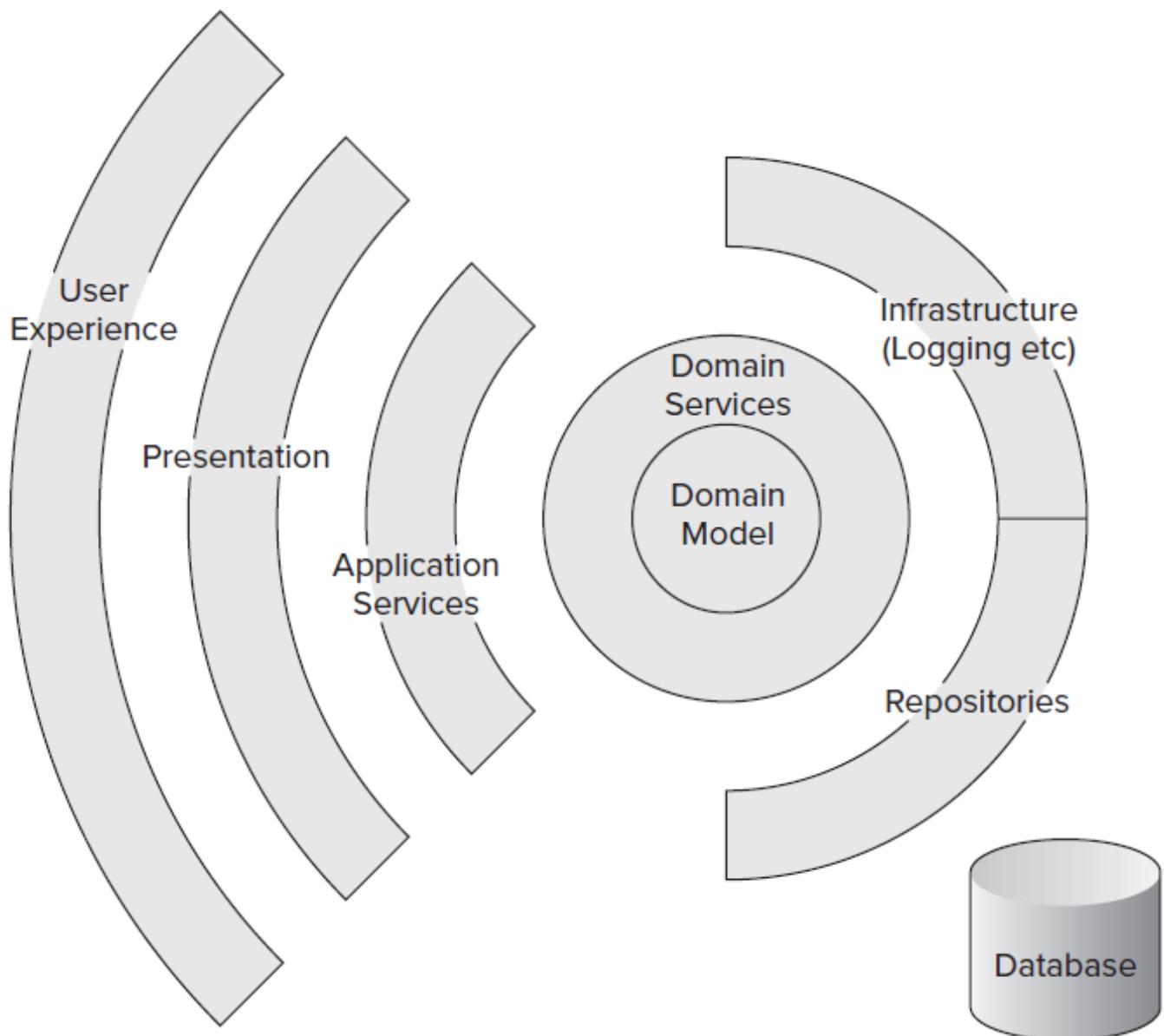
نویسنده: علیرضا کیانی مقدم
تاریخ: ۱۲:۸ ۱۳۹۲/۰۱/۲۸

با تشکر از نویسنده مقاله و اهتمام ایشان به بررسی دقیق مفاهیم ، از آنجا که reusable و flexible بودن برنامه ها را نمی توان نادیده گرفت تا آنجا که این تفکیک پذیری خود به مسئله ای بغرنج تبدیل نشده و تکرار داده ها و پاس دادن غیر ضرور آنها را موجب نشود تلاش در این باره مفید خواهد بود . امروزه توسعه دهنده گان به سمت کم کردن لایه های فرسایشی و حذف پیچیدگی های غیر ضرور قدم بر می دارند. خلق عبارات لامبادا در دات نت و delegate ها نمونه هایی از تلاش بشر برنامه نویس در این باره است .

نویسنده: مسعود ۲
تاریخ: ۹:۱۲ ۱۳۹۲/۰۲/۰۹

سلام

validation-2 و business Rule ها در کجا این معماری اعمال می شوند؟



منظور از DomainService چیست؟

ممکنه منابع بیشتری معرفی نمایید؟
ممنون.

نوبنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۰۹ ۱۶:۲۶

[منبع برای مطالعه بیشتر](#)

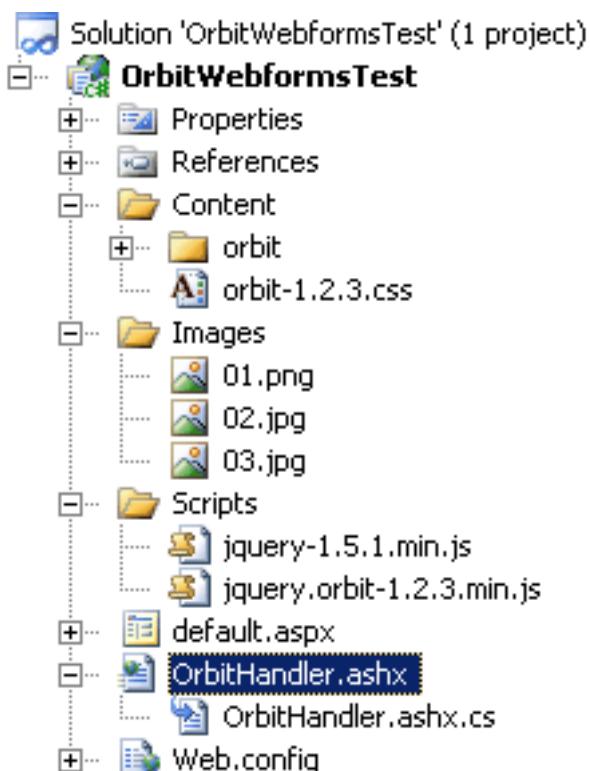
هر از چندگاهی سؤال «این مثال jQuery را نمی‌تونم اجرا یا باز سازی کنم» در این سایت یا سایت‌های مشابه تکرار می‌شوند. بنابراین بهتر است نحوه عیب یابی برنامه‌های ASP.NET مبتنی بر jQuery را یکبار با هم مرور کنیم. در اینجا، مثال تهیه یک Slider را که [پیشتر در سایت مطرح شده](#) است، به نحوی دیگر بررسی خواهیم کرد:

- (1) فرم‌آوش می‌کنیم تا اسکریپت اصلی jQuery را به درستی پیوست و مسیردهی کنیم.
- (2) مسیر Generic handler را ذکر می‌کنیم.
- (3) مسیرهای تصاویری را که Image slider باید نمایش دهد، کاملاً بی‌ربط ذکر می‌کنیم.
- (4) خروجی JSON نامربوطی را بازگشت می‌دهیم.
- (5) یکبار هم یک استثنای عمده دستی را در بین کدها قرار خواهیم داد.

و ... بعد سعی می‌کنیم با استفاده از [Firebug](#) عیوب فوق را یافته و اصلاح کنیم؛ تا به یک برنامه قابل اجرا برسیم.

معرفی برنامه‌ای که کار نمی‌کند!

یک برنامه ASP.NET Empty web application را آغاز کنید. سپس سه پوشه Scripts، Content و Images را به آن اضافه نمائید. در این پوشه‌ها، اسکریپت‌های نمایش دهنده تصاویر، CSS آن و تصاویری که قرار است نمایش داده شوند، قرار می‌گیرند:



سپس یک فایل default.aspx و یک فایل OrbitHandler.ashx را نیز به پروژه با محتویات ذیل اضافه کنید: (در این دو فایل، ۵ مورد مشکل ساز یاد شده لحاظ شده‌اند)

محتویات فایل OrbitHandler.cs مطابق کدهای ذیل است:

```

using System.Collections.Generic;
using System.IO;
using System.Web;
using System.Web.Script.Serialization;

namespace OrbitWebformsTest
{
    public class Picture
    {
        public string Title { set; get; }
        public string Path { set; get; }
    }

    public class OrbitHandler : IHttpHandler
    {
        IList<Picture> PicturesDataSource()
        {
            var results = new List<Picture>();
            var path = HttpContext.Current.Server.MapPath("~/Images");

            foreach (var item in Directory.GetFiles(path, "*.*"))
            {
                var name = Path.GetFileName(item);
                results.Add(new Picture
                {
                    Path = /*"Images/" + name*/ name,
                    Title = name
                });
            }

            return results;
        }

        public void ProcessRequest(HttpContext context)
        {
            var items = PicturesDataSource();
            var json = /*new JavaScriptSerializer().Serialize(items)*/ string.Empty;
            throw new InvalidDataException("همینطوری");
            context.Response.ContentType = "text/plain";
            context.Response.Write(json);
        }

        public bool IsReusable
        {
            get { return false; }
        }
    }
}

```

در اینجا جهت سهولت دموی برنامه (و همچنین امکان باز تولید آن توسط خوانندگان)، از بانک اطلاعاتی استفاده نشده و عمداً از یک لیست جنریک تشکیل شده در حافظه کمک گرفته شده است. تصاویر برنامه در پوشش Images واقع در ریشه سایت، قرار دارند. بنابراین توسط متدهای PicturesDataSource، فایل‌های این پوشش را یافته و مطابق ساختار کلاس Picture بازگشت می‌دهیم. نهایتاً این اطلاعات به ظاهر قرار است با فرمات JSON بازگشت داده شوند تا بتوان نتیجه را توسط افزونه Orbit استفاده کرد.

همچنین کدهای صفحه ASPX ای که قرار است (به ظاهر البته) از این Generic handler استفاده کند به نحو ذیل است:

```

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="default.aspx.cs"
Inherits="OrbitWebformsTest._default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <link href="Content/orbit-1.2.3.css" rel="stylesheet" type="text/css" />
    <script src="Script/jquery-1.5.1.min.js" type="text/javascript"></script>
    <script src="Scripts/jquery.orbit-1.2.3.min.js" type="text/javascript"></script>
</head>
<body>
    <form id="form1" runat="server">
        <div id="featured">
        </div>
    </form>
    <script type="text/javascript">

```

```
$function () {
    $.ajax({
        url: "Handler.ashx",
        contentType: "application/json; charset=utf-8",
        success: function (data) {
            $.each(data, function (i, b) {
                var str = '';
                $("#featured").append(str);
            });
            $('#featured').orbit();
        },
        dataType: "json"
    });
}
</script>
</body>
</html>
```

خوب! اگر پروژه را اجرا کنیم، کار نمی‌کند. یک مستطیل مشکی رنگ در کنار صفحه ظاهر شده و همین! حالا چکار باید کرد؟

مراحل عیب یابی برنامه‌ای که کار نمی‌کند!

ابتدا برنامه را در فایرفاکس باز کرده و سپس افزونه [Firebug](#) را با کلیک بر روی آیکن آن، بر روی سایت فعال می‌کنیم. سپس یکبار بر روی دکمه F5 کلیک کنید تا مجدداً مراحل بارگذاری سایت تحت نظر افزونه Firebug فعال شده، طی شود.



اولین موردی که مشهود است، نمایش عدد 3، کنار آیکن فایرباگ می‌باشد. این عدد به معنای وجود خطاهای اسکریپتی در کدهای ما است.

برای مشاهده این خطاهای، بر روی برگه Console آن کلیک کنید:

A screenshot of the Firebug Console tab. It displays several error messages in red text:

- NetworkError: 404 Not Found - http://localhost:1031/Script/jquery-1.5.1.min.js
- ReferenceError: jQuery is not defined
- ReferenceError: \$ is not defined
- ReferenceError: \$ is not defined

The 'All' tab is selected in the toolbar above the console.

بله. مشخص است که مسیر دهی فایل jquery-1.5.1.min.js صحیح نبوده و همین مساله سبب بروز خطاهای اسکریپتی گردیده است. برای اصلاح آن سطر زیر را در برنامه تغییر دهید:

```
<script src="Scripts/jquery-1.5.1.min.js" type="text/javascript"></script>
```

پیشتر پوشه Script ذکر شده بود که باید تبدیل به Scripts شود.

مجددا دکمه F5 را فشرده و سایت را با تنظیمات جدید اجرا کنید. اینبار در برگه Console و یا در برگه شبکه فایرباگ، خطای یافت نشدن Generic handler نمایان می‌شوند:

URL	Status	Domain
GET localhost:1031	200 OK	localhost:1031
GET orbit-1.2.3.css	200 OK	localhost:1031
GET jquery-1.5.1.min.js	200 OK	localhost:1031
GET jquery.orbit-1.2.3.min.js	200 OK	localhost:1031
GET loading.gif		localhost:1031
GET Handler.ashx	404 Not Found	localhost:1031
GET loading.gif	200 OK	localhost:1031
6 requests		

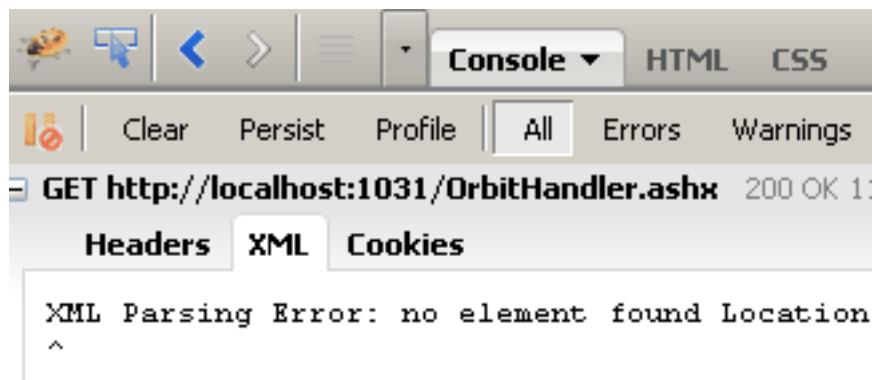
برای رفع آن به فایل default.aspx مراجعه و بجای معرفی Handler.ashx، نام OrbitHandler.ashx را وارد کنید.
مجددا دکمه F5 را فشرده و سایت را با تنظیمات جدید اجرا کنید.

```

<html>
  <head>
    <title>مینیتوری</title>

```

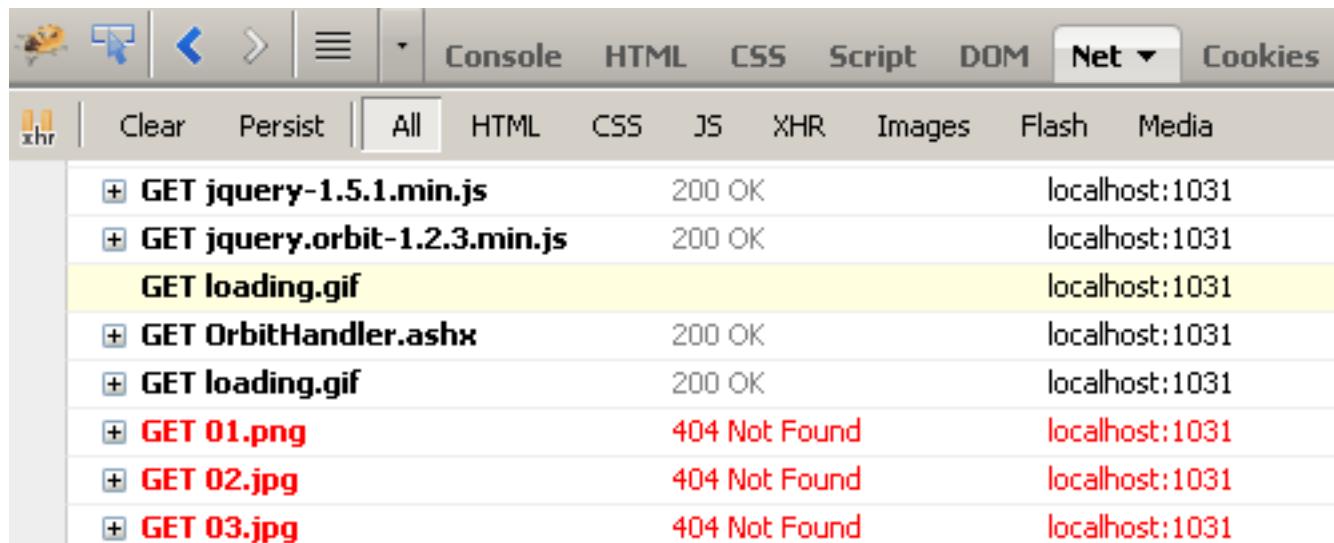
اگر به برگه کنسول دقیق کنیم، بروز استثناء در کدها تشخیص داده شده و همچنین در برگه Response پاسخ دریافتی از سرور، جزئیات صفحه خطای بازگشتی از آن نیز قابل بررسی و مشاهده است.
اینبار به فایل OrbitHandler.cs مراجعه کرده و سطر throw new InvalidDataException را حذف می‌کنیم. در ادامه برنامه را کامپایل و مجدداً اجرا خواهیم کرد.



با اجرای مجدد سایت، تبادل اطلاعات صحیحی با فایل OrbitHandler.ashx برقرار شده است، اما خروجی خاصی قابل مشاهده نیست. بنابراین بازهم سایت کار نمی‌کند.
برای رفع این مشکل، متد ProcessRequest را به نحو ذیل تغییر خواهیم داد:

```
public void ProcessRequest(HttpContext context)
{
    var items = PicturesDataSource();
    var json = new JavaScriptSerializer().Serialize(items);
    context.Response.ContentType = "text/plain";
    context.Response.Write(json);
}
```

برنامه را کامپایل کرده و اجرا می‌کنیم. برنامه اجرا می‌شود، اما باز هم کار نمی‌کند. مشکل از کجاست؟



بله. تمام تنظیمات به نظر درست هستند، اما در برگه شبکه فایرباگ تعدادی خطای 404 و یا «یافت نشد»، مشاهده می‌شوند. مشکل اینجا است که مسیرهای بازگشت داده شده توسط متد Directory.GetFiles، مسیرهای مطلقی هستند؛ مانند `c:\path\images\01.jpg` و جهت نمایش در یک وب سایت مناسب نمی‌باشند. برای تبدیل آنها به مسیرهای نسبی، اینبار کدهای متد تهیه منبع داده را به نحو ذیل ویرایش می‌کنیم:

```
IList<Picture> PicturesDataSource()
{
    var results = new List<Picture>();
```

```

var path = HttpContext.Current.Server.MapPath("~/Images");
foreach (var item in Directory.GetFiles(path, "*.*"))
{
    var name = Path.GetFileName(item);
    results.Add(new Picture
    {
        Path = "Images/" + name,
        Title = name
    });
}
return results;
}

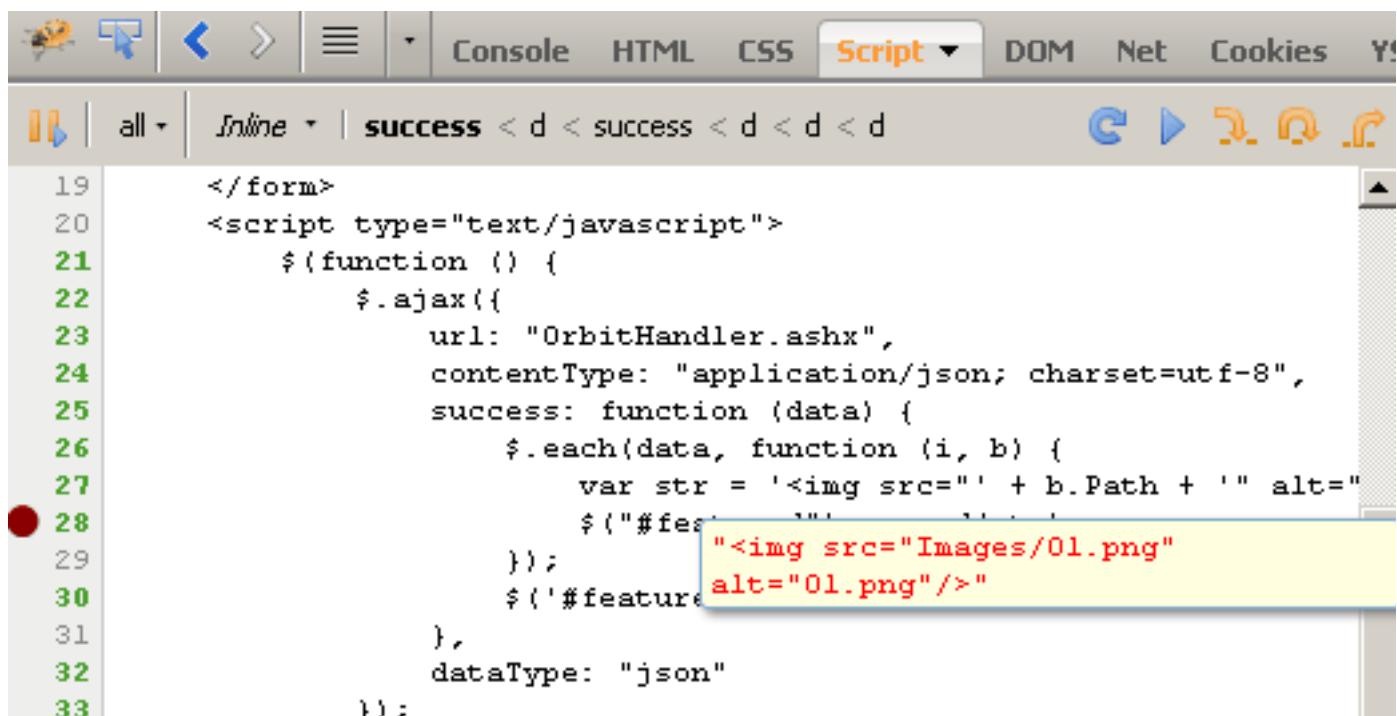
```

در این کدها فقط قسمت Path ویرایش شده است تا به مسیر پوشش Images واقع در ریشه سایت اشاره کند.
اینبار اگر برنامه را اجرا کنیم، بدون مشکل کار خواهد کرد.

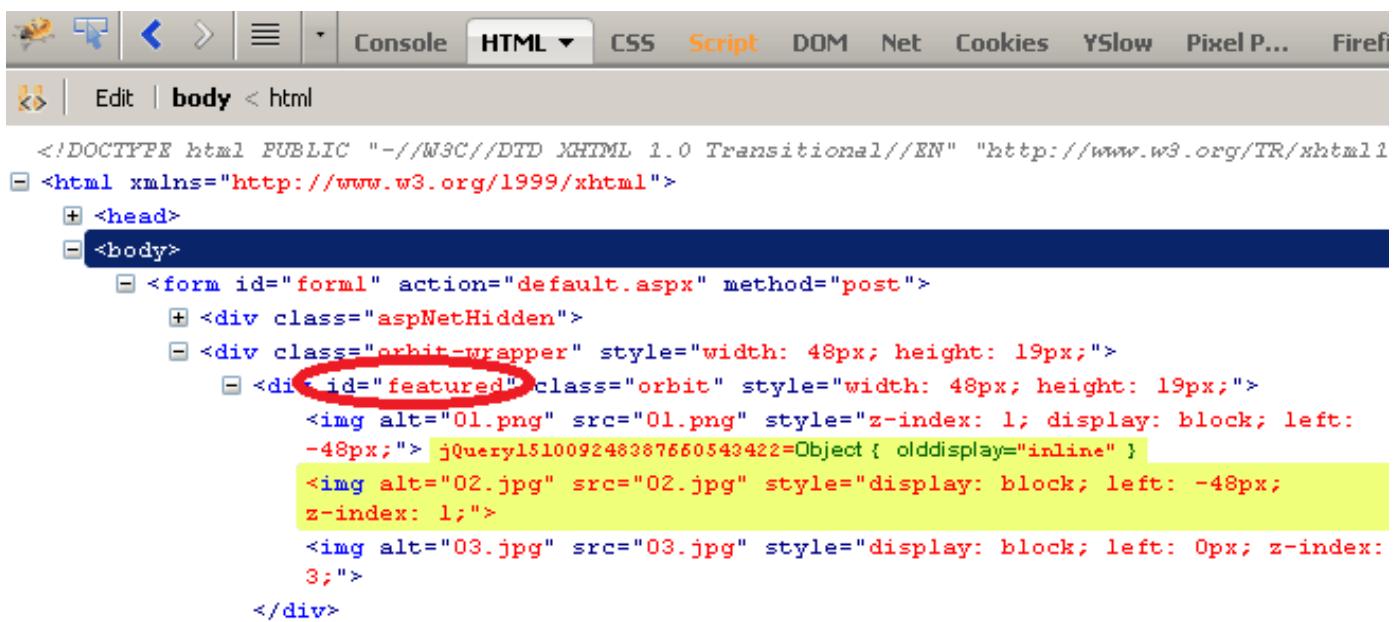
بنابراین در اینجا مشاهده کردیم که اگر «برنامه‌ای مبتنی بر jQuery کار نمی‌کند»، چگونه باید قدم به قدم با استفاده از فایرباگ و امکانات آن، به خطاهایی که گزارش می‌دهد و یا مسیرهایی را که یافت نشد بیان می‌کند، دقت کرد تا بتوان برنامه را عیب یابی نمود.

سؤال مهم: اجرای کدهای Ajax jQuery فوق، چه تغییری را در صفحه سبب می‌شوند؟

اگر به برگه اسکریپت‌ها در کنسول فایرباگ مراجعه کنیم، امکان قرار دادن breakpoint بر روی سطرهای کدهای جاوا اسکریپت نمایش داده شده نیز وجود دارد:



در اینجا همانند VS.NET می‌توان برنامه را در مرورگر اجرا کرده و تگ‌های تصویر پویای تولید شده را پیش از اضافه شدن به صفحه، مرحله بررسی کرد. به این ترتیب بهتر می‌توان دریافت که آیا src بازگشت داده شده از سرور فرمات صحیحی دارد یا خیر و آیا به محل مناسبی اشاره می‌کند یا نه. همچنین در برگه HTML آن، عناصر پویای اضافه شده به صفحه نیز بهتر مشخص هستند:



The screenshot shows the Firebug interface with the 'HTML' tab selected. The DOM tree is displayed, starting with the DOCTYPE declaration and the <html> tag. The 'body' tag is highlighted with a dark blue background. Inside the body, there is a form with id='form1', followed by an <aspNetHidden> div. The next node is a div with class='orbit-wrapper' and style='width: 48px; height: 19px;'. Inside this wrapper is a div with id='featured' and class='orbit', also with width and height set to 48px and 19px respectively. This div contains three images: '01.png', '02.jpg', and '03.jpg'. The '02.jpg' image is highlighted with a yellow selection box. A red oval surrounds the 'featured' div. The '02.jpg' image has a yellow border around its code line.

```
</DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml11<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
  <body>
    <form id="form1" action="default.aspx" method="post">
      <div class="aspNetHidden">
      <div class="orbit-wrapper" style="width: 48px; height: 19px;">
        <div id="featured" class="orbit" style="width: 48px; height: 19px;">
           jquery151009248387560543422=Object { olddisplay="inline" }
          
          
        </div>
      </div>
    </form>
  </body>
</html>
```

نظرات خوانندگان

نوبنده: صادق
تاریخ: ۱۳۹۲/۰۲/۰۴ ۱۲:۵۱

ممnon آقای نصیری - برای کروم چطور، آیا developer tools کفایت میکند یا ابزار بهتری سراغ دارد؟
البته من گاهی که نیاز به دستکاری ریسپانس و ریکوئست باشه از فیدلر استفاده میکنم

نوبنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۰۴ ۱۲:۵۵

تمام مواردی که در بحث جاری با فایرباگ مطرح شدن به صورت توکار در developer tools مربوط به کروم هم وجود دارند و مباحث آن تقریباً یکی است.

نوبنده: محسن جمشیدی
تاریخ: ۱۳۹۲/۰۲/۰۶ ۱۰:۴۹

یکی از مواردی هم که ممکن است آزار دهنده باشد خطای سینتکسی است که در FireFox با زدن Ctrl+Shift+J قابل مشاهده خواهد بود.

نوبنده: Abolfazl
تاریخ: ۱۳۹۲/۰۲/۰۸ ۲۳:۴۸

سلام آقای نصیری ..ممnonم ..خیلی عالی بود
من قبل از این مشکل را داشتم و نمی‌توانستم برطرفش کنم ..الآن درست شد..مرسی

نوبنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۰۹ ۱۶:۴۲

```
$(document).ready(function () {  
    $(':radio').click(function () {  
        debugger;  
    });  
});
```

یک نکته جانبی است برای فعال سازی دیباگر خود ویژوال استودیو در حین کار با جیکوئری

نوبنده: ahmad.valipour
تاریخ: ۱۳۹۲/۰۲/۱۴ ۱۲:۱۹

سلام
سوالی برام پیش اومده. اگر ما 10 تا عکس تو پوشه image داشته باشیم ولی فقط 5 تا اونها رو در صفحه ارجاع بهشون داشته باشیم، فقط اون 5 تا به طرف کلاینت میرن. درسته؟
پس چرا وقتی با firebug مسیر عکس رو تغییر میدیم به عکس دیگه، میشناسه اون عکس رو، و نشون میده؟

نوبنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۱۴ ۱۲:۲۵

همکاری متقابل موتور مرورگر و فایرباگ. فایرباگ درخواست می‌کنه، مرورگر دریافت و ارائه.
چیزی مثل دیباگر در VS.NET. زمانیکه مثلا در کدهای کار با Entity framework روی سط्रی break point قرار می‌دید و خروجی

یک کوئری را بررسی می‌کنید، این دیاگر قابلیت دریافت مقادیر بررسی شده و حتی نشده را هم از بانک اطلاعاتی دارا است (حتی اگر این مقادیر، در کوئری اولیه درخواست نشده باشند؛ نوعی lazy loading در اینجا صورت می‌گیرد)

نویسنده: ابوالفضل روشن ضمیر
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۱:۵

سلام

من اولین بار که اجرا می‌کنم تصویر را نشون نمی‌ده به این صورت نمایش میده :



بعد از لود صفحه یک بار که صفحه را Refresh کنم همه چیز درست میشه ؟
در قسمت Console هم خطای وجود ندارد
ممnon

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۱:۲۹

حداقل دو علت می‌توانه داشته باشد:

الف) تصاویر رو نمی‌تونه پیدا کنه، یا صفحه کش شده بیش از حد. قسمت «اجرای کدهای AjaxjQuery» فوق، چه تغییری را در صفحه سبب می‌شوند؟» را بررسی کنید که چه آدرسی توسط کدهای جی‌کوئری در حال پردازش است.
همچنین کش شدن نتایج قبلی رو هم می‌شود غیرفعال کرد:

```
$.ajax({  
    cache: false /* گاهی از اوقات خصوصا برای آی ای نیاز است */  
});
```

ب) چند وقت قبل در یکی از بحث‌های سایت دیدم که مورد زیر رعایت نشده بود و کدهای جی‌کوئری کار نمی‌کردند:

```
<script type="text/javascript">  
    $(function () {  
        کدهای جی‌کوئری در اینجا //  
    });  
</script>
```

اجرای کدهای جی‌کوئری نیازی به DOM حاضر و آماده دارند که توسط متド document ready آن مانند کدهای فوق باید تدارک دیده شود. نیازی به این کد نخواهد بود اگر اسکریپت‌ها در آخر صفحه و پیش از بسته شدن تگ body اضافه بشن.

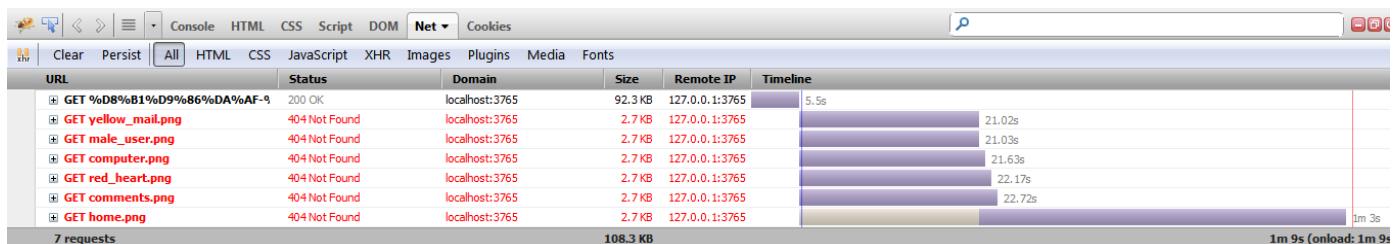
نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۰۷/۰۷ ۱۰:۵۸

با تشکر از مطلب بسیار کاربردیتون ، در هنگام استفاده از URL Routing که قبل راهنماییم کرده بودید برای آدرس فایل‌های جاوا اسکریپت از

```
<%=ResolveUrl("~/App_Themes/MainTheme/jquery.js")%>
```

استفاده کردم و مشکل حل شد ولی برای یو آر ال این تصاویر آیکون که در Jquery تعریف شدند میتوانید کمک کنید

```
<script type="text/javascript">
$(document).ready(function () {
    $('#exampleMenu').sweetMenu({
        top: 200,
        padding: 8,
        iconSize: 48,
        easing: 'easeOutBounce',
        duration: 500,
        icons: [
            'images/home.png',
            'images/comments.png',
            'images/red_heart.png',
            'images/computer.png',
            'images/male_user.png',
            'images/yellow_mail.png'
        ]
    });
});
</script>
```



اگر آدرس آیکون‌ها را به صورت

'<http://site.ir/images/home.png>'

تعریف کنم مشکل حل میشه ولی فکر کنم راه حل درستی نباشه . بسیار ممنون

نویسنده: **وحید نصیری**
تاریخ: **۱۱:۲۸ ۱۳۹۲/۰۷/۰۷**

امکان قرار دادن کدهای سمت سرور داخل اسکریپت‌ها هم هست؛ مثلا:

```
data: {"username": '' + $('#<%= TextBox1.ClientID %>').val() + "'"},
```

به این شرط که این اسکریپت داخل صفحه runat=server دار باشد یا داخل head ای باید مشخصات.

امروز می‌خواهیم نحوه کار با FTP بصورت ساده برای کاربران و برنامه نویسان مبتدی را آموختش بدیم.

برای استفاده از FTP نیاز به یک اکانت FTP در سایت مورد نظر به مردم دسترسی به پوشش ای مشخص می‌باشد.
برای مثال ما یک اکانت FTP در سایت dotnettipp.info داریم که به پوشش upload دسترسی دارد.

ابتدا در فایل Web.config و در بین تگ های appSettings مقادیر زیر را برای دسترسی به اکانت و نام کاربری و رمز عبور ذخیره می‌کنیم.

```
<add key="FtpAddress" value="ftp://ftp.dotnettips.info" />
<add key="FtpUser" value="uploadcenter" />
<add key="FtpPass" value="123123" />
<add key="FolderPath" value="~/Upload/" />
```

*نکته : برای امنیت بیشتر و دسترسی به اطلاعات اکانت می‌شود از روش‌های دیگری نیز استفاده کرد.

در ادامه یک کلاس در App_code پروژه خود با نام FtpHelper ایجاد می‌کنیم و کد زیر را در آن قرار می‌دهیم:
تکه کد بالا برای ساخت کردن مقادیر نام کاربری و رمز عبور و آدرس FTP در کلاس مذکور که بصورت پیش‌فرض از web.config پر می‌شود ایجاد و بکار خواهد رفت.

```
using System.Net;
using System.IO;
using System.Configuration;

public class FtpHelper
{
  public FtpHelper()
  {
    //Default Value Set From Application
    _hostname = ConfigurationManager.AppSettings["FtpAddress"][0];
    _username = ConfigurationManager.AppSettings["FtpUser"][0];
    _password = ConfigurationManager.AppSettings["FtpPass"][0];
  }

  #region "Properties"
  private string _hostname;
  /// <summary>
  /// Hostname
  /// </summary>
  /// <value></value>
  /// <remarks>Hostname can be in either the full URL format
  /// ftp://ftp.myhost.com or just ftp.myhost.com
  /// </remarks>
  public string Hostname
  {
    get
    {
      if (_hostname.StartsWith("ftp://"))
      {
        return _hostname;
      }
      else
      {
        return "ftp://" + _hostname;
      }
    }
    set
    {
      _hostname = value;
    }
  }
  private string _username;
```

نحوه کار با ftp - بخش اول

```
/// <summary>
/// Username property
/// </summary>
/// <value></value>
/// <remarks>Can be left blank, in which case 'anonymous' is returned</remarks>
public string Username
{
    get
    {
        return (_username == "" ? "anonymous" : _username);
    }
    set
    {
        _username = value;
    }
}
private string _password;
public string Password
{
    get
    {
        return _password;
    }
    set
    {
        _password = value;
    }
}

#endregion
}
```

سپس فضای نام‌های زیر را در کلاس خود قرار می‌دهیم.

```
using System.Net;
using System.IO;
```

حالا برای بارگذاری فایل می‌توانیم از یک تابع بصورت shared استفاده کنیم که بتوان با دادن آدرس فایل بصورت فیزیکی به تابع و مشخص کردن پوشه مورد نظر آنرا در هاست مقصود (FTP) بارگذاری کرد. توجه داشته باشید که تابع فوق فوق نیازی به قرار گرفتن در کلاس بالا (FtpHelper) ندارد. یعنی می‌توان آنرا در هرجای برنامه پیاده سازی نمود.

```
public static bool Upload(string fileUrl)
{
    if (File.Exists(fileUrl))
    {
        FtpHelper ftpClient = new FtpHelper();
        string ftpUrl = ftpClient.Hostname + System.IO.Path.GetFileName(fileUrl);

        FtpWebRequest ftp = (FtpWebRequest)FtpWebRequest.Create(ftpUrl);
        ftp.Credentials = new NetworkCredential(ftpClient.Username, ftpClient.Password);

        ftp.KeepAlive = true;
        ftp.UseBinary = true;
        ftp.Timeout = 3600000;
        ftp.KeepAlive = true;
        ftp.Method = WebRequestMethods.Ftp.UploadFile;

        const int bufferLength = 102400;
        byte[] buffer = new byte[bufferLength];
        int readBytes = 0;

        //open file for reading
        using (FileStream fs = File.OpenRead(fileUrl))
        {
            try
            {
                //open request to send
                using (Stream rs = ftp.GetRequestStream())
```

```
{  
    do  
    {  
        readBytes = fs.Read(buffer, 0, bufferLength);  
        fs.Write(buffer, 0, readBytes);  
    } while (!(readBytes < bufferLength));  
    rs.Close();  
}  
  
}  
  
catch (Exception)  
{  
    //Optional Alert for Exception To Application Layer  
    //throw (new ApplicationException("بارگذاری فایل با خطأ رو به رو شد"));  
  
}  
finally  
{  
    //ensure file closed  
    //fs.Close();  
}  
  
}  
  
ftp = null;  
return true;  
}  
return false;  
  
}
```

نکه کد بالا فایل مورد نظر را در صورت وجود به صورت نکههای 100 کیلوبایتی بر روی ftp بارگذاری می‌کند، که می‌توانید مقدار آنرا نیز تغییر دهید.
اینکار باعث افزایش سرعت بارگذاری در فایل‌های با حجم بالا برای بارگذاری می‌شود.

در بخش‌های بعدی نحوه ایجاد پوشه، حذف فایل، حذف پوشه و دانلود فایل از روی FTP را بررسی خواهیم کرد.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۲۲:۳۸ ۱۳۹۲/۰۲/۱۱

ممنون از مطلب شما.

چند نکته جزئی در مورد کدهای تهیه شده:

- وجود این try/catch در اینجا هیچ هدفی را برآورده نکرده. از قسمت throw هم توصیه می‌شود که استفاده نکنید. از finally استفاده کنید تا stack trace پاک نشه. به علاوه زمانیکه مشغول به طراحی یک کتابخانه هستید تا حد ممکن از ذکر خودداری کنید. وظیفه بررسی این مسایل مرتبط است به لایه‌های بالاتر استفاده کننده و نه کتابخانه پایه.

- از ابتدای متدهم ضرورتی ندارد. اگر قرار است باشد، باید به استفاده کننده در طی یک استثناء اعلام شود که چرا فایل درخواستی او آپلود نشده. در کل استفاده از متده File.Exists به همراه صدور استثناء در صورت عدم وجود فایل، در اینجا مناسب‌تر است.

- نامگذاری‌هایی مانند obj_ftp مربوط به دوران C است. در سی‌شارپ روش دیگری را باید استفاده کنید که در مطلب [اصول نامگذاری در دات نت](#) به تفصیل بررسی شده.

- بررسی صفر بودن readBytes بهتر است پیش از فراخوانی متده Write انجام شود.

- یک سری از اشیاء در دات نت پیاده سازی کننده IDispoable هستند. به این معنا که بهتر است از using برای استفاده از آن‌ها کمک گرفته شود تا کامپایلر قسمت finally به همراه آزاد سازی منابع را به صورت خودکار اضافه کند. این نکته برای مواردی که در بین کار استثنایی رخ می‌دهد جهت آزاد سازی منابع لازم است. یعنی بهتر بود بجای try/catch از try/finally و یا using در مکان‌های مناسب استفاده می‌شد.

- علت استفاده از شیء Application در اینجا چه چیزی بوده؟ AppSettings خوانده شده از وب کانفیگ برنامه و کل اطلاعات آن در آغاز به کار یک برنامه ASP.NET به صورت خودکار کش می‌شوند. به همین جهت است که اگر حتی یک نقطه در فایل وب کانفیگ تغییر کند برنامه ASP.NET [ری استارت می‌شود](#) (تا دوباره تنظیمات را بخواند). بنابراین مستقیماً از همان امکانات ConfigurationManager بدون انتساب آن به شیء سراسری Application استفاده کنید. اینکار سرباری آنچنانی هم ندارد؛ چون از حافظه خوانده می‌شود و نه از فایل. هر بار فراخوانی ConfigurationManager.AppSettings به معنای مراجعه به فایل web.config نیست. فقط بار اول اینکار انجام می‌شود؛ تا زمانیکه این فایل تغییر کند یا برنامه ری استارت شود.

نویسنده: بهمن آبادی
تاریخ: ۳:۳۰ ۱۳۹۲/۰۲/۱۲

با تشکر از نظر شما. تمام صحبت‌های شما منطقی است. اولاً من این بخش رو می‌خواستم بعداً تکمیل کنم ولی بدلاًیلی زودتر ارسال کردم.

مواردی هم که اعلام کردید کاملاً صحیح می‌باشد. فقط بخش تابعی که درون آن از try/catch استفاده شده رو می‌خواستم ابتدا در تیکه کد دیگری از یک صفحه aspx بنویسم که وقتی برای بررسی مجدد تابع پیدا نکرم و گرنه اونجا try/catch بلا استفاده است.

موارد مذکور شده برای سهولت بیشتر کاربران در کدهای فوق لحاظ و ویرایش گردید.
با تشکر.

نویسنده: محسن خان
تاریخ: ۱۱:۴ ۱۳۹۲/۰۲/۱۲

```
.locals init ([0] class [mscorlib]System.IO.TextWriter w)
IL_0000: ldstr      "log.txt"
IL_0005: call       class [mscorlib]System.IO.StreamWriter
            [mscorlib]System.IO.File::CreateText(string)
IL_000a: stloc.0
.try
{
    IL_000b: ldloc.0
```

نحوه کار با ftp - بخش اول

```
IL_000c: ldstr      "This is line one"
IL_0011: callvirt   instance void [mscorlib]
           System.IO.TextWriter::WriteLine(string)
IL_0016: leave.s    IL_0022
} // end .try
finally
{
IL_0018: ldloc.0
IL_0019: brfalse.s  IL_0021
IL_001b: ldloc.0
IL_001c: callvirt   instance void [mscorlib]
           System.IDisposable::Dispose()
IL_0021: endfinally
} // end handler
```

ماخذ

زمانیکه از using statement استفاده می‌کنید، [خود کامپایلر](#) try/finally close را اضافه می‌کنه. یعنی قسمت close الان بهش نیازی نیست چون استریم مورد استفاده حتما در اینجا dispose میشه.

نویسنده: بهمن آبادی
تاریخ: ۱۳۹۲/۰۲/۱۲ ۱۲:۵۰

از دستم در رفته ... شما ببخشید.

بیشتر هدفم روند بارگذاری و تکنیک بارگذاری اون بود ولی خوب درست کد نوشتن کاملً صحیحه و من می خواستم برای کاربرای آماتور بنویسم ولی مجبورم کردین که کاربرای حرفه ای هم مثل شما به کد ایرادی نباشه.

بازم ممنون از دقت و راهنماییتون .

حالا بهتر شد نسبت به قبلیه ؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۱۲ ۱۷:۲۸

من فکر می‌کنم اگر [ReSharper](#) رو نصب کنید، پیش از ارائه یک مطلب یا پروژه خیلی از ایرادات رو با خط کشیدن زیر اون یا نمایش یک علامت زرد کنار صفحه گوشزد می‌کنه. مثلا می‌گه که این نوع نامگذاری درست نیست یا این شء رو میشه با using محصور کرد. خلاصه از دستش ندید، حیفه !

نویسنده: سمیرا قادری
تاریخ: ۱۳۹۲/۰۶/۱۳ ۱۲:۳۷

با تشکر از مطلب خوبتان ببخشید این روش کار کردن با Ftp چه مزایایی دارد و چه کاربردی دارد؟ چون من تا حالا فقط از Cute Ftp استفاده کردم آن هم برای Upload سایت بعد از Publish

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۶/۱۳ ۱۲:۵۱

یه خورده وقت بدزاری، باهاش می‌تونی یک Cute Ftp بنویسی.

نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۱۲/۲۰ ۱۵:۵۰

با تشکر از مطلب خوبتون ،

برای اینکه به طور مستقیم از file.SaveAs استفاده کنیم و فایل را مستقیم به Ftp آپلود کنیم (نه اینکه از فضای ذخیره شده ای کپی کنیم) چه پیشنهادی می دهید ؟ مثلا ما هاست دانلود داریم و میخواهیم فایل های ارسالی را در هاست دانلود ذخیره کنیم.



 Add Note

می‌خواهیم با تغییر [jQuery Image Annotation](#) این پلاگین و برای asp.net استفاده کنیم

ایجاد دیتابیس

ابتدا یک دیتابیس به نام Coordinates ایجاد کنید و سپس جدول زیر را ایجاد کنید

```
USE [Coordinates]
GO
CREATE TABLE [dbo].[Coords2](
[top] [int] NULL,
[left] [int] NULL,
[width] [int] NULL,
[height] [int] NULL,
[text] [nvarchar](50) NULL,
[id] [uniqueidentifier] NULL,
[editable] [bit] NULL
) ON [PRIMARY]
GO
```

ایجاد کلاس Coords برای خواندن و ذخیره اطلاعات

```

public class Coords
{
    public string top;
    public string left;
    public string width;
    public string height;
    public string text;
    public string id;
    public string editable;

    public Coords(string top, string left, string width, string height, string text, string id, string
editable)
    {
        this.top = top;
        this.left = left;
        this.width = width;
        this.height = height;
        this.text = text;
        this.id = id;
        this.editable = editable;
    }
}

```

فرم اصلی برنامه شامل 3 وب سرویس به شرح زیر می‌باشد

1-GetDynamicContext

این متده در زمان لود اطلاعات از دیتابیس استفاده می‌شود (وقتی که postback صورت می‌گیرد)

```

[WebMethod]
public static List<Coords> GetDynamicContext(string entryId, string entryName)
{
    List<Coords> CoordsList = new List<Coords>();

    string connect = "Connection String";
    using (SqlConnection conn = new SqlConnection(connect))
    {
        string query = "SELECT [top], [left], width, height, text, id, editable FROM  Coords2";
        using (SqlCommand cmd = new SqlCommand(query, conn))
        {
            conn.Open();
            using (SqlDataReader reader=cmd.ExecuteReader())
            {
                while (reader.Read())
                {
                    CoordsList.Add(new Coords(reader["top"].ToString(), reader["left"].ToString(),
reader["width"].ToString(),
reader["height"].ToString(),
reader["text"].ToString(), reader["id"].ToString(),
reader["editable"].ToString()));
                }
            }
            conn.Close();
        }
    }
    return CoordsList;
}

```

DeleteCoords و SaveCoords- 2-3

این دو متده هم واسه ذخیره و حذف می‌باشند که نکته خاصی ندارند و خودتون بهینه اش کنید(در فایل ضمیمه موجودند)

تغییر فایل jquery.annotate.js جهت فراخوانی وب سرویس ها
فقط لازمه که سه قسمت زیر رو در فایل اصلی تغییر بدید

```

$.fn.annotateImage.ajaxLoad = function (image) {
    ///<summary>
    ///Loads the annotations from the "getUrl" property passed in on the
    ///    options object.
    ///</summary>

    $.ajax({
        type: "POST",
        contentType: "application/json; charset=utf-8",
        url: "Default.aspx/GetDynamicContext",
        data: "{ 'entryId': '" + 1 + "','" + entryName: '" + 2 + "'}",
        dataType: "json",
        success: function (msg) {
            image.notes = msg.d;
            $.fn.annotateImage.load(image);
        }
    });
};

$.fn.SaveCoords = function (note) {
    $.ajax({
        type: "POST",
        contentType: "application/json; charset=utf-8",
        url: "Default.aspx/SaveCoords",
        data: "{ 'top': '" + note.top + "','" + left': '" + note.left + "','" + width': '" + note.width +
        "','" + height': '" + note.height + "','" + text': '" + note.text + "','" + id': '" + note.id + "','" + editable': '" +
        note.editable + "'}",
        dataType: "json",
        success: function (msg) {
            note.id = msg.d;
        }
    });
};

$.fn.annotateView.prototype.edit = function () {

    ///<summary>
    ///Edits the annotation.
    ///</summary>

    if (this.image.mode == 'view') {
        this.image.mode = 'edit';
        var annotation = this;
        // Create/prepare the editable note elements
        var editable = new $.fn.annotateEdit(this.image, this.note);
        $.fn.annotateImage.createSaveButton(editable, this.image, annotation);
        // Add the delete button
        var del = $('<a>حذف</a>');
        del.click(function () {
            var form = $('#image-annotate-edit-form form');
            $.fn.annotateImage.appendPosition(form, editable)
            if (annotation.image.useAjax) {

```

```
$ajax({  
    type: "POST",  
    contentType: "application/json; charset=utf-8",  
    url: "Default.aspx/DeleteCoords",  
    // url: annotation.image.deleteUrl,  
    // data: form.serialize(),  
    data: "{ 'id': '" + editable.note.id + "' }",  
    dataType: "json",  
    success: function (msg) {  
        // image.notes = msg.d;  
        // $.fn.annotateImage.load(image);  
    },  
    error: function (e) { alert("An error occurred deleting that note.") }  
});  
}  
annotation.image.mode = 'view';  
editable.destroy();  
annotation.destroy();  
});  
editable.form.append(del);  
$.fn.annotateImage.createCancelButton(editable, this.image);  
}  
};
```

این پروژه شامل یه سری فایل css هم هست که می‌توانید کل پروژه رو از [اینجا](#) دانلود کنید

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۷ ۰:۲۲

ضمن تشکر، فقط نکته استفاده از `JSON.stringify` در حین کار با `jQuery Ajax` را بهتره اعمال کنید تا در دراز مدت و حالت‌های مختلف ورودی به مشکل برخورید. به صورت خلاصه اطلاعات ارسالی را جمع نزنید و تبدیل به رشته نکنید. یک [شیء کامل](#) درست کنید و اجازه بدید `JSON.stringify` اون را تبدیل کنه.

نویسنده: یزدان
تاریخ: ۱۳۹۲/۰۳/۱۸ ۱۰:۳۵

اگر متدهای وب سرویس رو درون صفحاتی قرار بدیم که نیاز به لاجین کردن و احراز هویت جهت دسترسی به آنها باشد ، آیا میتوان متدهای وب سرویس رو خارج از اون صفحات فراخوانی نمود ؟
چه تضمینی وجود دارد که در خارج از صفحات با سطح دسترسی (حداقل لاجین) وب سرویسها فراخوانی نشوند ؟
به نظر شما MS AJAX کارش به پایان رسیده ؟
تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۸ ۱۰:۵۴

- اگر دقت کرده باشید در کدهای فوق این متدها استاتیک تعریف شدن، یعنی مراحل چرخه طول عمر یک صفحه به آنها اعمال نشده و اصلاً جزئی از مباحث اعتبارسنجی صفحه جاری لحاظ نخواهد شد.
- در وب فرم‌ها استفاده از وب متدها یک روش برای کار با `jQuery Ajax` است. روش دوم استفاده از `Generic handler` ها و فایل‌های `ashx` است. در این موارد به علت استاتیک نبودن `handler` های تولیدی، می‌شود همه نوع اعتبارسنجی را اعمال کرد اعم از روش `Forms Authentication` یا `context.Request.IsAuthenticated` مثلاً توسط `IRequiresSessionState` برای اعتبارسنجی با پیاده سازی `.IsAuthenticated` از سشن‌ها.
- در مطلب فوق اصلاً از MS Ajax استفاده نشده. اون هم جایگاه خودش رو در کاربردهای خاص خودش دارد.

Cookie - قسمت اول : مقدمه، تاریخچه، معرفی، و شرح کامل

Cookie - قسمت دوم : کوکی در جاوا اسکریپت

نکته مهم: خواندن قسمت‌های قبلی این سری (مخصوصا [قسمت اول](#)) برای درک بهتر مطالب پیشنهاد می‌شود.

کوکی در ASP.NET - بخش اول



در قسمت‌های قبلی مقدمات و مباحثت کلی راجع به کوکی‌ها و انواع آن، شرح کامل خواص، نحوه رفتار مرورگرها با انواع کوکی‌ها و درنهایت نحوه کار کردن با کوکی‌ها در سمت کلاینت با استفاده از زبان محبوب جاوا اسکریپت پرداخته شد.

در ادامه این سری مطالب به نحوه برخورد ASP.NET با کوکی‌ها و چگونگی کار کردن با کوکی در سمت سرور آشنا خواهیم شد. در بخش اول این قسمت مباحثت ابتدایی و اولیه برای کار با کوکی‌ها در ASP.NET ارائه می‌شود. در بخش دوم مباحثت پیشرفته‌تر همچون SubCookie‌ها در ASP.NET و نیز سایر نکات ریز کار با کوکی‌ها در ASP.NET بحث خواهد شد.

ASP.NET در Request و Response

در قسمت اول این سری به مفاهیم **Http Request** و **Http Response** اشاره کوتاهی شده بود. به صورت خلاصه، درخواستی که از

سمت یک کلاینت به یک وب سرور ارسال می‌شود **Request** و پاسخی که وب سرور به آن درخواست می‌دهد **Response** نامیده می‌شود.

در ASP.NET، کلیه اطلاعات مرتبط با درخواست رسیده از سمت یک کلاینت در نمونه‌ای منحصر به فرد از کلاس **HttpRequest** نگه‌داری می‌شود. محل اصلی نگه‌داری این نمونه در پراپرتی **Request** از نمونه جاری کلاس **System.Web.HttpContext** (قابل دسترسی از طریق **HttpContext.Current**) است. البته کلاس **Page** هم یک پراپرتی با نام **Request** دارد که دقیقاً از همین پراپرتی کلاس **HttpContext** استفاده می‌کند.

هم‌چنان کلیه اطلاعات مرتبط با پاسخ ارسالی وب سرور به سمت کلاینت مربوطه در نمونه‌ای از کلاس **HttpResponse** ذخیره می‌شود. محل اصلی نگه‌داری این نمونه نیز در پراپرتی **Response** از نمونه جاری کلاس **HttpContext** است. همانند **Request**، کلاس **Page** یک پراپرتی با نام **Response** برای نگه‌داری این نمونه دارد که این هم دقیقاً از پراپرتی متناظر در کلاس **HttpContext** استفاده می‌کند.

کوکی‌ها در Request و Response

هر دو کلاس **HttpRequest** و **HttpResponse** یک پراپرتی با عنوان **Cookies** (^ و ^) دارند که مخصوص نگهداری کوکی‌های مربوطه هستند. این پراپرتی از نوع **System.Web.HttpCookieCollection** است که یک کالکشن مخصوص برای ذخیره کوکی‌هاست.

- این پراپرتی (**Cookies**) در کلاس **HttpRequest** محل نگهداری کوکی‌های ارسالی توسط مرورگر در درخواست متناظر آن است. کوکی‌هایی که مرورگر با توجه به شرایط جاری و تنظیمات کوکی‌ها اجازه ارسال به سمت سرور را به آن‌ها داده و در درخواست ارسالی ضمیمه کرده است (با استفاده از هدر: **Cookie** که در [قسمت اول](#) شرح داده شد) و پس از پردازش و **Parse** داده‌ها، درون این پراپرتی اضافه کرده است.

- این پراپرتی (**Cookies**) در کلاس **HttpResponse** محل ذخیره کوکی‌های ارسالی از وب سرور به سمت مرورگر کلاینت در پاسخ به درخواست متناظر است. کوکی‌های درون این پراپرتی پس از بررسی و استخراج داده‌های موردنیاز توسط ASP.NET در هدر **Set-Cookie**: (با استفاده از هدر: **Set-Cookie** که در [قسمت اول](#) توضیح داده شد).

ایجاد و بروزرسانی کوکی در ASP.NET

برای ایجاد یک کوکی و ارسال آن به سمت کلاینت همان‌طور که در بالا نیز اشاره شد، باید از پراپرتی **Cookies** از کلاس **HttpContext** استفاده کرد. برای ایجاد یک کوکی روش‌های مختلفی وجود دارد. در این روش، ابتدا بررسی می‌شود که کوکی موردنظر در لیست کوکی‌های جاری وجود دارد یا خیر. در صورتی که با این نام قبلاً یک کوکی ثبت شده باشد، مقدار کوکی موجود بروزرسانی خواهد شد. اما اگر این نام وجود نداشته باشد یک کوکی جدید با این نام به لیست افزوده شده و مقدار آن ثبت می‌شود. مثال:

```
HttpContext.Current.Response.Cookies["myCookie"].Value = "myCookieValue";
```

روش بعدی استفاده از متد [Add](#) در کلاس [HttpCookieCollection](#) است. در این روش ابتدا یک نمونه از کلاس [HttpCookie](#) ایجاد شده و سپس این نمونه به لیست کوکی‌ها اضافه می‌شود. کد زیر چگونگی استفاده از این روش را نشان می‌دهد:

```
var myCookie = new HttpCookie("myCookie", "myCookieValue");
HttpContext.Current.Response.Cookies.Add(myCookie);
```

روش دیگر استفاده از متد [Set](#) کلاس [HttpCookieCollection](#) است. تفاوت این متد با متد [Add](#) در این است که متد [Set](#) ابتدا سعی می‌کند عملیات [update](#) انجام دهد. یعنی عملیات افزودن تنها وقتی که نام کوکی موردنظر در لیست کوکی‌ها یافته نشود انجام خواهد شد. برای مثال:

```
HttpContext.Current.Response.Cookies.Set(new HttpCookie("myCookie", "myCookieValue"));
```

نکته: با توجه به توضیحات بالا، متد [Set](#) اجازه افزودن دو کوکی با یک نام را نمی‌دهد. برای اینکار باید از متد [Add](#) استفاده کرد. درباره این موضوع در قسمت بعدی بیشتر توضیح داده خواهد شد. روش دیگری که برای ایجاد یکی کوکی می‌توان از آن استفاده کرد، بکارگیری متد [AppendCookie](#) از کلاس [HttpResponse](#) است. در این روش نیز ابتدا باید یک نمونه از کلاس [HttpCookie](#) تولید شود. این روش همانند استفاده از متد [Add](#) از کلاس [HttpCookieCollection](#) است. کد زیر مثالی از این روش را نشان می‌دهد:

```
HttpContext.Current.Response.AppendCookie(new HttpCookie("myCookie", "myCookieValue));
```

روش بعدی استفاده از متد [SetCookie](#) از کلاس [HttpResponse](#) است. فرق این متد با متد [AppendCookie](#) در این است که در متد [SetCookie](#) ابتدا وجود یک کوکی با نام ارائه شده بررسی می‌شود و در صورت وجود، مقدار این کوکی بروزرسانی می‌شود. در صورتی که قبل از این نام وجود نداشته باشد، یک کوکی جدید به لیست کوکی‌ها اضافه می‌شود. این روش همانند استفاده از متد [Set](#) از کلاس [HttpCookieCollection](#) است. نمونه‌ای از نحوه استفاده از این متد در زیر آورده شده است:

```
HttpContext.Current.Response.SetCookie(new HttpCookie("myCookie", "myCookieValue"));
```

نکته: تمامی فرایندهای نشان داده شده در بالا تنها موجب تغییر محتویات کالکشن کوکی‌ها درون [HttpContext](#) می‌شود و تا زمانی که توسط وب سرور با استفاده از دستور [Set-Cookie](#) به سمت مرورگر ارسال نشوند تغییری در کلاینت بوجود نخواهد آورد.

برای آشنایی بیشتر با این روند کد زیر را برای تعریف یک کوکی جدید درنظر بگیرید:

```
HttpContext.Current.Response.Cookies["myCookie"].Value = "myValue";
```

برای مشاهده هدر تولیدی توسط وب سرور می‌توان از نرم افزار محبوب [Fiddler](#) استفاده کرد (از اواخر سال 2012 که نویسنده این ابزار به [Telerik](#) پیوسته، توسعه آن بسیار فعال تر شده و نسخه‌های جدید با لوگوی جدید! ارائه شده است). تصویر زیر مربوط به مثال بالاست:

The screenshot shows the Fiddler tool interface with the 'Response Headers' tab selected. The response is an 'HTTP/1.1 200 OK'. The 'Cookies / Login' section contains the highlighted 'Set-Cookie: myCookie=myValue; path=/'. A red arrow points from the text 'همانطور که مشاهده می‌کنید دستور ایجاد یک کوکی با نام و مقدار وارد در هدر پاسخ تولیدی توسط وب سرور گنجانیده شده است.' to this line.

Get SyntaxView	Transformer	Headers	To
Response Headers			
HTTP/1.1 200 OK			
Cache			
Cache-Control: private			
Date: Fri, 14 Jun 2013 16:53:10 GMT			
Cookies / Login			
Set-Cookie: myCookie=myValue; path=/			
Entity			
Content-Length: 2034			
Content-Type: text/html; charset=utf-8			
Miscellaneous			
Server: ASP.NET Development Server/10.0.0.0			
X-AspNet-Version: 4.0.30319			
Transport			
Connection: Close			

همانطور که مشاهده می‌کنید دستور ایجاد یک کوکی با نام و مقدار وارد در هدر پاسخ تولیدی توسط وب سرور گنجانیده شده است.

نکته: در ASP.NET به صورت پیش فرض از مقدار "/" برای پراپرتی Path استفاده می‌شود.

خواص کوکی در ASP.NET

برای تعیین یا تغییر خواص یک کوکی در ASP.NET باید به نمونه `HttpCookie` مربوطه دست یافت. سپس با استفاده از پراپرتی‌های این کلاس می‌توان خواص موردنظر را تعیین کرد. برای مثال:

```
var myCookie = new HttpCookie(string.Empty);
myCookie.Name = "myCookie";
myCookie.Value = "myCookieValue";
myCookie.Domain = "dotnettipp.info";
myCookie.Path = "/post";
myCookie.Expires = new DateTime(2015, 1, 1);
myCookie.Secure = true;
myCookie.HttpOnly = true;
```

نکته مهم: امکان تغییر خواص یک کوکی به صورت مستقیم در سمت سرور وجود ندارد. درواقع برای اعمال این تغییرات در سمت کلاینت باید به ازای هر کوکی موردنظر یک کوکی جدید با مقادیر جدید ایجاد و به کالکشن کوکی‌ها در `Http Response` مربوطه اضافه شود تا پس از قرار دادن دستور `Set-Cookie` متضاظر در هدر پاسخ ارسالی به سمت کلاینت و اجرای آن توسط مرورگر، مقادیر خواص موردنظر در سمت کلاینت بروزرسانی شوند. دقت کنید که تمامی نکات مرتبط با هویت یک کوکی که در [قسمت اول](#) شرح داده شد در اینجا نیز کاملاً صادق است.

روش دیگری نیز برای تعیین برخی خواص کوکی‌ها به صورت کلی در فایل وب کانفیگ وجود دارد. برای اینکار از تگ `httpCookies`

در قسمت system.web استفاده می‌شود. برای مثال:

```
<httpCookies domain="www.example.com" httpOnlyCookies="true" requireSSL="true" />
```

این امکان از ASP.NET 2.0 به بعد اضافه شده است. با استفاده از این تگ، تنظیمات اعمال شده برای تمامی کوکی‌ها در نظر گرفته می‌شود. البته در صورتی که تنظیم موردنظر برای کوکی به صورت صریح آورده نشده باشد. برای نمونه به کد زیر دقت کنید:

```
var myCookie = new HttpCookie("myCookie", "myCookieValue");
myCookie.Domain = "test.com";
HttpContext.Current.Response.Cookies.Add(myCookie);
var myCookie2 = new HttpCookie("myCookie2", "myCookieValue2");
myCookie2.HttpOnly = false;
myCookie2.Secure = false;
HttpContext.Current.Response.Cookies.Add(myCookie2);
```

با استفاده از تنظیمات تگ httpCookies که در بالا نشان داده شده است، هدر پاسخ تولیدی توسط وب سرور به صورت زیر خواهد بود:

Get SyntaxView	Transformer	Headers	TextView	ImageView	HexView
Response Headers					
HTTP/1.1 200 OK					
Cache					
Cache-Control: private					
Date: Fri, 14 Jun 2013 17:01:46 GMT					
Cookies / Login					
Set-Cookie: myCookie2=myCookieValue2; domain=www.example.com; path=/					
Set-Cookie: myCookie=myCookieValue; domain=test.com; path=/; secure; HttpOnly					
Entity					

همان‌طور که می‌بینید تنها مقادیر پر اپریتی‌هایی که صراحتاً برای کوکی آورده نشده است از تنظیمات وب کانفیگ خوانده می‌شود.

حذف کوکی در ASP.NET

برای حذف یک کوکی در ASP.NET یک روش کلی وجود دارد که در قسمت‌های قبلی نیز شرح داده شده است، یعنی تغییر خاصیت Expires کوکی به تاریخی در گذشته. برای نمونه داریم:

```
var myCookie = new HttpCookie("myCookie", "myCookieValue");
myCookie.Expires = DateTime.Now.AddYears(-1);
```

نکته مهم: در کلاس HttpCookieCollection یک متده با نام Remove وجود دارد. از این متده برای حذف یک کوکی از لیست موجود

در این کلاس استفاده می‌شود. دقت کنید که حذف یک کوکی از لیست کوکی‌ها با استفاده از این متاد تاثیری بر موجودیت آن کوکی در سمت کلاینت نخواهد گذاشت و تنها روش موجود برای حذف یک کوکی در سمت کلاینت همان تنظیم مقدار خاصیت `Expires` است.

خواندن کوکی در ASP.NET

برای خواندن مقدار یک کوکی ارسالی از مرورگر کلاینت در ASP.NET، باتوجه به توضیحات ابتدای این مطلب، طبیعی است که باید از پرآپرتی `Request.Cookies` در نمونه جاری از کلاس `HttpContext` استفاده کرد. برای این کار نیز چند روش وجود دارد. روش اول استفاده از `aindexer` کلاس `HttpCookieCollection` است. برای اینکار نیاز به نام یا ایندکس کوکی موردنظر در لیست مربوطه داریم. برای مثال:

```
var myCookie = HttpContext.Current.Request.Cookies["myCookie"];
```

یا این نمونه با استفاده از `aindexer عددی`:

```
var myCookie = HttpContext.Current.Request.Cookies[0];
```

روش دیگری که برای خواندن مقدار یک کوکی می‌توان بکار برد، استفاده از متاد `Get` از کلاس `HttpCookieCollection` است. این متاد همانند `aindexer` این کلاس نیاز به نام یا ایندکس کوکی موردنظر دارد. برای نمونه:

```
var myCookie = HttpContext.Current.Request.Cookies.Get("myCookie");
```

یا استفاده از ایندکس کوکی:

```
var myCookie = HttpContext.Current.Request.Cookies.Get(0);
```

بحث و نتیجه گیری

تا اینجا با مفاهیم اولیه درباره نحوه برخورد ASP.NET با کوکی‌ها آشنا شدیم. روش‌های مختلف ایجاد و یا بهروزرسانی کوکی‌ها نشان داده شد. با تعیین انواع خواص کوکی‌ها آشنا شدیم. نحوه حذف یک کوکی در ASP.NET را دیدیم. روش‌های خواندن مقادیر کوکی‌ها را نیز مشاهده کردیم.

باز هم تأکید می‌کنم که تمامی تغییرات اعمالی در سمت سرور تا زمانی که به صورت دستورات `Set-Cookie` در هدر پاسخ وب سرور قرار نگیرند هیچ کاری در سمت کلاینت انجام نمی‌دهند.

در قسمت بعدی این سری مطالب به مباحث پیشرفته‌تری چون SubCookie و هویت منحصر به فرد کوکی‌ها در سمت سرور پرداخته می‌شود.

منابع

[http://msdn.microsoft.com/en-us/library/ms178194\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/ms178194(v=vs.100).aspx)

[http://msdn.microsoft.com/en-us/library/aa289495\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/aa289495(v=vs.71).aspx)

<http://www.codeproject.com/Articles/31914/Beginner-s-Guide-To-ASP-NET-Cookies>

<http://www.codeproject.com/Articles/244904/Cookies-in-ASP-NET>

نظرات خوانندگان

نویسنده: ایمان اسلامی
تاریخ: ۱۸:۱۸ ۱۳۹۳/۰۸/۰۶

یک کوکی ایجاد کردم و یه سری دیتا درون اون قرار دادم. در جایی از سایت برای پرداخت اینترنتی به درگاه بانک متصل میشم اما بعد از برگشت از بانک ، وقتی میخوام مقدار کوکی رو بخونم ، کوکی مقدارش خالیه. حتی دستور `Page.User.Identity.IsAuthenticate` هم مقدار `false` بر می‌گردونه. در صورتی که قبل از ارسال به درگاه، کاربر لاغین بوده. ممنون میشم راهنمایی کنید.

نویسنده: وحید نصیری
تاریخ: ۱۰:۹ ۱۳۹۳/۰۸/۰۷

در دات نت 4.5 ، مشکل طولانی بودن حاصل `BinaryFormatter serialization` برطرف شده (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: ایمان اسلامی
تاریخ: ۱۲:۳۰ ۱۳۹۳/۰۸/۰۷

با تشکر؛ فقط یک نکته تکمیلی که فراموش کردم اینکه مشکل مورد نظر مربوط به `asp.net web form simple membership` برای فرآیند احراز هویت استفاده میکنم. با توجه به این مسائل ، انجام مواردی که شما فرمودید برای حل مشکل کفایت میکنه؟

نویسنده: وحید نصیری
تاریخ: ۱۲:۳۵ ۱۳۹۳/۰۸/۰۷

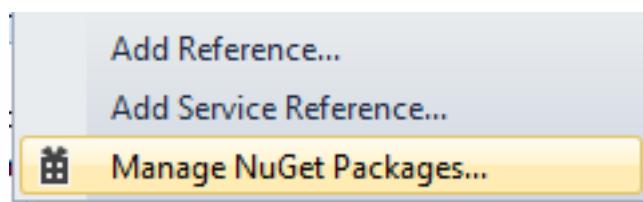
فرقی نمی‌کند. مباحثت پایه `Forms Authentication` برای تمام فناوری‌هایی که از آن استفاده می‌کنند یکسان است.

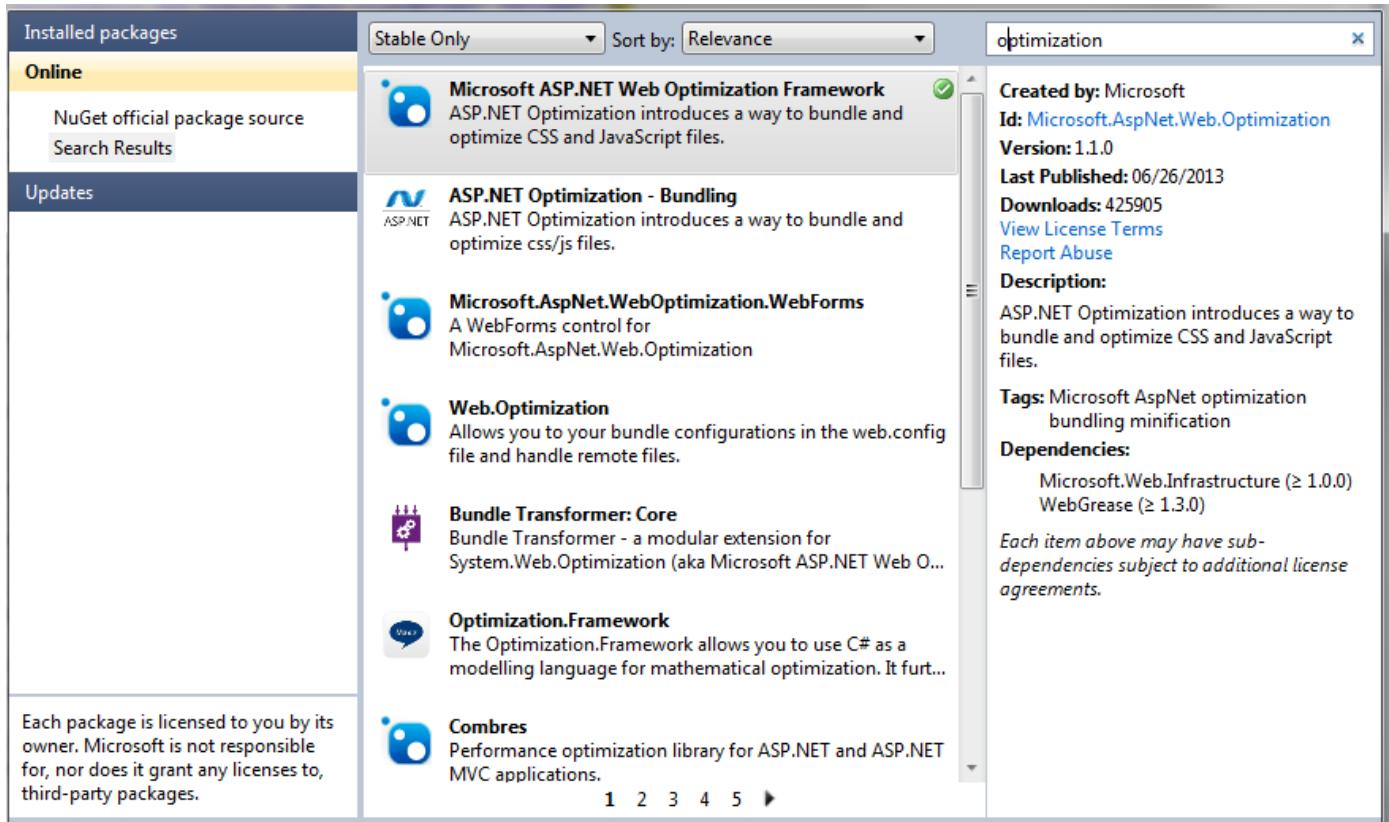
چند روز پیش تصمیم گرفتم از ASP.NET FriendlyUrls برای سفارشی کردن Url های سایت استفاده کنم

وقتی در لوکال تست میکردم همه چیز درست بود و بدون مشکل کار میکرد اما به محض اینکه سایت رو روی سرور Publish کردم و قصد پیمایش بین صفحات رو داشتم، با کلیک بر روی منوها با خطا 404 not found مواجه میشدم!

پس از کمی بررسی متوجه شدم که باید اسمبلی System.Web.Optimization.dll رو به وب سایت اد کنم و چند خط کد به web.config سایت اضافه کنم.

برای اضافه کردن 1- System.Web.Optimization.dll به سایت روی References کلیک راست کنید و گزینه Microsoft ASP.NET Web Optimization Packages رو انتخاب کنید و سپس با جستجوی عبارت optimize از آیتم های پیدا شده Framework رو نصب کنید.





2- خطوط زیر رو به web.config سایت اضافه کنید

```
<system.webServer>
<modules runAllManagedModulesForAllRequests="true">
<remove name="BundleModule" />
<add name="BundleModule" type="System.Web.Optimization.BundleModule" />
</modules>
</system.webServer>
```

با این تغییرات ایجاد ASP.NET FriendlyUrls برطرف شد

نظرات خوانندگان

نویسنده: محسن تقی پور
تاریخ: ۱۳۹۲/۱۱/۰۴ ۱۸:۵۷

سلام

من همین کاریو که شما فرمودینو انجام دادم ولی باز 404 میده
امکان داره دلیل دیگه ای داشته باشه ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۴ ۱۹:۲۴

- شخصی در [نظرات بحث معرفی Friendly URLs](#) عنوان کرده با نصب یک [hot fix](#) روی سرور مشکل خطای 404 آن حل شده.
- همچنین نصب [URL Rewrite extension](#) روی IIS هم ضروری است.

نویسنده: مجتبی صحرائی
تاریخ: ۱۳۹۲/۱۱/۰۴ ۲۰:۲۹

سلام

با توجه به بررسی هایی که انجام داده بودم، با استفاده از روش ذکر شده مشکلم برطرف شد و بیشتر از اون هم جستجو نکردم
در حال حاضر نظری ندارم و پیشنهاد میکنم توصیه آقای نصیری رو انجام بدید

نویسنده: محسن تقی پور
تاریخ: ۱۳۹۲/۱۱/۰۴ ۲۱:۴۳

ممnon از پاسخ دوستان

بله طبق بررسی بنده هاست من [URL Rewrite extension](#) نداره

امروز از ASP.NET FriendlyUrls که دوست عزیزمون به اشتراک گذاشتن استفاده می‌کردم ([اینجا](#)) و در صفحه اول سایتم مجبور بودم با استفاده از JSON یک متاد را از صفحه Default.aspx صدابزنم که کد زیر را نوشتم:

```
$ajax({
    type: "POST",
    url: "/Default.aspx/MyMethod",
    data: "{}",
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    async: true,
    cache: true,
    success: function (data) {
        $("#plandata").html(data.d);
    },
    error: function (x, e) {
        alert("The call to the server side failed. " + x.responseText);
    }
});
```

هنگام کار، alert ای بر روی صفحه ظاهر می‌شد و خطای Unauthorized 401 را نمایش می‌داد. حتی آدرس را به صورت امتحان کردم باز هم خطا داد و متاد را شناسایی نمی‌کرد. به جای این کار کد زیر را در web.config اضافه کردم:

```
<system.web.extensions>
  <scripting>
    <webServices>
      <authenticationService enabled="true" />
    </webServices>
  </scripting>
</system.web.extensions>
```

با استفاده از راه حل مطلب [اینجا](#)، به جای صفحه aspx یک صفحه WebService به پروژه اضافه و متدها را به آن جا منتقل کردم. نتیجه کد به شرح زیر شد:

```
$ajax({
    type: "POST",
    url: "/websrv.asmx/MyMethod",
    data: "{}",
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    async: true,
    cache: true,
    success: function (data) {
        $("#plandata").html(data.d);
    },
    error: function (x, e) {
        alert("The call to the server side failed. " + x.responseText);
    }
});
```

و خدا را شکر مشکل برطرف شد.

عنوان: رفع مشکل postBack در Master Page و دریافت خطای the object
نویسنده: حسین صدری
تاریخ: ۱۳۹۲/۰۵/۱۸
آدرس: www.dotnettips.info
برچسبها: ASP.Net, Tips, Handle Errors

با این خطا هنگامی مواجه شدم که در Master Page یک TextBox و یک دکمه قرار داده بودم و در رویداد دکمه به یک صفحه دیگر انتقال پیدا می‌کردم. کد زیر:

```
protected void imgBtnSearch_Click(object sender, ImageClickEventArgs e)
{
    if (TxtbSearch.Text != string.Empty)
    {
        Response.Redirect("/Exhibition>ShowSearch.aspx?title=" +
            Sanitizer.GetSafeHtmlFragment(TxtbSearch.Text) + "&b=true", false);
    }
}
```

این کد در ماشین خودم کار می‌کرد و مشکلی نداشت. وقتی پروژه را به سرور انتقال می‌دادم و بر روی IIS 6 تست می‌کردم خطای 404 دریافت می‌کردم. ضمن اینکه به صفحه خواسته شده انتقال پیدا نمی‌کرد، و همه چیز کاملاً گیج کننده بود. وقتی صفحه را از خود سرور باز و رویداد فوق را اجرا می‌کردم خطای Operation is not valid due to the current state of the object دریافت می‌کردم به همراه کد زیر در Track trace به شرح زیر:

```
at System.Web.HttpValueCollection.ThrowIfMaxHttpCollectionKeysExceeded()
at System.Web.HttpValueCollection.FillFromEncodedBytes(Byte[] bytes, Encoding encoding)
at System.Web.HttpRequest.FillInFormCollection()
```

با افزودن کد زیر در Web.config مشکل برطرف گردید.

```
<appSettings>
  <add key="aspnet:MaxHttpCollectionKeys" value="2001" />
</appSettings>
```

منابع:

[Code Project](#)
[stack overflow](#)

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۵/۰۸ ۱۸:۴۷

این مشکل زمانی به وجود میاد که تعداد انبوهی کنترل روی صفحه داشته باشد. ساختار صفحه شما به چه صورتی است؟ (کابین کنترل خلبان؟!)

نویسنده: حسین صدری
تاریخ: ۱۳۹۲/۰۵/۰۸ ۱۹:۲

D: نه بابا!
گفتم در master Page و صفحه اول سایت شامل منو، ۳ تا repatear (تبليغات، اخبار) و چندين باکس ديگه...

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۲/۰۵/۰۸ ۲۰:۳۸

خيلي ممنون بابت اين نکته.
ولي ديگه سعى کنيد که کوچ کنيد به MVC. تو وب فرم آدم نمیدونه تو وب هست يا ويندوز فرم. يك وب مصنوعي.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۵/۰۸ ۲۰:۴۶

مشکل وب فرماها اينه که کل صفحه رو يك فرم درنظر ميگيره. بنابراین موقع postback به سرور تمام صفحه رو ارسال میکنه که ممکنه سنگين باشه. در يك صفحه چندتا فرم داشته باشيد با محدودههای عملکرد کوچکتر.

نویسنده: حسین صدری
تاریخ: ۱۳۹۲/۰۵/۰۸ ۲۲:۷

چشم، مرسی نکات زیبایی دوستان گفتند، ولی راستش این پروژه برای ۲ سال پیش بود، امروز به دلایلی یه گزینه جستجو اضافه کردم و باقی ماجرا (-)

در این سلسله مقالات قصد دارم چندین مطلب راجع به افزایش سرعت نرم افزارهای تحت وب مطرح نمایم. این مطالب هر چند بسیار مختصر می‌باشند ولی در کارایی و سرعت برنامه‌های شما در آینده تاثیر خواهند داشت.

1. کش کردن همیشه آخرین حربه می‌باشد

این مهم است که بخش‌های مختلف سایت شما در سطح مختلف کش شوند (ASP.NET, Kernel, Server, Proxy Server, Browser,...). ولی این موضوع باید همیشه آخرین حربه و نکته ای باشد که آن را در مورد سایت خود اعمال می‌کنید. یعنی همیشه مطمئن شوید ابتدا تمامی نکات مربوط به افزایش کارایی در برنامه خود را رعایت کرده اید، سپس اقدام به کش داده‌ها در سطوح مختلف نمایید. توجه کنید کش کردن داده‌ها و صفحات می‌تواند مشکلات را برای شما به عنوان یک برنامه نویس یا تست کننده برنامه پنهان کند و به شما اطمینان دهد که همه چیز خوب کار می‌کند در حالی که این چنین نیست! البته ذکر این نکته هم بی فایده نیست که کش کردن همه چیز بعضی مواقع دشمن بر نامه شما محسوب می‌شود! هیچ وقت یادم نمی‌رود، در پورتال داخلی یک شرکت که در وقت استراحت به کارکنان اجازه مطالعه روزنامه‌های روز را می‌داد (به صورت آفلاین)، این نکته در بالای صفحه آورده شده بود: «لطفا برای به روز رساندن صفحات روزنامه‌ها از کلید Ctrl+F5 استفاده نمایید». این موضوع یعنی بحث کشینگ در برنامه آن پورتال در سطح فاجعه می‌باشد! حالا فرض کنید این مشکل در فرم ورود و یا مرور اطلاعات یک برنامه به وجود آید...

2. حذف View Engine های غیر ضروری

به عنوان یک برنامه نویس ASP.NET MVC، یابد اطلاع داشته باشید که CLR به صورت خودکار View Engine های Web Forms و Razor را لود می‌کند. این موضوع به این دلیل است که اطلاعی از نحوه برنامه نویسی شما ندارد. اگر شما فقط از یکی از این دو Engine استفاده می‌کنید، لطفاً دیگری را غیر فعال کنید! فعل بودن هر دوی آنها یعنی اتلاف وقت گرانهای CPU سرور شما برای رندر کردن تمامی صفحات شما توسط دو انجین! ایتدا view های شما با Web Forms Engine رندر شده سپس نتیجه به Razor Engine منتقل شده و مجدد توسط این انجین رندر می‌شود. این موضوع در سایت‌های با تعداد کاربر بالا یعنی فاجعه! برای حل این مشکل کافی است خطوط زیر را در فایل Global.asax و در رویداد بخش Application_Start وارد نمایید:

```
ViewEngines.Engines.Clear();
ViewEngines.Engines.Add(new RazorViewEngine());
```

این دو خط یعنی خدا حافظ ...Web Forms Engine

قبل از استفاده از این کد، اطمینان حاصل کنید کل برنامه شما توسط Razor تهیه شده است و گرنه بنده هیچ مسئولیتی در رابطه با فریادهای کارفرمای شما متقابل نمی‌شوم!

صد البته برای حذف Razor Engine و استفاده از Web Form Engine می‌توان از کد زیر در همان موقعیت فوق استفاده کرد:

```
ViewEngines.Engines.Clear();
ViewEngines.Engines.Add(new WebFormViewEngine());
```

البته همانطور که حتماً دوستان مطلع هستند امکان گسترش Engine های فوق توسط ارت بری از کلاس BuildManagerViewEngine جهت ایجاد Engine های دیگر همیشه ممکن است. در این صورت می‌توانید تنها انجین سفارشی مورد نظر خود را لود کرده و از لود دیگر انجین‌ها پرهیز کنید.

3. استفاده از فایل‌های PDB مایکروسافت برای دیباگ و یا پروفایل کردن DLL های دیگران

دیباگ یا پروفایل کردن برنامه ها، DLL ها، اسambilی‌ها و منابعی از برنامه که شما آن را خود ننوشته اید (سورس آنها در دسترس شما نمی‌باشد) همیشه یکی از سخت‌ترین مراحل کار می‌باشد. جهت کمک به دیباگرها یا پروفایلرها، نیاز است فایل‌های PDB مرتبط با DLL ها را در اختیار آنها قرار دهید تا به بهترین نتیجه دسترسی پیدا کنید. این فایل‌ها محتوى نام توابع، شماره خطوط برنامه و metadata های دیگر برنامه اصلی قبل از optimize شدن توسط کامپایلر یا JIT می‌باشد. خوب حالا اگر نیاز شد این کار را در رابطه با DLL ها و کلاس‌های پایه Microsoft.NET انجام دهیم چه کار کنیم؟

خیلی ساده! خود Microsoft سروری جهت این موضوع تدارک دیده که فایل‌های PDB را جهت دیباگ کردن در اختیار تیم‌های برنامه نویسی قرار می‌دهد. کافی است از منوی Tools گزینه Options را انتخاب، سپس به بخش Debugging و به بخش Symbols بروید و گزینه Microsoft Symbol Servers as your source for Symbols را انتخاب کنید. برای اطمینان از اینکه هر مرتبه که برنامه را دیباگ می‌کنید مجبور به دانلود این فایل‌ها نشوید، فراموش نکنید پوشه‌ای را جهت کش این فایل‌ها ایجاد و آدرس آن را در بخش Cache symbols in this directory همین صفحه وارد نمایید.

این امکان در Visual Studio 2012 در دسترس می‌باشد.

نظرات خوانندگان

نویسنده: محسن خان

تاریخ: ۱۴۰۵/۰۵/۱۰ ۱۳۹۲

اثر View Engine رو با [Glimpse](#) بهتر میشه دید.

قسمت اول

4. فشرده سازی HTTP را فعال کنید

اطمینان حاصل کنید که HTTP Compression در تمامی بخش‌های اصلی برنامه شما فعال است. حداقل کاری که می‌توانید در این رابطه بکنید این است که خروجی HTML که توسط برنامه شما تولید می‌شود را فشرده سازی کنید. جهت فعال سازی فشرده سازی در برنامه خود بهتر است در اولویت اول از ماژول ویژه‌ای که جهت این کار در IIS در نظر گرفته شده استفاده کنید. این ماژول تمامی کارها را به صورت خودکار برای شما انجام می‌دهد. اگر دسترسی به IIS جهت فعال سازی آن را ندارید، می‌توانید از ماژول‌های ASP.NET که جهت این کار تهیه شده استفاده کنید. می‌توانید [کمی جستجو کنید](#) و یا خودتان یکی تهیه کنید!

5. تنظیم CacheControlMaxAge

مقدار [CacheControlMaxAge](#) را در فایل web.config را طوری تنظیم کنید تا هیچ کاربری هیچ فایل static را دیگر درخواست نکند. مثلاً می‌توانید این مقدار را بر روی چند ماه تنظیم کنید و البته فراموش نکنید این مقدار را در صفحات پویای خود بازنویسی (override) کنید تا مشکلی در رابطه با کش شدن فرم‌های اصلی برنامه (همانطور که در نکته اول بخش اول ذکر شد) پدید نیاید. البته کش کردن فایل‌های استاتیک برنامه بار مالی نیز برای شما و کاربرانتان خواهد داشت. دیگر هزینه پهنانی باند اضافی جهت دانلود این فایل‌ها در هر درخواست برای شما (در سمت سرور) و کاربرانتان (در سمت کاربر) پرداخت خواهد شد!

6. استفاده از OutputCache

اگر از MVC استفاده می‌کنید، فراموش نکنید که از [OutputCache](#) در کنترل‌های MVC استفاده نمایید. اگر سرور شما بتواند اطلاعات را از رم خود بازیابی کند بهتر از آن است که آن را مجدد از دیتابیس واکشی نماید و عملیاتی نیز بر روی آن انجام دهد. البته مدیریت حافظه .NET به صورت خودکار کمبود حافظه را مدیریت کرده و از نشت حافظه جلوگیری خواهد کرد. برای توضیحات بیشتر در این رابطه می‌توانید از این [مقاله](#) کمک بگیرید.

7. بهره برداری از ORM Profiler

ORM Profiler ها تمامی فعالیت‌های ORM تحت نظر گرفته، دستورات T-SQL ارسالی به بانک اطلاعاتی را واکشی کرده و برای شما نمایش می‌دهند. [تعدادی از آنها](#) نیز این دستورات را آنالیز کرده پیشنهاداتی در رابطه با بهبود کارایی به شما ارائه می‌دهند. برای مثال به جای اینکه شما 2000 رکورد را یکی یکی از بانک بازیابی کنید، می‌توانید آن را به صورت یک query به بانک ارسال کنید. این موضوع به سادگی توسط ORM Profiler ها قابل بررسی است. نمونه ای از این نرم افزارها را می‌توانید در [این سایت](#) یا [این سایت](#) پیدا کنید. البته در صورتی که نمی‌خواهید از نرم افزارهای جانبی استفاده کنید، می‌توانید از ابزارهای توکار بانک‌های اطلاعاتی مانند [SQL Profiler](#) نیز استفاده کنید ([راهنمایی](#)).

قسمت دوم

ORM Lazy Load.8

در هنگام استفاده از ORM‌ها دقت کنید کجا از [Lazy Load](#) استفاده می‌کنید. Lazy Load باعث می‌شود وقتی شما اطلاعات مرتبط را از بانک اطلاعات واکشی می‌کنید، این واکشی اطلاعات در چند query از بانک انجام شود. در عوض عدم استفاده از [Lazy Load](#) باعث می‌شود تمامی اطلاعات مورد نیاز شما در یک query از بانک اطلاعاتی دریافت شود. این موضوع یعنی سربار کمتر در شبکه، در بانک اطلاعاتی، در منابع حافظه و منابع پر از رش cpu در سرورها. البته استفاده از include در حالت فعال بودن یا نبودن lazy هم داستان مجزایی دارد که اگر عمری باقی باشد راجع به آن مقاله‌ای خواهیم نوشت.
به این نمونه دقت کنید:

```
List<Customer> customers = context.Customers.ToList();
foreach (Customer cust in context.Customers){
    Console.WriteLine("Customer {0}, Account {1}", cust.Person.LastName.Trim() + ", " +
    cust.Person.FirstName, cust.AccountNumber);
}
```

همچین کدی (در صورت فعال بودن Lazy Load در ORM) در صورتی که جدول Customers دارای 1000 رکورد باشد، باعث می‌شود برنامه 1001 دستور sql تولید و در بانک اجرا گردد.
برای اطلاع بیشتر می‌توانید به این [مقاله](#) مراجعه نمایید.

9. استفاده از MiniProfiler

سعی کنید از MiniProfiler در تمامی پروژه‌ها استفاده کنید. البته وقتی نرم افزار را در اختیار مصرف کننده قرار می‌دهید، آن را غیر فعال کنید. می‌توانید از متغیرهای compiler برای مجزا کردن build‌های متفاوت در برنامه خود استفاده کنید:

```
#if DEBUG then
    // MiniProfiler
#endif
```

ایده دیگری هم وجود دارد. شما می‌توانید MiniProfiler یا کاربر Admin را برای Debugger فعال و برای بقیه غیر فعال کنید. در باب MiniProfiler مسائل زیادی وجود دارد که چند نمونه از آن در همین سایت در [این مقاله](#) و [این مقاله](#) در دسترس است. البته می‌توانید از ابزارهای دیگری مانند [Glimpse](#) که در این زمینه وجود دارد نیز استفاده کنید. لب کلام این نکته استفاده از profiler برای نرم افزار خود می‌باشد.

10. در بانک اطلاعاتی Data Paging

هنگامیکه از کامپوننت‌های شرکت‌های دیگر (Third party) استفاده می‌کنید، اطمینان حاصل کنید که صفحه بندی اطلاعات در بانک اطلاعاتی انجام می‌شود. برای نمونه کاپوننت گرید شرکت Telerik چند نوع صفحه بندی را پشتیبانی می‌کند. صفحه بندی سمت کاربر (توسط JavaScript)، صفحه بندی سمت سرور توسط کامپوننت و صفحه بندی مجازی. صفحه بندی سمت کاربر یعنی تمامی اطلاعات از سرور به کاربر فرستاده شده و در سمت کاربر عمل صفحه بندی انجام می‌شود. این یعنی واکشی تمامی اطلاعات از بانک و در مورد نرم افزارهای پر کاربر با حجم اطلاعات زیاد یعنی فاجعه. صفحه بندی سمت سرور ASP.NET هم یعنی واکشی اطلاعات از سرور بانک به سرور برنامه و سپس صفحه بندی توسط برنامه. این موضوع هم ممکن است مشکلات زیادی را ایجاد نماید چون باید حداقل تمامی رکوردها از اولین رکورد تا آخرین رکورد صفحه جاری از بانک واکشی شود که این عمل علاوه بر ایجاد سربار شبکه، سربار IO در بانک اطلاعاتی و سربار cpu در سرور ASP.NET ایجاد می‌کند. استفاده از صفحه بندی مجازی، شما را قادر می‌کند بتوانیم اطلاعات را در بانک صفحه بندی کرده و فقط صفحه مورد نظر خود را از بانک واکشی کنیم.

این حالت مجازی در اکثر component‌ها که توسط شرکت‌های مختلف ایجاد شده وجود دارد ولی ممکن است نام‌های متفاوتی داشته باشد. برای این موضوع باید به راهنمای component خریداری شده مراجعه کنید و یا به فروم‌ها و... مراجعه نمایید.

11. بررسی تعداد کوئری‌های صادر شده در یک صفحه و تعداد رکوردهای بازگشت داده شده توسط آن‌ها

این [به این معنا نیست](#) که برای هر query یک context مجزا ایجاد کنید، منظور این است که به بهانه اینکه اطلاعات مختلفی از جداول مختلف مورد نیاز است، query خود را آن قدر پیچیده یا گسترده نویسیم که یا process آن در بانک زمان و سربار زیادی ایجاد کند و یا حجم اطلاعات بلا استفاده ای را از بانک به سرور برنامه لود نماید. به جای این موضوع می‌توانید در یک یا چند context دستورات مجزای واکشی اطلاعات صادر کنید تا تنها اطلاعات مورد نیاز خود را واکشی نمایید. البته این موضوع باعث نشود که تعداد query‌ها مثلاً به 1000 عدد برسد! یعنی باید فیما بین query‌های پیچیده و query‌های ساده ولی با تعداد یکی را که مناسب‌تر با پروژه است انتخاب کنید که این موضوع با تجربه و تست حاصل می‌شود.

نظرات خوانندگان

نویسنده: جواد
تاریخ: ۱۳۹۲/۰۵/۱۲ ۳:۱۵

"استفاده از صفحه بندی مجازی، شما را قادر می‌کند بتوانید اطلاعات را در بانک صفحه بندی کرده و فقط صفحه مورد نظر خود را از بانک واکشی کنیم. این حالت مجازی در اکثر component‌ها که توسط شرکت‌های مختلف ایجاد شده وجود دارد."

میشه این رو بیشتر توضیح بدید که منظورتون چیه . یا باید تمامی اطلاعات رو بفرستیم بعد صفحه بندی کنه یا اینکه به ازای هر صفحه یک کوئری به بانک بفرسته و اطلاعات رو نشون بده . حالا این صفحه بندی مجازی کجا کاربرد داره .

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۵/۱۲ ۱۵:۱۷

هیچ کامپوننتی وجود خارجی نداره که قسمت مدیریت سمت بانک اطلاعاتی رو هم خودش به تنها یی انجام بد. همین کنترل‌های پیش فرض ASP.NET رو هم اگر از شون درست استفاده کنیم، مشکلات کارآیی ندارند. مثلا: (نکته مهمش Skip.Take.ToList استفاده شده هست)

واکشی اطلاعات به صورت chunk chunk (تکه تکه) و نمایش در ListView

نویسنده: مرتضی
تاریخ: ۱۳۹۲/۰۵/۱۲ ۲۱:۲۴

چرا وجود نداره!
از گرید Kendo استفاده کنید اگر paging رو فعال کنید خودش مدیریت میکنه

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۵/۱۲ ۲۲:۲۷

بله. مدیریت می‌کنه، نه به تنها یی. اینجا هم باید اطلاعات Skip و Take رو بهش بدی تا صفحه بندی کم هزینه‌ای رو اعمال کنه. خود GridView در وب فرم‌ها هم paging داره. مشکلش اینه که در حالت پیش فرض کل اطلاعات رو از سرور واکشی می‌کنه و بعد یک صفحه رو نمایش می‌ده. بحث اینه که گرید اگر قراره یک صفحه رو نمایش بده که 20 ردیف داره، بتونه فقط 20 رکورد رو واکشی کنه و نه کل اطلاعات رو و این نیاز به کوئری‌های خاصی در سمت سرور داره. یک نمونه‌اش رو در واکشی اطلاعات به صورت تکه تکه لینک دادم.

نویسنده: م منفرد
تاریخ: ۱۳۹۲/۰۵/۱۲ ۲۲:۴۶

۱. اکثر کنترل‌های ASP.NET WebForm قابلیت bind به DataSet را دارد. اگر از آنها در این مدل استفاده نمایید کار صفحه بندی به عهده کنترل DataSet+ می‌افتد.
۲. صفحه بندی مجازی یعنی شما به کنترل می‌گویید کل اطلاعات شما مثلا ۵۱ صفحه است، الان صفحه ۴ را نمایش می‌دهی این هم اطلاعات صفحه ۴. بعد وقتی کاربر بر روی گزینه صفحه بعد کلیک کرد، به کنترل می‌گویید کل اطلاعات ۵۱ صفحه است، الان صفحه ۵ را نمایش می‌دهی و این هم اطلاعات آن.
برای مثال می‌توانی به این مثال در DataTables مراجعه کنید

نویسنده: مرتضی

گفتم که خودش مدیریت می‌کنه یعنی اینکه اطلاعات Skip , Take رو نمی‌خواهد بهش بدی و خودش اینکار را انجام میده - من دارم ازش تو پروژم استفاده می‌کنم - کل اطلاعات رو واکشی نمی‌کنه و همون 20 رکورد فرضًا صفحه 3 رو واکشی می‌کنه

نوبسند: محسن خان
تاریخ: ۰:۳۱ ۱۳۹۲/۰۵/۱۳

می‌تونی یک مثال با پروفایل SQL نهایی آن ارائه بدی. مطابق بررسی که کردم و حداقل دو تا لینکی که دادم در مورد این کتابخانه، موارد پردازش Skip و Take سمت سرور اون خودکار نیست.

نوبسند: مرتضی
تاریخ: ۴:۴۹ ۱۳۹۲/۰۵/۱۳

کد SQL - خروجی VB - Razor

```
@(Html.Kendo().Grid<Of.Models.vProject>().Name("ProjectsGrid") _  
    .Columns(Sub(column)  
        With column  
            .Bound(Function(p) p.ProjectId).Hidden()  
            .Bound(Function(p) p.Supervisor).Title("ناظر")  
            .Bound(Function(p) p.MapNumber).Title("شماره نقشه")  
            .Bound(Function(p) p.MapCode).Title("کد نقشه")  
            .Bound(Function(p) p.NewStructureArea).Title("متراژ")  
            .Bound(Function(p) p.NumberOfFloors).Title("تعداد طبقات")  
            .Bound(Function(p) p.InsuranceName).Title("بیمه")  
        End With  
    End Sub).Pageable(Sub(p)  
        p.Enabled(True)  
        p.Info(True)  
        p.PageSizes(True)  
        p.Messages(Sub(m)  
            m.Display("{0} - {1} از {2} رکورد - رکوردی برای نمایش وجود ندارد")  
            m.Empty("از")  
            m.Of("از")  
            m.Page("صفحه")  
            m.ItemsPerPage("رکورد در هر صفحه")  
            m.Refresh("بروزرسانی")  
        End Sub)  
    End Sub).Selectable(Sub(s)  
        s.Enabled(True).Mode(GridSelectionMode.Single).Type(GridSelectionType.Row)) _  
        .DataSource(Sub(ds)  
            dsAjax.ServerOperation(True).Read("GetProjects", "Home") _  
                .Model(Sub(m) m.Id(Function(modelId)  
                    modelId.ProjectId)))  
    )
```

```
<HttpPost>  
    Function GetProjects(<DataSourceRequest> request As DataSourceRequest) As JsonResult  
        Return Json(db.vProjects.ToDataSourceResult(request))  
    End Function
```

```
SELECT TOP (5) [Extent1].[ProjectId] AS [ProjectId],  
    [Extent1].[Supervisor] AS [Supervisor],  
    [Extent1].[MapCode] AS [MapCode],  
    [Extent1].[MapNumber] AS [MapNumber],  
    [Extent1].[EmployerName] AS [EmployerName],  
    [Extent1].[InsuranceName] AS [InsuranceName],  
    [Extent1].[NewStructureArea] AS [NewStructureArea],  
    [Extent1].[NumberOfFloors] AS [NumberOfFloors]  
FROM    (SELECT [Extent1].[ProjectId] AS [ProjectId],  
        [Extent1].[Supervisor] AS [Supervisor],  
        [Extent1].[MapCode] AS [MapCode],  
        [Extent1].[MapNumber] AS [MapNumber],  
        [Extent1].[EmployerName] AS [EmployerName],  
        [Extent1].[InsuranceName] AS [InsuranceName],  
        [Extent1].[NewStructureArea] AS [NewStructureArea],  
        [Extent1].[NumberOfFloors] AS [NumberOfFloors],  
        row_number() OVER (ORDER BY [Extent1].[ProjectId] ASC) AS [row_number]  
    FROM    (SELECT [vProject].[ProjectId] AS [ProjectId],  
        [vProject].[Supervisor] AS [Supervisor],
```

```
[vProject].[MapCode] AS [MapCode],  
[vProject].[MapNumber] AS [MapNumber],  
[vProject].[EmployerName] AS [EmployerName],  
[vProject].[InsuranceName] AS [InsuranceName],  
[vProject].[NewStructureArea] AS [NewStructureArea],  
[vProject].[NumberOfFloors] AS [NumberOfFloors]  
FROM [dbo].[vProject] AS [vProject]) AS [Extent1]) AS [Extent1]  
WHERE [Extent1].[row_number] > 5  
ORDER BY [Extent1].[ProjectId] ASC
```

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۵/۱۳ ۸:۲۰

کد سمت سرور db.vProjects.ToDataSourceResult چطور تهیه شده؟ در موردش [اینجا بحث شده](#). شما یک IQueryable باید در اختیارش قرار بدی (که از لحاظ لایه بندی کار مشکل داره) تا بر اساس اطلاعات شماره صفحه و غیره‌ای که از کلاینت می‌رسه خودش مباحثت Skip و Take را پیاده سازی کنه. در حقیقت این کتابخانه فقط [یک متد الحقیقی](#) اضافه‌تر برای اینکار جهت مدیریت مباحثت سمت سرور داره.

نویسنده: مرتضی
تاریخ: ۱۳۹۲/۰۵/۱۳ ۱۳:۴۵

درسته محسن خان- متد الحقیقی ToDataSourceResult در خواست رو می‌گیره و....
بحث سر این بود که هیچ کامپوننتی وجود خارجی نداره که قسمت مدیریت سمت بانک اطلاعاتی رو هم خودش به تنها‌ی انجام بدی

آره شیء DataSourceRequest شامل PageNumber , PageSize میشه ولی دیگه ما کاری بهش نداریم و خودش صفحه بندی رو انجام می‌ده حالا به طریقی نمی‌شود گفت اگر ما از IQueryable استفاده کردیم حتما لایه بندی ما مشکل داره

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۵/۱۳ ۱۴:۱۷

داره. [بهش می‌گن leaky abstraction](#)

نویسنده: مرتضی
تاریخ: ۱۳۹۲/۰۵/۱۳ ۱۴:۲۴

درسته بهش می‌گن [leaky abstraction](#) اما نمی‌گن 100% ایراد

قسمت سوم

12. استفاده از validation سمت کاربر

برای جلوگیری از ارسال و دریافت‌های متنابض اطلاعات به سرور، از `validation` سمت کاربر استفاده نمایید. فرم‌های `html 5` قابلیت‌های چک کردن نوع ورودی‌ها را به صورت خودکار دارد ولی از اتکای به آن پرهیز کنید چون ممکن است یا کاربران برنامه شما از مرورگری استفاده کنند که از `html5` پشتیبانی نکند و یا پشتیبانی کاملی از آن نداشته باشند. برای حل این مشکل می‌توانید از کتابخانه‌هایی مانند `JQuery Validation` و ابزارهایی مانند `MVC` استفاده وسیعی از `Validation` شده که می‌تواند مورد استفاده قرار گیرد.

فراموش نکنید از ابزارهایی مانند `Regex` برای چک کردن سختی کلمات عبور و... نیز در `JavaScript` بهره برداری نمایید. البته دقت کنید که حتماً پیامی مرتبط با خطای به وقوع پیوسته در اختیار کاربر قراردهید تا بتواند آن را بر طرف کند در غیر این صورت بندۀ مسئولیتی راجع به از دست دادن کاربرانتان و یا عصباتی کارفرما بر عهده نمی‌گیرم!

13. استفاده از validation سمت سرور

حتماً به خود می‌گویید نویسنده دچار چندگانگی شخصیت شده است! ولی چنین نیست. این مطلب بیشتر از اینکه در رابطه با ایجاد سرعت بیشتر باشد مربوط به امنیت است. چون `validation` سمت کاربر به سادگی قابل دور زدن می‌باشد. اگر شما تنها `validation` را سمت کاربر انجام دهید و سمت سرور از آن چشم پوشی کنید، به سرعت تمام برنامه شما هک می‌شود. لطفاً دقت کنید که امنیت را **فداء هیچ چیز نکنید**. این یک نکته کلیدی است. البته سوای اینکه این یک نکته امنیتی است، `validation` سمت سرور باعث می‌شود شما بخشی از درخواست‌ها را قبل از انجام `process` زیاد از گردونه خارج کنید و از ارسال اطلاعات اضافی به بانک و ایجاد سربار اضافی جلوگیری کنید.

14. چک کردن اسکریپت‌های مورد استفاده سمت کاربر

استفاده از `master page`ها بسیار سرعت کار را زیاد می‌کنند. بیشتر دوستان `script`‌های سمت کاربر خود را در `cpu` قرار می‌دهند تا در تمامی صفحات لود شوند. این موضوع از طرفی سرعت برنامه نویسی را زیاد می‌کند ولی از طرف دیگر به دلیل اینکه باعث می‌شود فایل‌های `script` در تمامی صفحات بارگذاری شوند، باعث هدر رفت متابع شبکه شما (و کاربرانتان)، ایجاد سربار حافظه و در نتیجه سرعت پایین‌تر برنامه شما خواهد شد. سخت گیری در این موضوع می‌تواند این باشد که حتی شما `function` اضافی هم در سمت کاربر نداشته باشید.

برخی ناظران پژوهه به این موضوعات دقت زیادی می‌کنند. در پژوهه ای که به عنوان ناظر بودم مجری همین کار را انجام داده بود و به دلیل نیاز مبرم کارفرما به سرعت برنامه، این بخش از نظر اینجانب مردود اعلام شده و مجری مجبور به نوشتمندویاره کدهای آن گردید.

قسمت چهارم

15. استفاده از using

اگر از object هایی استفاده می کنید که using interface را پیاده سازی کرده اند، حتماً از عبارت using استفاده کنید. استفاده از دستور using باعث می شود زمانی که دیگر نیازی به object شما نباشد، به صورت خودکار از حافظه حذف شود و در روال جمع آوری زباله (GC) قرار گیرد. این عمل باعث حداقل رسیدن احتمال نشت حافظه در نرم افزار شما می شود. برای مثال:

```

using System;
using System.Text;

class Program
{
    static void Main()
    {
        // Use using statement with class that implements Dispose.
        using (SystemResource resource = new SystemResource())
        {
            Console.WriteLine(1);
        }
        Console.WriteLine(2);
    }
}

class SystemResource : IDisposable
{
    public void Dispose()
    {
        // The implementation of this method not described here.
        // ... For now, just report the call.
        Console.WriteLine(0);
    }
}

```

برای اطلاعات بیشتر می توانید از [این مقاله](#) استفاده کنید.

16. اطلاعات ارسالی توسط شبکه را به حداقل برسانید

حجم اطلاعات ارسالی به شبکه را به حداقل برسانید. ارسال اطلاعات در شبکه به معنی گذر اطلاعات شما از 7 لایه مختلف شبکه در رایانه شما، گذر از media شبکه، گذر مجدد از 7 لایه شبکه در رایانه مقصد می باشد. به این معنی که هر چه اطلاعات بیشتری در شبکه ارسال کنید، سریعتر می شود. برای رفع این مشکل از فشرده سازهای css و javascript استفاده کنید. این فشرده سازها فواصل خالی، دستورات اضافی و... را از کد شما حذف و حجم آن را به حداقل می رسانند. کم کردن تعداد درخواست ها و در نتیجه کم کردن تعداد فایل های ارسالی از سرور به کاربر نیز حربه ای در این زمینه می باشد. برای مقایسه فشرده سازها به صورت آنلاین و استفاده از بهترین آنها (متاسب کد شما) می توانید از [این سایت](#) استفاده کنید. امکانات توکاری هم وجود دارد که در زمان اجرای برنامه css و javascript شما را فشرده و تلفیق می کند ولی با توجه به اینکه برای سرور در هر مرتبه فراخوانی سریع دارد (حتی در صورت کش کردن) اکیدا توصیه می شود از فشرده سازها قبل از اجرای برنامه (Pre-Compressed) به جای فشرده سازهای زمان اجرا (Run-time Compression) استفاده کنید.

نظرات خوانندگان

نویسنده: حسین حقیان
تاریخ: ۱۳۹۲/۰۵/۱۶ ۱۲:۳۸

با سلام و تشکر از مجموعه مطالب مرتبط که ارائه کردید میشه برای این قسمت مثالی رو ذکر بفرمایید اکیدا توصیه میشود از فشرده سازها قبل از اجرای برنامه (Pre-Compressed) به جای فشرده سازهای زمان اجرا (Run-time) استفاده کنید.

با تشکر

نویسنده: م منفرد
تاریخ: ۱۳۹۲/۰۵/۱۶ ۱۹:۵۱

فشرده سازی قبل از اجرای برنامه (Pre-Compression) یعنی که شما قبل از اینکه برنامه خود را در محیط اصلی نصب و اجرا کنید، فایل‌های اسکریپت آن را فشرده کنید. یعنی کاربران فایل اسکریپت فشرده شده را درخواست و دانلود می‌کنند و عملیات اضافی در سمت سرور انجام نمی‌شود. به عنوان مثال شما از فایل `jQuery.min.js` به جای `jquery.js` استفاده کنید. یعنی استفاده از نسخه فشرده شده اسکریپت‌ها.

فشرده سازی زمان اجرا (Run-time Compression) یعنی فشرده سازی اسکریپت‌های مورد نیاز کاربر توسط خود برنامه وب (به صورت خودکار و یا توسط یک ماژول اضافی). این عمل باعث می‌شود که در هر بار درخواست هر کاربر برای یک فایل، برنامه آن را مجدد فشرده سازی کند (و یا از `cache` استفاده کند). این عمل به معنی استفاده بیشتر از منابع پر ارزش سرور شما می‌باشد. به عنوان مثال شما بخواهید در هر مرحله درخواست هر کاربر `jquery.js` را فشرده کنید!

از مقایسه دو حالت بالا مشخص است وقتی شما فقط یکبار اسکریپت‌های خود را فشرده می‌کنید بسیار از حالتی که در هر مرتبه از درخواست کاربران آن را فشرده کنید بهتر است و کمتر منابع سرور را هدر می‌دهد.

قسمت پنجم

17. پرهیز از استفاده نسخه debug

وقتی به ASP.NET مراجعه می‌کنید، توجه فرمایید که از چه نوع build برای محصول نهایی استفاده می‌کنید. وقتی از نسخه **debug** برنامه استفاده می‌کنید، بهبود دهنده‌های سطح کامپایلر عمل نکرده و کدشما در حالت بهینه اجرا نخواهد شد (کد شما همانگونه که هست اجرا می‌شود!).

برای مثال هنگامی که از نسخه release استفاده می‌کنید، کامپایلر C# به صورت خودکار از **StringBuilder**‌ها به جای تلفیق عادی رشته‌ها، از آرایه‌ها به جای لیست‌ها، از دستور **switch/case** به جای دستورات **if/then/else**، تلفیق شروط با یکدیگر و... استفاده کرده و کد شما را در حالت بهینه‌تری اجرا می‌کند. عدم استفاده از این نسخه شما را از این مزايا محروم می‌سازد و نرم افزار شما به کندی اجرا خواهد شد. البته ناگفته نماند این موضوع فقط باید برای محصول نهایی استفاده شود و جهت دیباگ کردن برنامه همچنان باید از نسخه **debug** استفاده نمایید.

توجه نمایید می‌توانید با استفاده از متغیرهای کامپایلر در کد خود بخشی از کد را مختص **build** خاصی از برنامه کنید. مثلاً اگر برنامه در حال کامپایل شد، **MiniProfiler** را فعال کن در غیر این صورت غیر فعال باشد.

```
#if DEBUG
    MiniProfiler فعال کردن //
#endif
```

18. تنظیم دقیق لاغ‌های سیستم در محیط اجرا

وقتی محصول نهایی را آماده می‌کنید، فراموش نکنید که سطح لاغ گیری را در سطح مطلوبی قرار دهید تا بتوانید در صورت نیاز برنامه را اشکال زدایی کنید. البته زیاده روی در این مورد نیز می‌تواند مشکل زا باشد.

اکثر برنامه نویسان هنگامی که محصول نهایی را برای مشتری آماده می‌کنند، لاغ را غیر فعال می‌کنند تا کاربر سرعت بیشتری را تجربه کند. این سیاست غلط شما را از امکانات بی نظیر لاغ کردن (مانند وقابع نگاری امنیتی، رفع سریع مشکلات و...) محروم می‌سازد. بنابر این حتماً سیستم لاغ خود را در زمان تولید محصول اصلی (و نصب بر روی سرور اصلی) در حالت متعادلی تنظیم نمایید. کمی تست و تجربه شما را در این امر یاری می‌کند.

19. مشخص کردن اندازه عکس

مشخص کردن اندازه عکس در تک **img** به صورت **css** یا **attribute** باعث می‌شود که همان اولین بار که صفحه رندر می‌شود، اندازه مورد نیاز عکس به آن اختصاص یابد تا در صورت دانلود سریعاً جایگرین آن گردد. عدم مشخص کردن سایز عکس (طول و عرض) باعث رندر شدن مجدد تمامی المان‌های صفحه بعد از دانلود هر عکس از سرور می‌شود و منابع با ارزش **cpu** کاربر شما را به سادگی از بین می‌برد.

```

```

نظرات خوانندگان

نوبسند: مرتضی
تاریخ: ۱۳۹۲/۰۷/۰۷ ۱۶:۵۴

سلام. ممنون از مطلب خوبتون. بند وقتي سايتم رو با ويژوال استاديو باز ميکنم فقط مود ديباگ داره و مود release رو نداره. فقط برای برنامه های ویندوز فرم و wpf مود release داره. پس با اين اوصاف بند چطوری سايتم رو تویه مود release پابلیش کنم؟ فقط کد زیر رو تویه و ب کانفیگ قرار میدم

```
<compilation debug="false" targetFramework="4.0">
```

آيا قطعه کد بالا همون کار release رو انجام مide. با تشکر

نوبسند: م منفرد
تاریخ: ۱۳۹۲/۱۰/۱۴ ۰:۱۰

در بخش publish از منوی debug میتوانید نوع خروجی را مشخص کنید که برنامه با debug پابلیش شود یا release

قسمت ششم

20. اسکریپت در پایین صفحه

لینک‌های مربوطه به javascript خود را تا جای ممکن در پایین صفحه قرار دهید. وقتی parser مرورگر به فایل‌های javascript می‌رسد، تمامی فعالیت‌ها را متوقف کرده و سعی در دانلود و سپس اجرای آن دارد. برخلاف اینکه مرورگرها امکان دانلود چند فایل را به صورت همزمان از سرور دارند، هنگامی که به اسکریپت‌ها می‌رسند، تنها یک فایل را دانلود می‌کنند. یعنی اجرای برنامه و دانلودهای مرتبط با صفحه شما متوقف شده و پس از دانلود و اجرای اسکریپت اجرای آنها ادامه پیدا می‌کند. این مسئله وقتی نمود بیشتری پیدا می‌کند که شما فایل‌های اسکریپت با حجم و تعداد بالا در برنامه خود استفاده می‌کنید. برای فرار از این مشکل می‌توانید تگهای مربوط به اسکریپت را در آخر صفحات خود بگذارید. فقط دقت کنید اگر نیاز است که قبل از نمایش صفحه تغییری در DOM ایجاد کنید، باید حتی اسکریپت‌های مربوطه را بالای صفحه قرار دهید. روش دیگر دانلود فایل‌های اسکریپت به وسیله AJAX است که انشاء الله در آینده مقاله‌ای در این رابطه خواهم نوشت.

CDN.21

سرورهای Content Delivery Network یا CDN یا سرورهای توزیع شده ای در سطح دنیا هستند که یک نسخه از برنامه شما بر روی آن قرار دارد. هنگامی که کاربر می‌خواهد به سایت شما دسترسی پیدا کند به صورت خودکار به نزدیکترین سرور منتقل می‌شود تا بتواند سرعت بیشتری را تجربه کند. علاوه بر این CDN باعث بالانس شدن بار ترافیک شبکه شما شده خط حملات D.O.S و D.D.O.S را به حداقل می‌رساند.



زمانیکه شما یک سیستم CDN را فعال میکنید تاثیر آن بصورت زیر خواهد بود:

- ۱- شبکه توزیع محتوا یا همان CDN تمامی سرورهای شبکه جهانی اینترنت را پوشش میدهد. بنابراین زمانیکه شما این سیستم را برای سایت خود فعال میکنید اطلاعات شما بر روی تمامی این سرورها کپی و ذخیره میشود و زمانیکه یک بازدیدکننده به سایت یا وبلاگ شما وارد میشود محتوای سایت شامل تصاویر و متن را از نزدیکترین سرور نزدیک به خود دریافت میکند و مستقیماً به هاست یا سرور شما متصل نمیشود. این کار موجب بهبودی چشمگیر در عملکرد سایت شما خواهد شد.
- ۲- تمام اطلاعات ثابت شما مانند تصاویر، کدهای CSS و javascript, mp3, pdf و فایلهای ویدئویی شما را پشتیبانی میکند و تنها اطلاعاتی که قابل تغییر و بروزرسانی هستند مانند متون و کدهای HTML از سرور اصلی شما فراخوان میشوند. با این کار مصرف پنهانی باند هاست شما کاهش یافته و هزینه ای که سالانه برای آن میبردازید کاهش چشمگیری خواهد داشت.
- ۳- تفاوت سرعت و عملکرد برای خودتان یا افرادی که در نزدیکی سرور اصلی شما هستند تفاوت زیادی نخواهد داشت ولی برای کسانی که از نقاط مختلف جهان به سایت شما وارد میشوند این افزایش سرعت ناشی از CDN کاملاً محسوس خواهد بود، با توجه به اینکه سایتها ایرانی معمولاً سرور و هاست خود را از خارج و کشورهایی مانند آلمان و آمریکا تهیه میکنند و عموم بازدیدکنندگان

از داخل کشور هستند استفاده از CDN میتواند بسیار موثر باشد. برای تعیین تاثیر CDN بر سرعت سایت میتوانید عملکرد خود را با ابزارهایی مانند [GTmetrix](#) و [Pingdom](#) بعد و قبل از فعال سازی CDN بررسی و مقایسه کنید. ابزارها، تکنیک‌ها و روش‌های متفاوتی برای راه اندازی CDN وجود دارد که نیازمند مقاله‌ای مجزا می‌باشد.

قسمت هفتم

22. استفاده از CSS Sprites

ایده اصلی این تکنیک به این صورت است که تمامی عکس‌های کوچک (دراینجا همه 100 عکس) در قالب یک تصویر بزرگ قرار خواهد گرفت و با استفاده از CSS مختصات هر عکس کوچک را در تصویر بزرگ پیدا کرده و نمایش می‌دهیم. یکی شدن 100 عکس کوچک به یک عکس بزرگ، تاثیر زیادی در پایین آمدن حجم عکس جدید خواهد داشت و مرورگر شما به جای درخواست 100 عکس از سرور، تنها یکی دانلود می‌کند و از این به بعد از کش مرورگر برای بازیابی آن استفاده می‌کند. این موضوع به معنی ترافیک کمتر شبکه و آزاد شدن منابع پر ارزش حافظه، CPU و پهنای باند در سمت سرور و کاربران. برای اطلاع بیشتر از این تکنیک می‌توانید [به این مقاله](#) مراجعه نمایید.

23. استفاده مطلوب از AJAX

شما می‌توانید برای لود کردن بخش‌های مخفی در صفحه خود از AJAX کمک بگیرید. به جای دانلود کردن تمامی بخش‌های صفحه در مرورگر کاربر، بخش‌هایی که در دید کاربر قرار ندارد را به صورت AJAX بارگیری کنید. نمونه ای از این تکنیک را در این [صفحه](#) مشاهده نموده و البته از کد آن استفاده نمایید:

24. حذف HTTP modules اضافی

ASP.NET HTTP modules را که در برنامه خود استفاده نمی‌کنید. این کار یعنی سربار مدیریتی کمتر در ماژول سرور شما. برای اجرای این مورد می‌توانید از کدی مشابه این کد در web.config خود استفاده کنید:

```
<httpModules>
<remove name="OutputCache"/>
<remove name="Session"/>
<remove name="WindowsAuthentication"/>
<remove name="FormsAuthentication"/>
<remove name="PassportAuthentication"/>
<remove name="RoleManager"/>
<remove name="UrlAuthorization"/>
<remove name="FileAuthorization"/>
<remove name="AnonymousIdentification"/>
<remove name="Profile"/>
<remove name="ErrorHandlerModule"/>
<remove name="ServiceModel"/>
</httpModules>
```

البته حواستان به این موضوع باشد ماژول‌های مورد استفاده در برنامه خود را حذف نکنید که در این صورت ممکن است این آخرین پروژه شما با صاحب کارتان باشد!

یک DLL رایگان برای انجام بازنویسی URL ها در ASP.NET میباشد. به این صورت که بر اساس تنظیماتی که در فایل Web.Config میشود، عمل تغییر URL را انجام میدهد و با زبان فارسی نیز سازگار است؛ مثل استفاده از عنوان فارسی در URL برای SEO بهتر:



برای استفاده، ابتدا DLL مورد نظر را دانلود و به References میکنیم؛ دانلود فایل [Intelligencia.UrlRewriter.dll](#)

دانلود آخرین نسخه‌ی فایل از سایت مرجع: [urlrewriter.net](#)

مرحله‌ی بعد تنظیمات Web.Config است که به صورت زیر میباشد:

```
<configuration>
  <configSections>
    <section name="rewriter" requirePermission="false"
    type="Intelligencia.UrlRewriter.Configuration.RewriterConfigurationSectionHandler,
    Intelligencia.UrlRewriter" />
  </configSections>
  <system.web>
    <httpModules>
      <add type="Intelligencia.UrlRewriter.RewriterHttpModule,Intelligencia.UrlRewriter"
      name="UrlRewriter" />
    </httpModules>
  </system.web>
  <rewriter>
    <rewrite url="^.*-T([0-9]+)/?*$" to("~/ShowTerol.aspx?ID=$1" processing="stop" />
  </rewriter>
</configuration>
```

مرحله بعد تنظیم این است که کدام لینک‌ها بازنویسی شوند که در کدهای بالا فقط یک مدل بازنویسی وجود دارد:

```
<rewrite url="^.*-T([0-9]+)/?*$" to "~/ShowTerol.aspx?ID=$1" processing="stop" />
```

و میگوید تمام لینک‌هایی را که در آخر آنها T و عدد وجود دارد، به صفحه‌ی ShowTerol.aspx با یک کوئری استرینگ به نام ID که مقدار آن، همان عدد بعد حرف T میباشد بفرست.

يعني اين لينك را



به این لینک تبدیل میکند

ShowTerol.aspx?ID=87

مثال‌های بیشتر

```
<rewrite url="^.*-p([0-9]+)/?*$" to="~/Products.aspx?p=$1" processing="stop"/>
<rewrite url="^.*-c([0-9]+)pa([0-9]+)/?*$" to="~/Default.aspx?c=$1&pa=$2" processing="stop"/>
<rewrite url="^.*-c([0-9]+)/?*$" to="~/Default.aspx?c=$1" processing="stop"/>
```

الگوی تبدیلات توسط [Regular Expansion](#)

نظرات خوانندگان

نویسنده: رضا
تاریخ: ۲۳:۷ ۱۳۹۲/۰۶/۲۹

سلام ، با تشکر از پست خوبتون .
بعد از اجرا سایت ارور 500 میده چیکار کنم ؟
هم لوکال و هم روی هاست .
خطا مربوط به این خط برنامه در Web.Config است :

```
<httpModules>
  <add type="Intelligencia.UrlRewriter.RewriterHttpModule,Intelligencia.UrlRewriter"
name="UrlRewriter" />
</httpModules>
```

نویسنده: محسن خان
تاریخ: ۲۳:۲۵ ۱۳۹۲/۰۶/۲۹

خطای 500 یک خطای عمومی هست. اصل خطای چی بود؟ چه stack trace ایی در صفحه لاغ میشه؟

نویسنده: رضا
تاریخ: ۲۳:۴۶ ۱۳۹۲/۰۶/۲۹

با تشکر از شما،
مشکل را پیدا کردم. فکر کنم برای IIS‌های از 6 به بالا باید به سبک زیر تنظیم بشه .

```
<configuration>
<configSections>
  <section name="rewriter" requirePermission="false"
type="Intelligencia.UrlRewriter.Configuration.RewriterConfigurationSectionHandler,
Intelligencia.UrlRewriter" />
</configSections>

<system.webServer>
  <modules>
    <add type="Intelligencia.UrlRewriter.RewriterHttpModule,Intelligencia.UrlRewriter"
name="UrlRewriter" />
  </modules>
</system.webServer>

<rewriter>
  <rewrite url="^.*-T([0-9]+)/?*$" to("~/ShowTerol.aspx?ID=$1" processing="stop" />
</rewriter>
</configuration>
```

نویسنده: رضا منصوری
تاریخ: ۱۶:۴۱ ۱۳۹۲/۰۶/۳۰

خوشحالم مشکلتون رفع شد ولی من بارها استفاده کردم و مشکلی نداشتم

نویسنده: heaven
تاریخ: ۲۲:۴۱ ۱۳۹۲/۱۱/۰۳

سلام
من این کارهایی که گفتید رو انجام دادم ولی تغییری حاصل نشد. به علاوه چند تا مسیح میده:

```
Could not find schema information for the element 'rewrite'
Could not find schema information for the attribute 'url'
Could not find schema information for the attribute 'to'
Could not find schema information for the attribute 'processing'
```

چی کار کنم؟

نویسنده: heaven
تاریخ: ۲۳:۵۲ ۱۳۹۲/۱۱/۰۳

برای تبدیل ۱ به ۱ در `Details.aspx?code=1` این کد درسته؟

```
<rewrite url="~/Details/(.*)" to="~/Details.aspx?code=$1" processing="stop" />
```

راستی میشه بپرسم چطور میتونم توی آدرس اسم محصولی رو که روش کلیک شده نشون بدم؟ یعنی به جای اون `code=1` اسم محصولم رو نشون بده

نویسنده: رضا منصوری
تاریخ: ۱۱:۶ ۱۳۹۲/۱۱/۰۴

سلام

برای IIS‌های 6 به بالا همانطور که دوستمون اشاره کرد باید به جای

```
<system.web>
  <httpModules>
    <add type="Intelligencia.UrlRewriter.RewriterHttpModule,Intelligencia.UrlRewriter"
name="UrlRewriter" />
  </httpModules>
</system.web>
```

از این استفاده کنیم

```
<system.webServer>
  <modules>
    <add type="Intelligencia.UrlRewriter.RewriterHttpModule,Intelligencia.UrlRewriter"
name="UrlRewriter" />
  </modules>
</system.webServer>
```

دیگه نباید مشکلی وجود داشته باشه

نویسنده: رضا منصوری
تاریخ: ۱۱:۱۵ ۱۳۹۲/۱۱/۰۴

یه نمونه برنامه برآتون ضمیمه کردم که همین کاری که گفتیدو انجام دادم

[Test.zip](#)

موفق باشید

نویسنده: heaven
تاریخ: ۲۰:۳۸ ۱۳۹۲/۱۱/۰۴

واقعاً ممنونم. من دقیقاً دنبال این یه خط می‌گشتدم:

```
string.Format("Details/{0}-{1}", Eval("Name").ToString().Replace(" ", "-"), Eval("id"))
```

که البته من اون P رو حذف کرم. اگه بخوام id رو هم نشون نده فقط همون اسم محصول رو نشون بده باید چی کار کنم؟ یا مثلا بخوام فیلدهای دیگه محصول رو هم نشون بده.

نویسنده: رضا منصوری
تاریخ: ۲۳:۴۴ ۱۳۹۲/۱۱/۰۴

خواهش میکنم

شما باید بر اساس کلید در صفحه‌ی مقصد محصولی که فرستادید و پیدا کنید منظورم کار این خطه

```
Label1.Text =
    GetProducts.Get()
        .Where(x => x.Id == int.Parse(Request.QueryString["PID"].ToString()))
        .Select(x => x.Name)
        .FirstOrDefault();
```

اگه کد محصولو به صفحه‌ی که میخواهید جزیاتو تو اون نشون بدید نفرستید چجوری میخواید محصول مورد نظر رو شناسایی کنید تا بتونید جزیات اونو نشون بدید؟
البته راه‌های دیگه هم هست که مثلا نام محصولتون به عنوان کلید باشه و بر اساس اون محصولو تو صفحه‌ی مقصد شناسایی کنید که به نظر من اصلا کار منطقی نیست.
یا کد محصولو توسط راه دیگری بجز URL بفرستید مثل session که غیر منطقی تره.
در کل میتوانید فقط نام محصولو بفرستید که اینطوری میشه

```
<a href='## "Details/" + Eval("Name")##'><%# Eval("Name") %></a>
```

ولی چجوری محصولو تو صفحه‌ی مقصد میخواید شناسایی کنید تا جزئیاتشو نشون بدید؟
بهترین راه اینه که کد محصول هم بفرستید.

نویسنده: heaven
تاریخ: ۱۲:۱۰ ۱۳۹۲/۱۱/۰۵

بله در این که باید کد محصول رو بفرستیم که شکی نیست ولی منظورم این بود راهی نیست که توی آدرس این کد نشون داده نشه؟ در واقع کد توی آدرس اصلی هست اما توی آدرس rewrite شده نشون داده نمیشه. مثلا این نمونه رو ببینید. یه چیزی مثل این مد نظرم بود که البته زیاد مهم نیست.

<http://www.site.com/Product/DKP-14997/Samsung-Galaxy-S4-I9500-16GB/name>

یه سوال دیگه هم داشتم. اگه بخوام پسوند aspx رو از آخر آدرس همه صفحات سایت حذف کنم باید چی کار کنم؟ ممنون.

نویسنده: محسن خان
تاریخ: ۱۲:۱۸ ۱۳۹۲/۱۱/۰۵

- اون DKP-14997 کد محصول است که یا منحصر بفرد تعریف شده یا کلید اصلی.

- برای حذف از تمام سایت روش [friendly urls](#) هم هست علاوه بر روش مطلب جاری.

نویسنده: رضا منصوری

تاریخ: ۱۵:۱۶ ۱۳۹۲/۱۱/۰۵

همونطور که دوستمون اشاره کرد DKP-14997 تو سایت دیجی کالا کلید محصولاته. روشی که در این مطلب برای انجام UrlRewrite خوندین به نظرم روش کامل و اصولی برای انجام باز نویسی یو آر الها نیست. اگه بخواید کاملاً بر روی یو آر الهای سایت مدیریت داشته باشید باید از امکان مسیر یابی خود ASP.Net استفاده کنید. در وب فرم [این مقاله](#) کمکتون میکنه. در ام وی سی هم [این مطلب](#). سایتهايی که اصولی هستند از قابلیت مسیر یابی خود ASP.Net استفاده میکنند. مثل همون دی جی کالا یا سایت جاری.

نویسنده: afsoon تاریخ: ۱۱:۴۰ ۱۳۹۲/۱۱/۰۷

من می خوام علاوه بر اینکه وقتی به یه صفحه جدید میرم آدرس عوض بشه، توی همون صفحه هم با ایجاد تغییراتی آدرس صفحه ام عوض بشه. مثلاً وقتی کاربر سرچ می کنه صفحه تغییر نمی کنه و نتایج سرچ توی همون صفحه نشون داده میشه اما می خوام آدرس عوض بشه. به این صورت که عبارت سرچ شده توی آدرس بیاد. یا مثلاً وقتی از داخل دراپ دان لیست یک ویژگی خاص محصول انتخاب میشه (که اینم به نوعی سرچ هست) اون ویژگی توی آدرس همون صفحه نشون داده بشه.

برای این کار از کدوم روش استفاده کنم بهتر و راحت تر؟ url rewriting, url routing, url friendly؟
اگه یک راهنمایی هم در مورد اینکه چطور باید این کار رو انجام بدم بکنید ممنون میشم. یعنی وقتی صفحه عوض نمیشه من این تغییر رو دقیقاً کجا باید اعمال کنم؟

نویسنده: محسن خان تاریخ: ۱۴:۳۱ ۱۳۹۲/۱۱/۰۷

- اطلاعات ارسالی توسط کاربر فقط زمانی در URL صفحه ظاهر می شوند که [متده ارسال اطلاعات فرم](#)، بجای Post حالت Get باشد.
- یا اینکه در حین Redirect خودتان دستی این کوئری استرینگ را تنظیم کنید.

```
Response.Redirect("tst.aspx?id=" + txtSrch.Text)
```

الان که این کوئری استرینگ رو دارید، مابقی آن مباحثی هست که در این مطلب و پرسش و پاسخ های آن مطرح شده.

نویسنده: afsoon تاریخ: ۱۷:۲۷ ۱۳۹۲/۱۱/۰۸

اگه بخوام آدرسم به این صورت دربیاد tst/Keyword=txtSrch.Text باید به جای عبارت داخل " " چی بنویسم؟ الگوی تبدیلش رو هم میشه بگید؟ این الگوی تبدیل ها و نحوه نوشتمن آدرس جدید خیلی سخته چطور باید اینا رو فهمید؟

نویسنده: رضا منصوری تاریخ: ۲۰:۱۶ ۱۳۹۲/۱۱/۰۸

برای ساختن لینکش همونطور کو دوستمون توضیح داد.

```
Response.Redirect("tst/" + txtSrch.Text)
```

برای باز نویسی آدرس اون هم

```
<rewrite url="~/tst/(.*)" to="~/mypage.aspx?searchtxt=$1" />
```

که در صفحه‌ی مقصود با کوئری استرینگ searchtxt می‌توانید اطلاعاتو بدست بیارید. همونطور که تو مطلب گفتم الگوی تبدیلات توسط Regular Expansion انجام میشه. اینم یه [مطلب](#) که کمکتون میکنه.

نویسنده: مائده

تاریخ: ۲۳:۲۶ ۱۳۹۲/۱۱/۰۹

اگه وب فرم‌ها توی فولدر جدا باشند، الگوی تبدیل چجوری میشه؟
اگه اسم فولدر رو قبل از اسم پیج بیاریم صفحه بهم میریزه! اگه نیاریم هم جواب نمیده!

نویسنده: رضا منصوری

تاریخ: ۱۰:۳۶ ۱۳۹۲/۱۱/۱۰

من از این روش برای تبدیلات ساده و کوچک استفاده کردم، همونطور که [گفتم](#) این روش به نظرم روش کامل و اصولی برای بازنویسی Url‌ها نیست و روش درستش استفاده از امکان [Url Routing](#) خود Asp.Net هستش.
[این مطلب کمکتون میکنه](#).

نویسنده: مائده

تاریخ: ۱۱:۴ ۱۳۹۲/۱۱/۱۰

من دات نت فریم ورکم 3.5 هست برای همین نمی‌تونم از url routing و url friendly استفاده کنم.

نویسنده: وحید نصیری

تاریخ: ۱۱:۱۱ ۱۳۹۲/۱۱/۱۰

از دات نت 3 و نیم، سرویس پک یک به بعد به صورت توکار و استاندارد، اضافه شده. فقط برای حالت دات نت سه و نیم باید یک سری تنظیمات اضافه‌تر به وب کانفیگ [اضافه شوند](#) (تعريف [System.Web.Routing.UrlRoutingModule](#) به [httpModules](#) ذکر شود).

نویسنده: زیلابی

تاریخ: ۱۰:۳ ۱۳۹۳/۰۵/۲۰

سلام. من کل مراحل رو انجام دادم ولی یک مشکل دارم که فکر می‌کنم به عبارت ریگولار رو اشتباه نوشتم.
میخواهم به طور مثال این آدرس:
www.sample.com/page2/page2.aspx?MID=1251
به این تبدیل بشه :

www.sample.com/1251

سوال دیگه ای که داشتم اینه که آیا توی صفحاتی که میخواهم کوری استرینگ رو میخواهم بگیرم باید کاری کنم؟
و اینکه آیا با این روش فایل‌های CSS, javascript و مشکل پیدا نمی‌کنم؟

نویسنده: رضا منصوری

تاریخ: ۱۶:۴۳ ۱۳۹۳/۰۵/۲۰

سلام . همونطور که [گفتم](#) به نظرم استفاده از امکان مسیر یابی خود دات نت بهترین گزینه است و نمیخواهد خودتونو با ریگولار و ... درگیر کنید .
آموزششو از [اینجا میتوانید](#) بخونید.

یه نمونه برنامه ساختم و کاری که میخوايد رو از [روش روتینگ خود دات نت](#) انجام دادم . یعنی این آدرس رو www.sample.com/1251 به صفحه News.aspx میفرسته و اونجا میتوانید کد 1251 رو بخونید و کارتون را ادامه بدید. [Test](#)-

52676fa0948d4909be628f7a6a24bc5d.zip

در مورد روش درست لینک کردن فایل‌های جاوا اسکریپت و CSS میتوانید [اینجا](#) متوجه بشید.

چرا افسانه‌ای که می‌گوید PHP از ASP.NET سریعتر است اینقدر شایع است؟ در این مقاله به بیان حقایقی می‌پردازیم که این افسانه را زیر سوال می‌برد؟

خیلی وقتها در بسیاری از نوشهای اینقدر شایع است. آزار دهنده‌ترین بخش این ادعاهای اینکه PHP بسیار سریعتر از ASP.net است و اینکه لحاظ سرعت کند است. هر یک از آنها را که نگاه می‌کنی بصورت کاملاً غیر واقع بینانه به موضوع نگاه می‌کنند و فقط بدون دلیل این موضوع را ادعا می‌کنند. زیرا به این موضوع بصورتی کاملاً متعصبانه و بدور از واقعیتها نگاه می‌شود. به همین دلیل بصورت گسترده‌ای این افسانه در میان اهالی وب پذیرفته شده است.

حال بجای اینکه این موضوع را بارها و بارها در جاهای مختلف بیان کنیم، این مقاله را نوشه و در هر کجا که لازم باشد به آن ارجاع خواهیم داد. باید توجه کنید این حقیقت که زبان PHP یک زبان اصیل و قدرتمند است هیچ شکی در آن نیست اما اینکه بخواهیم بصورت مغرضانه و به این دلیل که ما از این زبان استفاده می‌کنیم، آنرا از هر لحاظ برتر از سایر زبانها بدانیم (کمی که نه) بسیار اغراق آمیز است.

این مقاله برای این نیست که ما هریک از این زبانها را زیر سوال ببریم. بلکه برای آن است که این موضوع را با دلایل منطقی و حقیقی بررسی کنیم که آیا اینکه می‌گویند PHP از ASP.net سریعتر است واقعیت دارد یا نه؟

:Interpreted Languages **vs** Compiled Languages

قبل از هرچیز ذکر این نکته الزامی است که این دو زبان تفاوت‌های اساسی در base دارند. ASP.net یک زبان بهینه سازی و کامپایل شده است، به این معنی که کدهای نوشته شده در این زبان قبل از اینکه قابل اجرا شوند، به مجموعه ای از دستورالعمل‌های خاص ماشین تبدیل می‌شوند. از سوی دیگر PHP یک زبان تفسیر شده است، به این معنی که کدهای نوشته شده به همان شکل ذخیره شده و در زمان اجرا این کدها تفسیر می‌شوند. این موضوع بطور گسترده‌ای پذیرفته شده و ثابت شده است که برنامه‌های کامپایل شده به مراتب سریعتر از برنامه‌های تفسیر شده اجرا می‌شوند، به این دلیل که برنامه‌های تفسیر شده نیاز دارند تا در زمان اجرا به دستورالعمل‌های ماشین تبدیل شوند.

در اینجا به یک نقل قول از دانشنامه آزاد ویکی پدیا اشاره می‌کنم که میزان سریعتر بودن برنامه‌های کامپایل شده را نشان می‌دهد:

A program translated by a compiler tends to be much faster than an interpreter executing the same program: even ".a 10:1 ratio is not uncommon. The mixed solution's efficiency is typically somewhere in between"

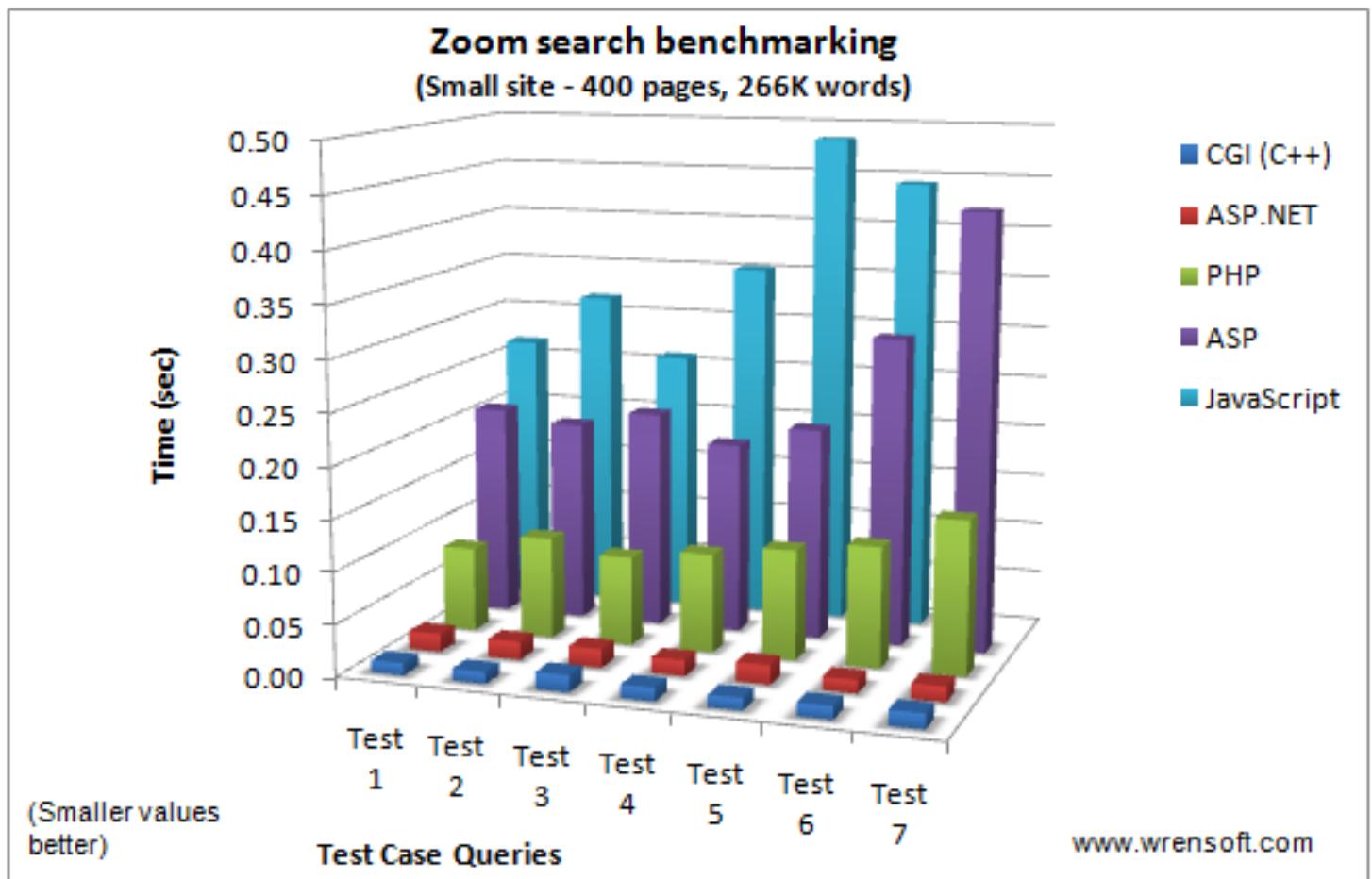
به این معنا که یک برنامه بصورت کامپایل شده بسیار سریعتر از همان برنامه بصورت تفسیر شده، اجرا می‌شود.

اعداد و ارقام:

حال که تئوری خود را مبنی بر دلیل سریعتر بودن ASP.net بیان کردیم باید با هم نگاهی به برخی آمارها بیاندازیم تا این تئوری را در عمل هم نشان داده باشیم.

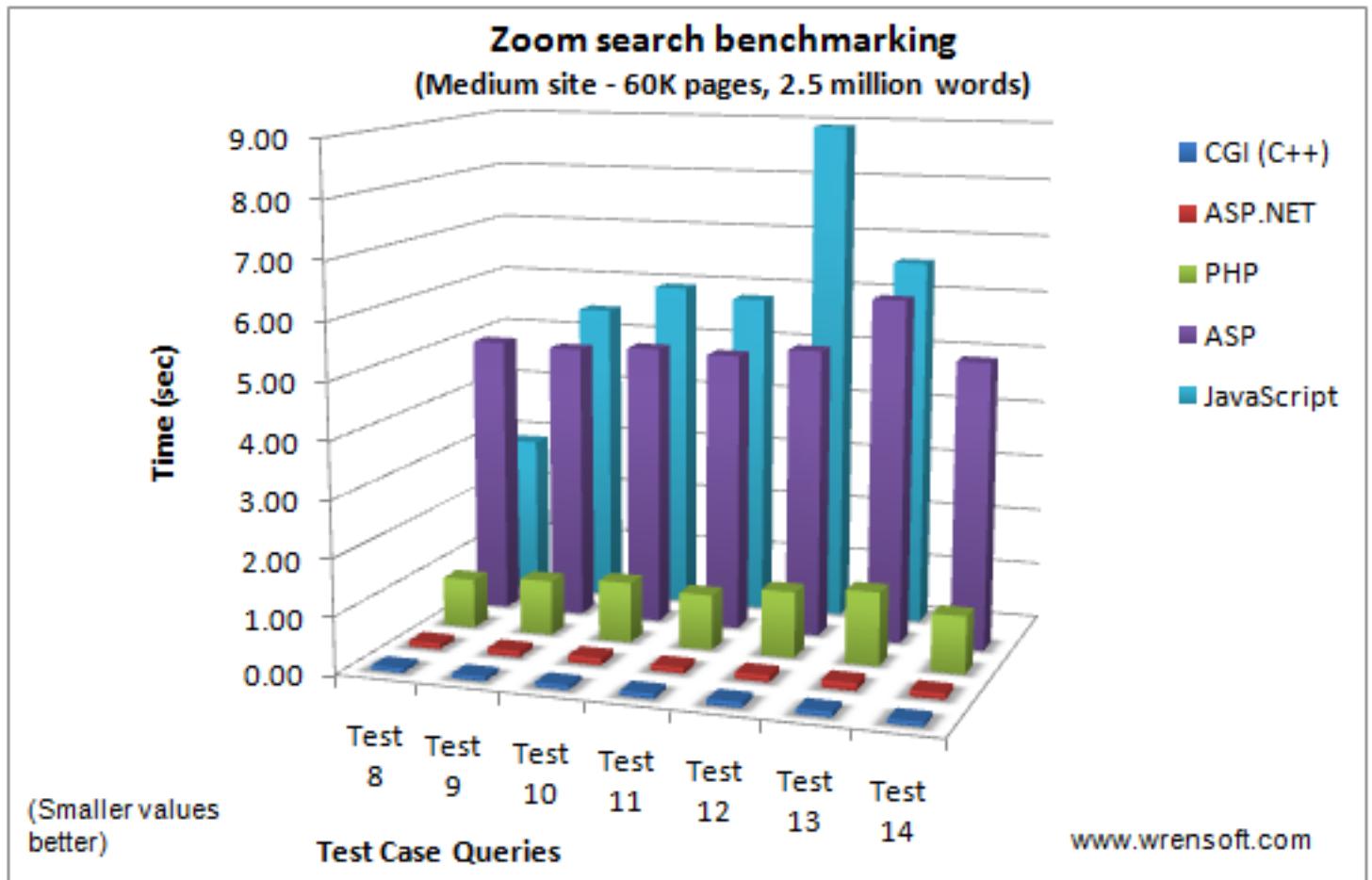
آمارهای زیر توسط شرکت WrenSoft جهت مقایسه زمان اجرای یک کد مشابه در زبانهای مختلف تهیه شده است. اگر می‌خواهید توصیف عمیق‌تری از آمارها داشته باشید لطفاً لینک را دنبال کنید.

نمودار اول: زمان صرف شده برای تولید و نمایش نتایج برای جستجوی وب سایت‌های کوچک



همانطور که میبینید زمان متوسط برای سایت PHP 0.1500 ثانیه و برای سایت ASP.net 0.0150 ثانیه است. یک تفاوت بزرگ: PHP ده برابر نسبت به ASP.net کندتر است!

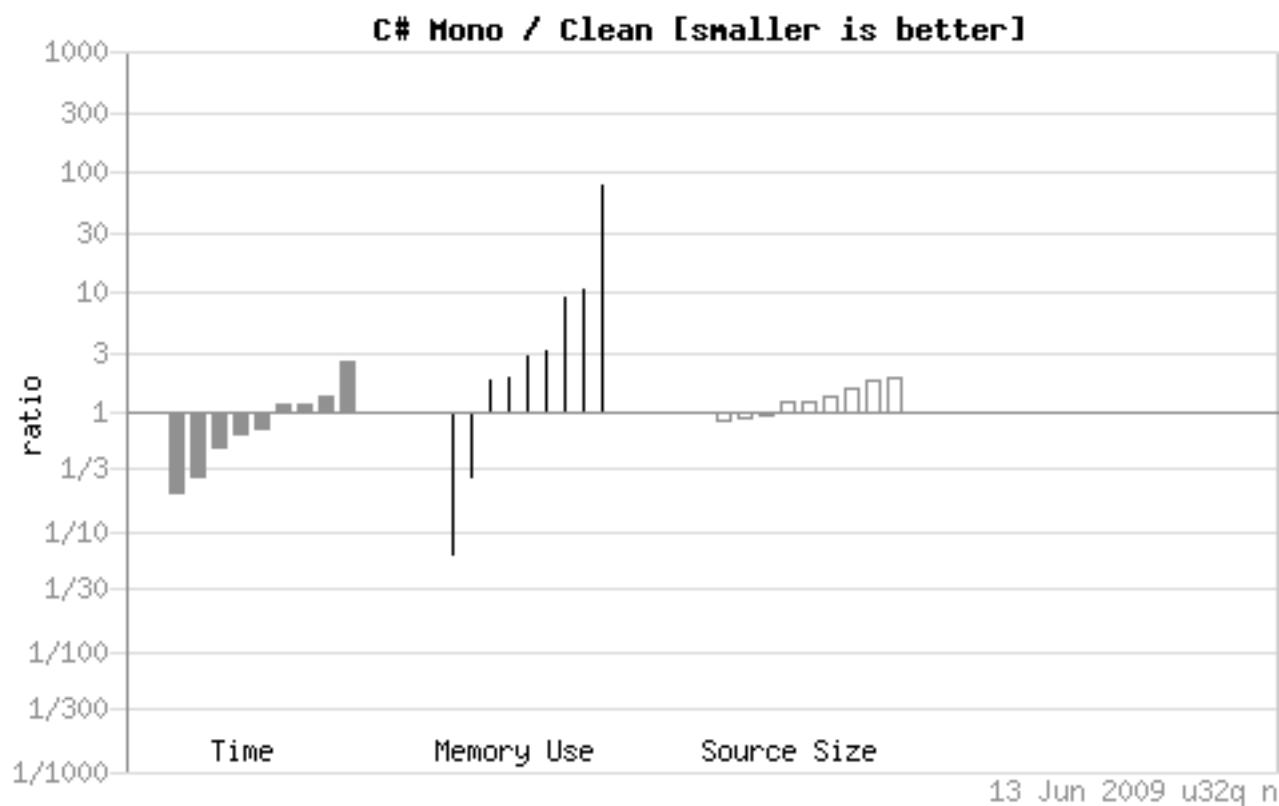
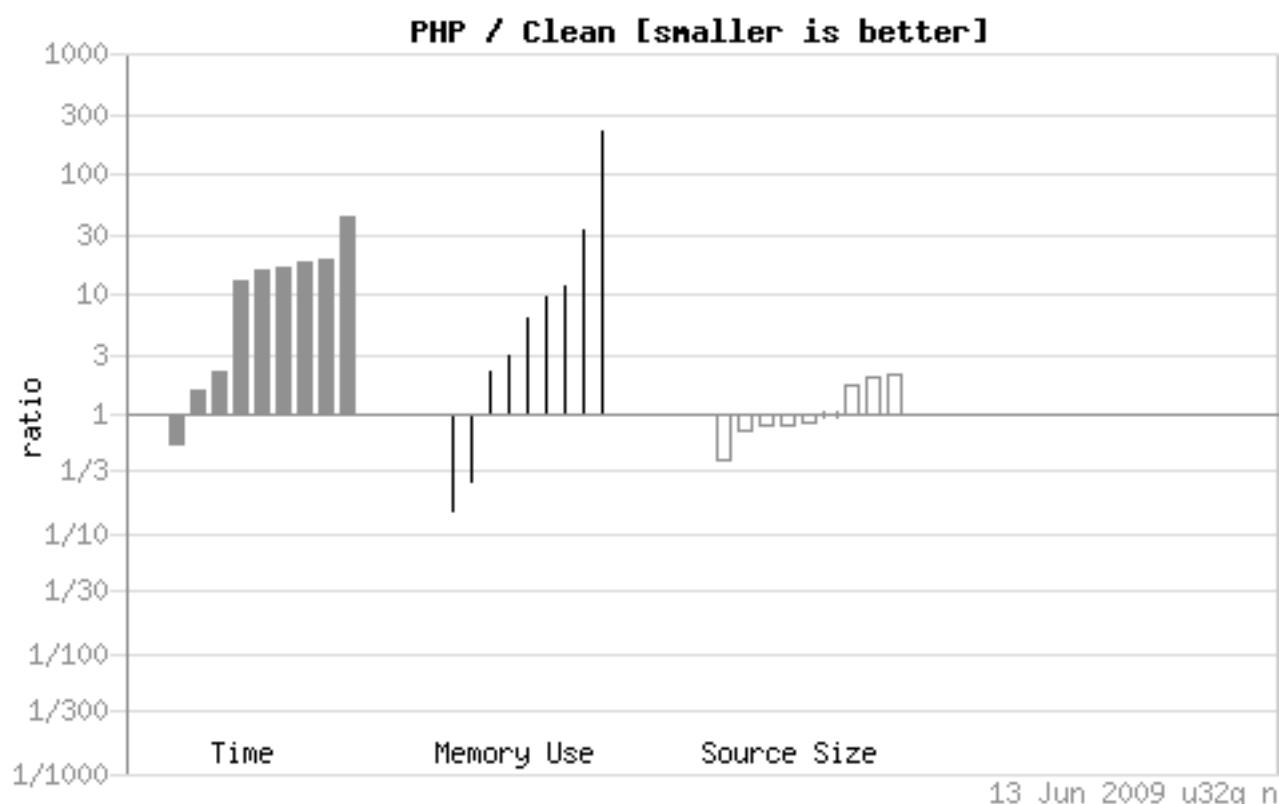
نمودار دوم: زمان صرف شده برای تولید و نمایش نتایج برای جستجوی وب سایت‌های متوسط



PHP 1.0097 ثانیه طول می‌کشد در حالی که ASP.net 0.0810 ثانیه زمان نیاز دارد. می‌بینیم که PHP دوازده بار بیشتر از ASP.net زمان می‌برد.

در حال حاضر این آزمون با یک کد مشابه در زبانهای برنامه نویسی مختلف پیاده سازی و اجرا شد و نتیجه را مشاهده نمودید. حال این موضوع پیش می‌اید که این اجرای کدها بر روی سیستم عامل ویندوزی بوده است و این می‌تواند به نفع ASP.net باشد، پس همین آزمون را بر روی سیستم عامل لینوکسی مشاهده می‌کنیم.

آمارهای زیر از سایت معتبر shootout.alioth.debian.org گرفته شده است. این آمارها نحوه اجرای همان کد را بر روی سیستم عامل لینوکسی برای هردو زبان نشان می‌دهد:



همانطور که مشاهده می‌کنید در سیستم لینوکسی نیز همچنان ASP.net سریعتر از PHP عمل می‌کند.

نتیجه گیری:

همین حالا جمله‌ی "asp.net vs php speed" را در google جستجو کنید. خواهید دید که در اکثر پست‌ها گفته شده که PHP از ASP.net سریعتر است اما دلیل بر این ادعا نخواهید یافت و فقط در حد حرف است. مشکل این است که اکثر مردم وقتی چیزی را زیاد می‌بینند یا زیاد می‌شنوند بدون آنکه دلیل بخواهند آنرا می‌پذیریند و حتی بعضی اوقات از آن نیز دفاع می‌کنند که واقعاً جای تاسف دارد.

توسعه وب بوسیله PHP کار خوبی است، بسیاری از اپلیکیشن‌ها و وبسایتهای شگفت انگیز توسط این زبان نوشته شده‌اند. اگر احساس می‌کنید PHP یک زبان برتر است از آن استفاده کنید اما این دلیل نمی‌شود که اطلاعات غلط را به دیگران القاء کنید و بدون دلیل و مدرک این زبان را از هر لحاظ برتر بدانید حال آنکه در این مقاله دیدیم که براساس چیزی که ارائه شد، ASP.net سرعت بیشتری نسبت به PHP دارد.

اگر با من در این امر موافق نیستید می‌توانید با نظرهای مستدل خود ما را راهنمایی کنید.

نظرات خوانندگان

نویسنده: سیروس
تاریخ: ۱۳۹۲/۰۶/۲۶ ۱۳:۱

به دلیل وسعت استفاده بیشتر از php و نیز استفاده سایت‌ها و شرکت‌های بزرگ از php خیلی‌ها فکر می‌کنند php بهتر و سریعتر از asp.net هست در حالیکه این وسعت استفاده با خاطر اوپن سورس و رایگان بودن php هست و چون وب سرور apache هم معمولاً رو لینوکس نصب می‌شود و خود لینوکس هم اپن سورس، تمام این دلایل دست به دست هم داده تا php بهتر به نظر بیاد. جدا از بحث سرعت اگر از لحاظ ساختاری بررسی کنیم php بیشتر یک زبان اسکریپتی است تا برنامه نویسی و ویژگی‌های زبان‌های خوب و شی گرا رو ندارد.

نویسنده: مسلم
تاریخ: ۱۳۹۲/۰۶/۲۶ ۱۳:۴

نمی‌دونم چرا ولی توی عمل واسه استارت زدن دات نت خیلی دیر می‌جنبه. حتی بعد از کامپایل و پابلیش یه خورده تاخیر داره ولی یکم که باهاش کار کنی می‌فته رو دور و خوب می‌شه! ولی پی‌اچ‌پی همون اول سریعه. شاید بخاط پیچیدگی فریم ورک هست. چرا که linq entity هم در پرژه استفاده شده است.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۶/۲۶ ۱۳:۲۰

چندین علت داره:

- پروسه کامپایل کدهای دات نت یک مرحله‌ای نیست. کلا در دات نت کدها به یک زبان میانی به نام IL ترجمه می‌شون. بعد این JIT compiler تبدیل به کدهای ماشین می‌شون. در ASP.NET این مساله هم برای کدهای پشت صحنه برنامه و هم برای خود صفحات رخ می‌ده. بنابراین برای اول مشاهده، روند این پروسه الزامی هست. ولی برای دفعات بعدی مشاهده خیر. البته روش برای پیش کامپایل کردن کامل صفحات هم وجود داره و یا در IIS های جدید یک سری مبحث warmup توکار پیش بینی شده.

- همچنین IIS برای مدیریت منابع سرور، یک سایت رو مدام در حافظه نگه نمی‌داره. فقط زمانیکه اولین درخواست به سرور میرسه سایت رو بارگذاری می‌کنه و application pool اون رو استارت. این هم یک زمان اولیه اندکی رو ممکنه به خودش اختصاص بده. بعلاوه پس از مدتی، یک سایت بیکار رو از حافظه خارج می‌کنه. بهش می‌گن ریسایکل کردن. در ASP.NET 4.0 به بعد امکان تنظیم auto-start برای سایت‌ها هست.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۶/۲۶ ۱۲:۳۰

به نظر من برای بحث در مورد PHP مقایسه سرعت در رده آخر اهمیت هست. مسایل بهتری برای بحث وجود دارند. مثلا:

- [بدترین زبانی که تا حال با آن کار کردید، کدام بود؟](#)

[The PHP Singularity](#) -

[PHP: a fractal of bad design](#) -

[PHP Sadness](#) -

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۲/۰۶/۲۶ ۱۳:۵۰

هر کسی به من گفت php سریع تر هست و یا ASP.NET سریع تر هست؛ من هم در جواب گفتم شما درست می‌فرمایید و هیچ گاه با آن‌ها بحث نکردم، هنوز که هنوزه من نمی‌فهمم که واقعاً دارید چی را با چی مقایسه کنید. ASP.NET و php کاملاً دو مقوله‌ی متفاوت هستند. اگر قرار است مقایسه‌ای در سرعت عمومی انجام شود، معقول‌تر است که در سطح فریم ورک هایی با قدرت برابر انجام شود؛ برای مثال سرعت عمومی Zend بالاتر است یا ASP.NET MVC باشد.

اصلاً بهتر است یه مقایسه با مستندات کافی برای شما مطرح کنم تا به کندی ASP.NET MVC پی‌برید:

هدف از این برنامه نمایش عبارت Hello,World در مرورگر است.

در php خام نوشتن کد زیر کفایت می‌کند:

```
echo 'Hello, World';
```

اما در ASP.NET MVC شما باید ابتدا یک کنترلر تعریف کرده سپس در یک Action عبارت Hello,World را بازگشت دهید. اگر این دو برنامه را اجرا کنید از سرعت فوق العاده‌ی php متحیر خواهید شد.

البته بگذریم از اینکه در ASP.NET سربارهای به نام Routing و ... در ابتدای کار وجود داره. نتیجه این هست که ASP.NET خیلی کنده و حرفی برای گفتن نداره در مقابل.php

از این دست مقایسه‌ها من هم زیاد دیدم. واقعاً خودشان هم نمی‌فهمند چی را با چی مقایسه می‌کنند.

این نوع مقایسه‌ها بیشتر منو یاد کسی می‌اندازه که گوشی موبایل خریده بود چهار هسته ای و می‌گفت چقدر تکنولوژی پیشرفت کرده که از لپتاپم دو هسته بیشتر داره و سریعتره!

من هم با خواندن این مقاله به جمله‌ی شما درست می‌فرمایید بسنده می‌کنم.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۶/۲۶ ۱۲:۵۷

مثال بی‌ربطه دوست عزیز. echo خام در PHP معادل هست با Response.Write خام در ASP.NET در حالتیکه در یک فایل ashx اجرا می‌شود. احتمالاً می‌دونی که این نوع فایل‌ها در حالت پیش فرض حتی ماژول سشن هم برآشون فعال نیست چه برسد به مسیریابی و در حداقل سربار کار می‌کنند.

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۲/۰۶/۲۶ ۱۳:۳

خب منظور من دقیقاً همین بود. یک مقایسه کاملاً بی‌ربط. یک فریم ورک با کلی امکانات را با یک زبان خام مقایسه کردم! پی نوشت: من در آینده نظر قبلی را دادم. ساعت 13:50

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۶/۲۶ ۱۳:۹

خوب، مثال‌های بحث جاری در ساده‌ترین حالت ممکن تهیه شدند. یعنی یک فریم ورک ASP.NET با کلی دم و دستگاه (ماژول سشن، ماژول مسیریابی، ماژول امنیت، ماژول اعتبارسنجی درخواست‌ها، ماژول فشرده سازی و ...) از یک سیستم ساده PHP سریعتر عمل می‌کنه. این چطور بی‌ربط به عنوان مقاله هست؟

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۲/۰۶/۲۶ ۱۳:۲۲

آخه من از منبع این مقاله چیزی نفهمیدم.

کلا منبع داره در مورد یه چیز دیگه صحبت میکنه. <http://www.wrensoft.com/zoom/benchmarks.html>

نوبنده: احسان
تاریخ: ۱۴:۱۵ ۱۳۹۲/۰۶/۲۶

فکر کنم خیلی مشخصه که یک زبان کامپیل شده چقدر میتوانه از یک واسط سریعتر باشه کاش با جاوا مقایسه میکردی

نوبنده: رضا منصوری
تاریخ: ۱۸:۲۱ ۱۳۹۲/۰۶/۲۶

توسعه وب بوسیله PHP کار خوبی است، بسیاری از اپلیکیشن‌ها و وبسایتهاش شگفت انگیز توسعه این زبان نوشته شده اند. اگر احساس می‌کنید PHP یک زبان برتر است از آن استفاده کنید اما این دلیل نمی‌شود که اطلاعات غلط را به دیگران القاء کنید و بدون دلیل و مدرک این زبان را از هر لحاظ برتر بدانید حال آنکه در این مقاله دیدیم که براساس چیزی که ارائه شد، ASP.net سرعت بیشتری نسبت به PHP دارد.

نوبنده: ناصر فرجی
تاریخ: ۲۲:۵۰ ۱۳۹۲/۰۶/۲۶

مطلوبی مشابه که چند روز پیش خوندم . لحظاتی یاد بحث‌های فروم برنامه نویس افتادم :)

<http://www.rezashirazi.com/post261.aspx>

نوبنده: آرایه
تاریخ: ۲۱:۳۷ ۱۳۹۲/۰۶/۲۷

به نظرم این بحث سود چندانی برای خوانندگان ندارد، benchmarchها بر مبنای کد مشابه هستند اما عملاً کد مشابه روی دو پلتفرم مختلف رو نمی‌شه مقایسه کرد.

مطلوبی در این خصوص نوشتمن: [اینجا](#)

نوبنده: آرمان فرقانی
تاریخ: ۸:۹ ۱۳۹۲/۰۶/۲۸

فناوری-زبان PHP بسیار محترم است و بسیار قابل توصیه اما برای آن زمانی که در رقابت با ASP کلاسیک بود و پدیده ای به نام دات نت ظهور نکرده بود حال آنکه پدیده یاد شده در زمان حال به پختگی و پیشرفت چشمگیری دست یافته است. دنیای دات نت و مباحث مربوط به آن گسترددهتر و پیچیدهتر از آن است که برای توجیه استفاده از دات نت سرعت مقایسه شود. دهها ویژگی منحصر به دات نت وجود دارد که برای انتخاب فناوری سمت سرور مجالی برای تفکر درباره سرعت باقی نمی‌گذارد و آنان که اهل تفکر باشند را جذب خود می‌کند و نه گمراهان (کسانی که یا تعصب دارند یا خسته‌تر از آن هستند که دنیای جدیدی را تجربه نمایند).

در مورد سرعت کافی است نکات ساده ای که چند دقیقه بیشتر زمان نمی‌برند رعایت شود تا وب سایت‌های دات نت چندین برابر سریعتر عمل کنند.

فراموش نکنید دنیای دات نت تا حد بیشتری با اصول مهندسی نرم افزار گره خورده است و باب میل همگان نیست. عموماً سرویس دهنده بر حسب فناوری انتخاب می‌شود و نه بر عکس! در مقالاتی که مقایسه انجام می‌دهند صحبت از رایگان بودن لینوکس و آپاچی و ... چندان ضرورتی ندارد.

بسیاری از آن‌ها که بودن Open Source را با آب و تاب و به عنوان برتری مطرح می‌کردند هرگز در عمر خود به کدهای سورس آن نگاهی نکرده اند. البته اگر بداند از کجا باید دانلود کنند. توصیه می‌کنم در مورد رویکرد و سیاست‌های چند سال اخیر مایکروسافت در رابطه با Open Source تحقیق بیشتری صورت گیرد. مثل هایی از سایت‌های موفق و یا تعداد سایت‌ها در یک فناوری هرگز دلیلی برای انتخاب فناوری نیست. ضمناً زمان ظهور این دو فناوری یکسان نبوده است که تعداد وب سایت‌ها معیار مقایسه باشد. فناوری-زبان PHP هنوز وجود دارد. هنوز قدرت خود را دارد. و هنوز هر کجا به هر دلیلی امکان استفاده از دات نت نبود، با کمال

نویسنده: نیما
تاریخ: ۷:۱۸ ۱۳۹۲/۰۶/۳۰

با سلام خدمت دوستان گرامی

نکته ای که باید به آن دقت داشت، فرق بین مجانية و سورس باز است. لطفاً این موارد را از پکدیگر جدا نمایید.

نکته دوم هم اینست که چند نفر در کشور خودمون داریم که میتوانند سورس PHP، لینوکس و ... را سفارشی کنند؟

نویسنده: حامد وهابی املشی
تاریخ: ۱۴:۸ ۱۳۹۲/۰۷/۰۱

چرا در نمودار جاوا اسکریپت توی نمودار قرار گرفته؟ اونها زبانهایی سمت سرورند، جاوا اسکریپت سمت کلاینت. البته ممکنه منظورش جاوا اسکریپت سمت سرور باشه! در این صورت هم باز اشتباهه، چون جاوا اسکریپت سمت سرور مربوط به تکنولوژی ASP کلاسیک بوده که توی نمودار اون رو هم آوردن. ASP کلاسیک هم وبی اسکریپت داشت و هم جاوا اسکریپت.

نویسنده: محسن خان
تاریخ: ۱۴:۱۷ ۱۳۹۲/۰۷/۰۱

منظورش پیاده سازی خاص خودشون بوده.

نویسنده: حامد وهابی املشی
تاریخ: ۱۵:۱۹ ۱۳۹۲/۰۷/۰۱

بلاخره پیاده سازی خودش، سمت سرور بوده؟

نویسنده: محسن خان
تاریخ: ۱۸:۸ ۱۳۹۲/۰۷/۰۱

بله. ضمناً node.js یک فناوری دیگر سمت سرور مبتنی بر جاوا اسکریپت است (و از این دست دارن زیاد میشن)

نویسنده: علی یگانه مقدم
تاریخ: ۵:۱۰ ۱۳۹۳/۰۹/۱۴

البته این مورد هم هست که بسیاری از شرکت‌ها یا افراد که از php استفاده می‌کنن به خاطر عدم وابستگی‌شون به یک شرکت خاص چون مایکروسافت هم هست مثل گوگل که کلا استفاده از برنامه‌های شرکت مایکروسافت رو در شرکتش ممنوع اعلام کرده، کلا عدم وابستگی پی اچ پی و همچنین متن باز بودنش برای به روزآوری و تغییراتش توسط برنامه نویسان سراسر جهان بیشتر مورد توجه قرار گرفته.

البته این نکته هم هست که اکثر مردم با دیدن آدرس‌های بدون پسوند فکر می‌کنن که php هست مثلاً همین stack overflow رو خیلی‌ها فکر می‌کنن با php نوشتن.

البته بودن بیشتر سیستم‌های آماده از نوع php و همچنین ارزونتر بودن هاست‌های لینوکس به خصوص در ایران هم سبب این اتفاق بوده.

در ادامه مباحث بررسی اجرای برنامه‌های دات نت بر روی لینوکس، قصد داریم برنامه‌های ASP.NET را به کمک Mono 3.0 و یک وب سرور لینوکسی، بر روی Ubuntu [اجرا کنیم](#).

پیشنبازها

دو پروژه خالی ASP.NET MVC و ASP.NET Web forms را در VS.NET تحت ویندوز ایجاد نمائید. آن‌ها را یکبار کامپایل کرده و اجرا کنید. سپس فایل‌های آن‌ها را به ubuntu منتقل کنید (پوشه‌های bin پروژه‌ها فراموش نشوند؛ خصوصاً نگارش MVC که به همراه یک سری کتابخانه جانبی است).

برای انتقال فایل‌ها به لینوکس، اگر از VMWare workstation Ubuntu استفاده می‌کنید، کپی و paste مستقیم فایل‌ها از ویندوز به درون ماشین مجازی لینوکس پشتیبانی می‌شود.

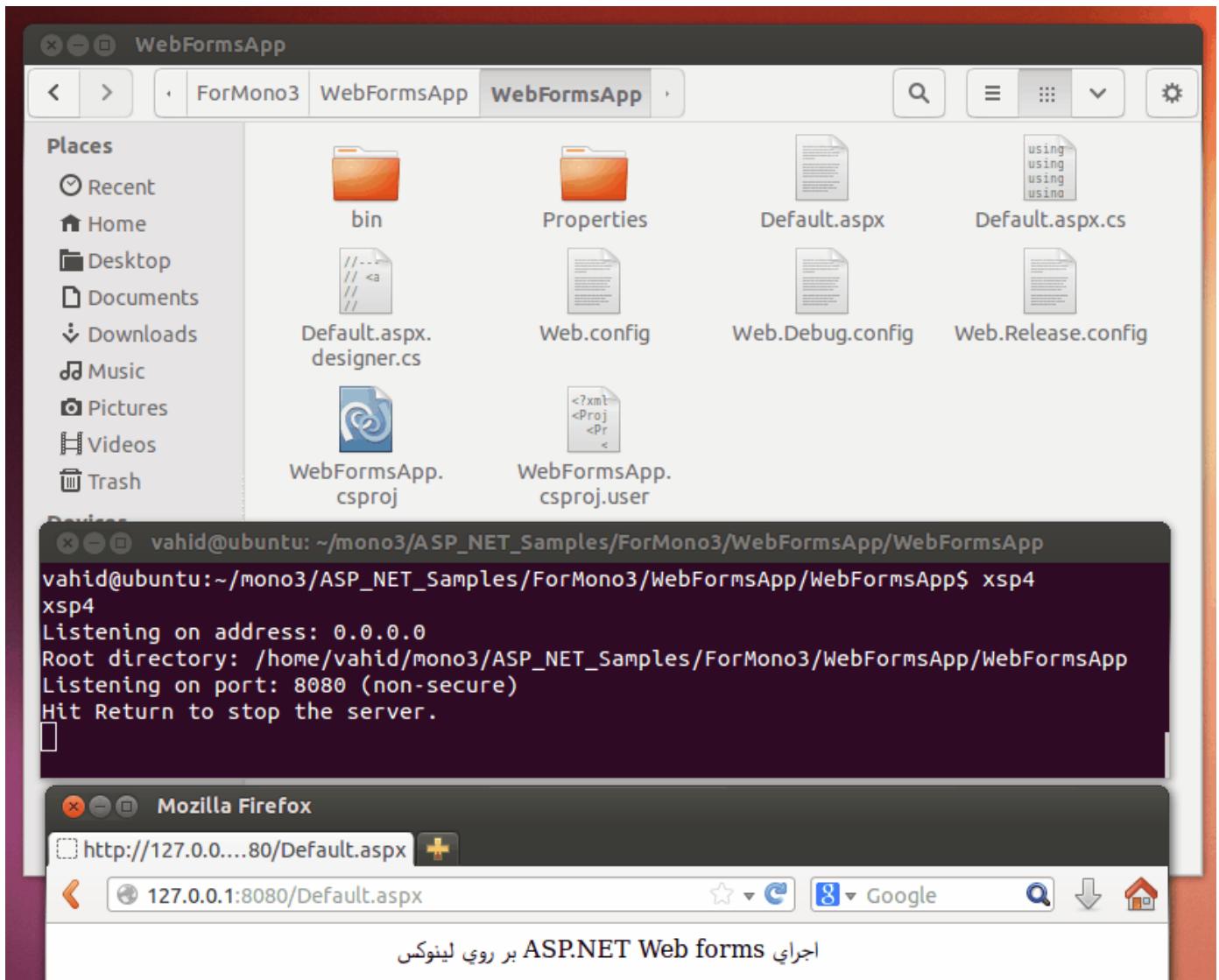
نصب وب سرور آزمایشی مونو یا XSP

اگر نیاز به یک وب سرور آزمایشی، چیزی شبیه به وب سرور توکار VS.NET داشتید، پروژه [XSP](#) جهت این نوع آزمایشات ایجاد شده است.

پس از نصب آن (که به همراه همان بسته PPA [قسمت قبل](#)، هم اکنون بر روی سیستم شما نصب است)، در ترمینال لینوکس، با استفاده از دستور cd به ریشه وب سایت خود وارد شوید، سپس دستور xsp4 را اجرا کنید تا وب سرور xsp4 مشغول هاست سایت شما شود (برای اجرا در مسیر /opt/bin/xsp4 نصب شده است).

اجرای برنامه 4 ASP.NET Web forms توسط XSP

بدون هیچ مشکل خاصی در همان ابتدای کار اجرا شد (البته باید دقت داشت که لینوکس به کوچکی و بزرگی حروف حساس است. یعنی حتماً باید Default.aspx وارد شود و نه :default.aspx).



اجرای برنامه ASP.NET MVC 4 توسط XSP

اجرا نشد! پیام می‌دهد که

"Missing method System.Web.Security.FormsAuthentication::get_IsEnabled() in assembly System.Web.dll"

و یا

Compiler Error Message: CS1703: An assembly with the same identity `mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089` has already been imported. Consider removing one of the references

علت اینجا است که XSP4 همراه با نسخه PPA، قدیمی است. بنابراین باید نسخه اصلی را از مخزن کد آن دریافت و کامپایل کنیم. پیشنبازهای اینکار مانند Git و Mono، در قسمت قبل دریافت شدند. سپس فرآمین ذیل را در خط فرمان لینوکس اجرا کنید:

```
git clone git://github.com/mono/xsp.git
cd xsp
./autogen.sh --prefix=/opt
make
sudo make install
```

پس از کامپایل، اگر این نگارش جدید را اجرا کنید، به خطای ذیل خواهد رسید:

```
System.IO.FileNotFoundException: Could not load file or assembly XSP, Version=3.0.0.0
```

برای رفع این مشکل باید اینبار وب سرور جدید را با فرمان `sudo` یا دسترسی مدیریتی اجرا کنید تا مشکل برطرف شود.
البته من [سورس دریافت شده را](#) در خود `monodevelop` کامپایل کردم (فایل `sln` آن را در `monodevelop` باز کرده و پروژه را `xsp/src/Mono.WebServer.XSP/bin/Debug` در پوشش `Mono.WebServer.XSP.exe` و `Mono.WebServer.dll` می‌شوند).

یکی دیگر از دلایل ظاهر شدن خطای فوق، نیاز به نصب این دو فایل در GAC است که به نحو زیر قابل انجام می‌باشد:

```
cd xsp/src/Mono.WebServer.XSP/bin/Debug  
sudo gacutil -i Mono.WebServer.XSP.exe  
sudo gacutil -i Mono.WebServer.dll
```

بعد این دو فایل `dll` و `exe` را در پوشش برنامه MVC خود کپی کنید و سپس دستور ذیل را اجرا نمایید:

```
cd myMvcAppPath  
sudo mono Mono.WebServer.XSP.exe
```

اینبار وب سرور جدید، روی پورت 9000 شروع به کار می‌کند. اکنون اگر در فایرفاکس آدرس `http://localhost:9000` را باز کنید، برنامه اجرا شده اما به خطای ذیل خواهد رسید:

```
CS0234: The type or namespace name `Helpers' does not exist in the namespace `System.Web'.  
Are you missing an assembly reference?
```

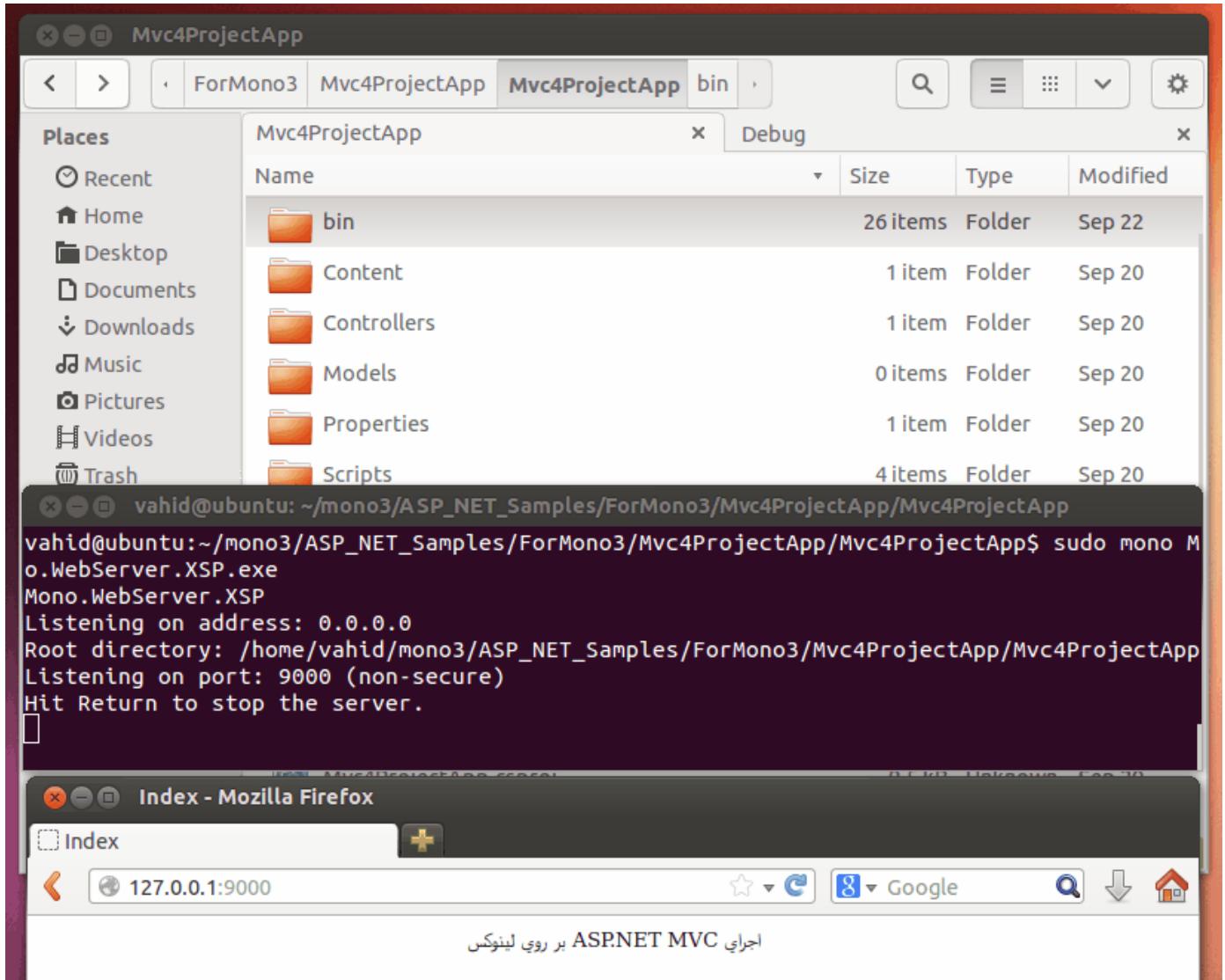
برای رفع این مشکل باید اندکی فایل `web.config` برنامه را ویرایش کرد:

```
<system.web>  
  <compilation debug="true" targetFramework="4.0">  
    <assemblies>  
      <add assembly="System.Web.Helpers, Version=2.0.0.0, Culture=neutral,  
      PublicKeyToken=31BF3856AD364E35" />  
      <add assembly="System.Web.Mvc, Version=4.0.0.0, Culture=neutral,  
      PublicKeyToken=31BF3856AD364E35" />  
    </assemblies>  
  </compilation>
```

سعی بعدی ... اجرا نشد! با هر بار `refresh` صفحه یک خطای جدید نمایش می‌داد که ... `Type` خاصی را نمی‌تواند بارگذاری کند (به همراه نام اسمبلی مربوطه). برای رفع این مشکل `dll`‌های ذیل را از پوشش `bin` پروژه MVC خود که از ویندوز به لینوکس کپی کرده‌اید، حذف کنید:

```
Microsoft.Web.Infrastructure.dll  
System.Net.Http.dll  
System.Net.Http.Formatting.dll  
System.Web.Http.dll  
System.Web.Http.WebHost.dll
```

این فایل‌ها توسط تیم Mono به صورت مستقل پیاده سازی شده‌اند و نیازی نیست تا از ویندوز به لینوکس کپی شوند.
بعد از حذف این فایل‌های اضافی، برنامه ASP.NET MVC نیز اجرا شد:



چند نکته تکمیلی

- نحوه تشخیص موجود بودن یک DLL خاص، در نگارش جاری Mono نصب شده:

```
$ gacutil -l Microsoft.Web.Infrastructure
The following assemblies are installed into the GAC:
Microsoft.Web.Infrastructure, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35
Number of items = 1
```

- اگر می‌خواهید مطمئن شوید که تمام اسembly‌های موجود در GAC درست نصب شده‌اند یا خیر، فرمان ذیل را اجرا کنید:

```
cd /opt/lib/mono/gac # assuming this is your main gac
sudo find . */*.dll -exec gacutil -i '{}' \;
```

- در نسخه لینوکسی System.Web ممکن است یک سری از فضاهای نام هنوز موجود نباشند. لیست آن‌ها را در این آدرس می‌توانید مشاهده کنید:

<http://go-mono.com/status/status.aspx?reference=4.0&profile=4.0&assembly=System.Web>

نظرات خوانندگان

نویسنده: علی صداقت
تاریخ: ۱۳۹۲/۰۷/۰۱ ۱۷:۲۲

جناب نصیری، دیتابیس رو چه کنیم؟ واقعا جانشین منبع بازی، همسان با sql server در این طرف داریم؟ MySQL کفايت می‌کنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۰۱ ۱۷:۵۶

MySQL هم خوبه (البته مجوز استفاده اش رو هم باید [مد نظر داشته باشید](#)). هر چند از زمانیکه به تملک اوراکل درآمده، خیلی از شرکت‌ها دارند به سمت MariaDB کوچ می‌کنند: [^](#) و [^](#) Oracle + [نسخه لینوکسی](#) هم دارد. - و برای کارهای سبک SQLite [نسخه لینوکسی](#) دارد.

نویسنده: محمد جواد تواضعی
تاریخ: ۱۳۹۲/۰۷/۰۲ ۱:۱۲

شما می‌توانید از postgresql استفاده کنید . کاملا متن باز بوده و چیزی از Ms SqlServer و oracle کم ندارد .

عنوان:	اجرای برنامه‌های ASP.NET به کمک وب سرور Apache توسط Mono در Ubuntu
نویسنده:	وحید نصیری
تاریخ:	۱۳۹۲/۰۷/۰۲ ۸:۰
آدرس:	www.dotnettips.info
گروه‌ها:	ASP.NET, MVC, Mono, Linux

در قسمت قبل نحوه اجرای برنامه‌های ASP.NET را به کمک وب سرور آزمایشی Mono بررسی کردیم. بدیهی است این وب سرور برای هاست کردن سایتها و خدمات دهی عمومی، مناسب نیست و صرفاً جهت کارهای برنامه نویسی طراحی شده است. در ادامه، ابتدا وب سرور معروف آپاچی را نصب کرده و سپس سعی خواهیم کرد تا همان دو برنامه ASP.NET Web forms و ASP.NET MVC قسمت قبل را بر روی آن اجرا کنیم (دو برنامه خالی ASP.NET کامپایل شده که از ویندوز به لینوکس کپی شده بودند).

نصب وب سرور آپاچی

به اینترنت متصل شده، ترمینال Ubuntu را گشوده (با میانبر `ctrl+alt+t`) و سپس فرمان‌های ذیل را صادر کنید:

```
sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get install apache2
```

به این ترتیب Apache web server بر روی سیستم نصب خواهد شد.

نصب ماژول‌های ASP.NET مخصوص آپاچی

سپس نیاز است ASP.NET runtime for Apache را نصب کنیم:

```
sudo apt-get install libapache2-mod-mono mono-apache-server4
```

درون پروسه آپاچی اجرا شده و تمام درخواست‌های رسیده به یک وب سایت ASP.NET را به `mod_mono` که کار هاست سایت را انجام می‌دهد، انتقال خواهد داد. اگر علاقمند به مشاهده تنظیمات آن بودید باید به مسیر `etc/apache2/mods-enabled` مراجعه کرده و فایل `mod_mono.conf` را بررسی کنید (اختیاری). برای مثال، در آن حالت اجرا، بر روی 4 ASP.NET تنظیم شده است.

تنظیمات آپاچی برای کار با ASP.NET

نیاز است فایل تنظیمات پیش فرض وب سرور آپاچی را جهت معرفی ASP.NET به آن، اندکی ویرایش کنیم:

```
sudo gedit /etc/apache2/sites-available/default
```

در ادیتور باز شده، ذیل قسمت `ServerAdmin`، موارد زیر را اضافه کنید:

```
MonoAutoApplication disabled
AddHandler mono .aspx .ascx .asax .ashx .config .cs .asmx .axd
MonoApplications "/:/var/www"
MonoServerPath default "/usr/bin/mod-mono-server4"
```

در ادامه به المان Directory /var/www سطرهای زیر را اضافه کنید:

```
SetHandler mono
DirectoryIndex index.aspx index.html default.aspx Default.aspx
```

المان جدید ذیل را در انتهای فایل اضافه کنید تا یک سری از پوشش‌های خاص ASP.NET همانند محیط‌های ویندوزی در آپاچی نیز محافظت شوند:

```
<DirectoryMatch "/([bB]in|[Aa]pp_[Cc]ode|[Aa]pp_[Dd]ata|[Aa]pp_[Gg]lobal[Rr]esources|[Aa]pp_[Ll]ocal[Rr]esources)"/>
```

```
Order deny,allow  
Deny from all  
</DirectoryMatch>
```

فایل را ذخیره کرده و ادیتور را بیندید. اکنون وب سرور آپاچی را اجرا و ری استارت کنید:

```
sudo service apache2 restart
```

همچنین سطح دسترسی مسیر var/www که حاوی برنامه‌های ما خواهد بود نیز باید تغییر کند:

```
# r,w,exec for user + group  
sudo chgrp -R www-data /var/www  
sudo chown -R www-data /var/www  
sudo chown -R www-data:www-data /var/www  
sudo chmod -R 755 /var/www
```

اکنون اگر آدرس http://127.0.0.1 را در مرورگر وارد کنید، باید پیام «It Works» را مشاهده کنید و اگر خواستید محتوای آن را ویرایش کنید، دستور ذیل را صادر نمائید:

```
gedit /var/www/index.html
```

اجرای مثال‌ها

با تنظیمات فوق، برنامه‌های کپی شده در مسیر var/www به کمک مونو و آپاچی اجرا خواهند شد.
دسترسی‌های فعلی کاربر وارد شده به لینوکس اجازه کپی فایل‌ها را به مسیر var/www نمی‌دهد. همچنین می‌خواهیم این کارها را توسط File browser آن انجام دهیم و نه خط فرمان. برای این منظور دستور ذیل را اجرا کنید تا آن با دسترسی مدیریتی اجرا شود:

```
sudo nautilus
```

الان می‌توان بدون مشکل در همان File browser گرافیکی آن، پوشه مثال وب فرم‌های خود را به درون پوشه var کپی کرد.
پس از کپی دو برنامه Web forms و MVC ابتدای بحث، نیاز است مجدداً فایل تنظیمات آپاچی را ویرایش کنیم:

```
sudo gedit /etc/apache2/sites-available/default
```

اکنون تنظیم برنامه وب فرم را به نحو ذیل به این فایل اضافه کنید:

```
AddMonoApplications default "/webforms_test:/var/www/WebFormsApp/"  
<Location /webforms_test>  
SetHandler mono  
</Location>
```

در تنظیم AddMonoApplications نامی است که در مرورگر وارد خواهد شد و قسمت webforms_test نامی است که در مرورگر وارد خواهد شد و قسمت بعد از : مسیر فایل‌های برنامه و ریشه سایت است.
و تنظیم برنامه MVC به صورت زیر می‌باشد:

```
AddMonoApplications default "/mvc_test:/var/www/Mvc4ProjectApp/"  
<Location /mvc_test>  
SetHandler mono  
</Location>
```

بعد از هر بار تغییر فایل تنظیمات آپاچی باید یکبار و ب سرور را توسط دستور sudo service apache2 restart ری استارت کنیم.

اکنون برنامه وب فرم‌ها در مسیر `http://127.0.0.1/mvc_test` و برنامه MVC در مسیر `http://127.0.0.1/webforms_test` قابل دسترسی است.

خلاصه بحث

پس از نصب وب سرور آپاچی و مازول مونوی مخصوص آن، فایل `etc/apache2/sites-available/default` را به نحو ذیل ویرایش کنید و مسیرهای برنامه‌های خود را در آن تعریف نمائید:

```
<VirtualHost *:80>
ServerAdmin webmaster@localhost
    MonoAutoApplication disabled
    AddHandler mono .aspx .ascx .asax .ashx .config .cs .asmx .axd
    MonoApplications "/:/var/www"
    MonoServerPath default "/usr/bin/mod-mono-server4"

DocumentRoot "/var/www"
<Directory />
Options FollowSymLinks
AllowOverride None
</Directory>
<Directory /var/www/>
    SetHandler mono
    DirectoryIndex index.aspx index.html default.aspx Default.aspx
Options Indexes Includes FollowSymLinks MultiViews
AllowOverride None
Order allow,deny
allow from all
</Directory>

ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
AllowOverride None
Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
Order allow,deny
Allow from all
</Directory>

    <DirectoryMatch
"/([bB]in|[Aa]pp_[Cc]ode|[Aa]pp_[Dd]ata|[Aa]pp_[Gg]lobal[Rr]esources|[Aa]pp_[Ll]ocal[Rr]esources)/">
        Order deny,allow
        Deny from all
    </DirectoryMatch>

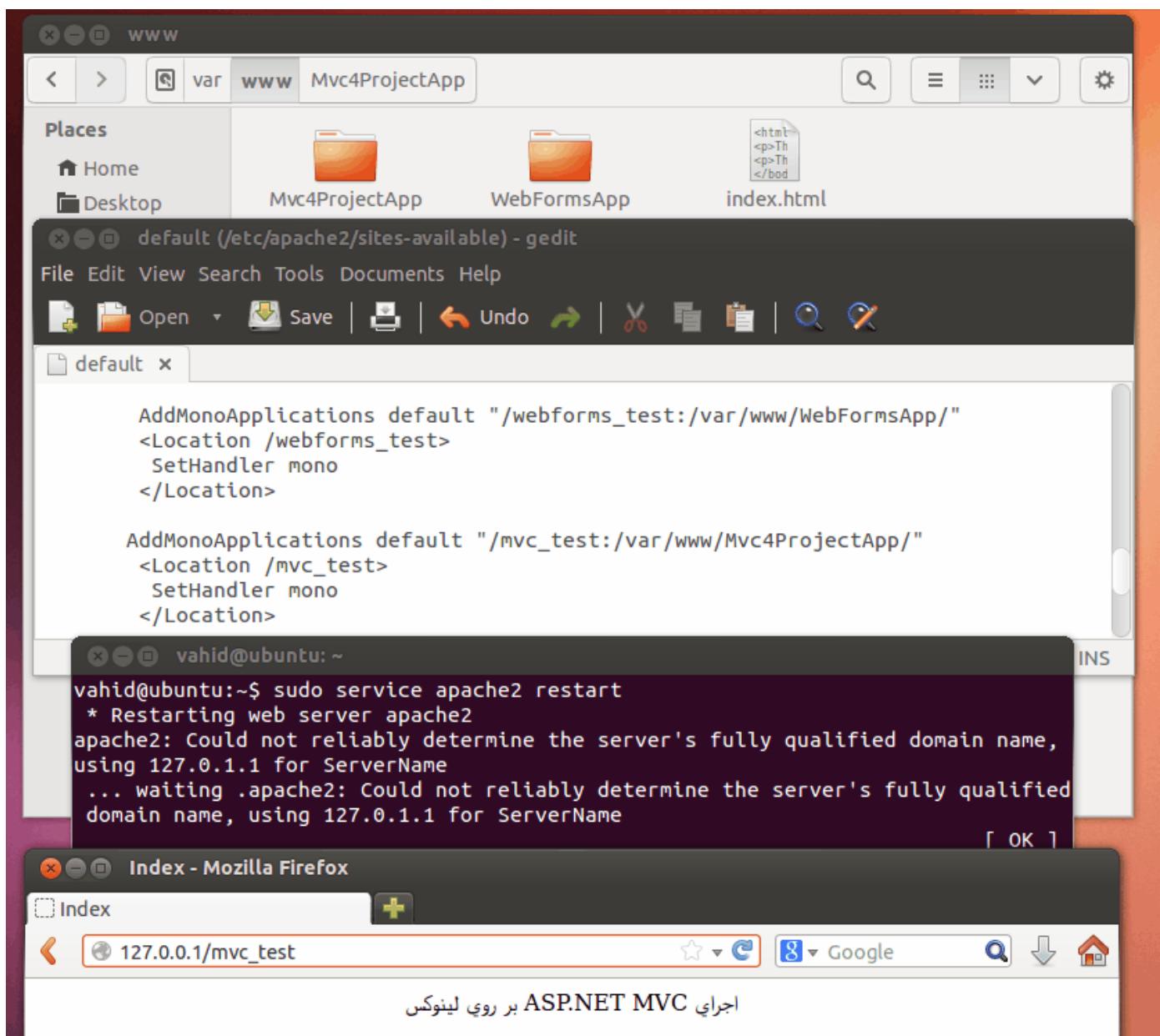
    AddMonoApplications default "/webforms_test:/var/www/WebFormsApp/"
    <Location /webforms_test>
        SetHandler mono
    </Location>

    AddMonoApplications default "/mvc_test:/var/www/Mvc4ProjectApp/"
    <Location /mvc_test>
        SetHandler mono
    </Location>

ErrorLog ${APACHE_LOG_DIR}/error.log

# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn

CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```



نظرات خوانندگان

نویسنده: مهدی پایروند
تاریخ: ۹:۴۷ ۱۳۹۲/۰۷/۰۳

در مورد هاست کردن سرویس‌های WCF هم اگه ممکنه توضیح بدین.

نویسنده: وحید نصیری
تاریخ: ۱۴:۳ ۱۳۹۲/۰۷/۰۴

الف) تنظیمات سرور

برای اضافه کردن تنظیمات WCF، ابتدا فایل mod_mono.conf را باز کنید:

```
sudo gedit /etc/apache2/mods-available/mod_mono.conf
```

بعد در سطر اول آن پسوند svc را هم اضافه کنید:

```
AddType application/x-asp-net .svc .aspx .ashx .asmx .ascx .asax .config .ascx
```

و همچنین اینکار را برای فایل default آپاچی انجام دهید:

```
sudo gedit /etc/apache2/sites-available/default
```

و پسوند svc را در اینجا نیز اضافه نمائید:

```
AddHandler mono .svc .aspx .ascx .asax .ashx .config .cs .asmx .axd
```

بعد از آن یکبار سرور را با دستور sudo service apache2 restart ری استارت کنید.

ب) تنظیمات وب کانفیگ

در اینجا serviceMetadata را طوری تنظیم کنید تا با HTTP GET قابل دریافت باشد

```
<system.serviceModel>
  <behaviors>
    <serviceBehaviors>
      <behavior>
        <serviceMetadata httpGetEnabled="true" httpGetUrl="wsdl"/>
      </behavior>
    </serviceBehaviors>
```

بعد از آن سرویس شما در آدرس http://127.0.0.1/webforms_test/Service1.svc/js یا در آدرس http://127.0.0.1/webforms_test/Service1.svc/jsdebug برای کارهای Ajax ای قابل استفاده خواهد بود (فرض بر این بود که یک VS.NET Ajax Enabled WCF Service را در لینوکس منتقل کردید).

```
WebFormsApp.Service1._staticInstance.set_userContext(value), }  
WebFormsApp.Service1.get_defaultUserContext = function() { return  
WebFormsApp.Service1._staticInstance.get_defaultUserContext(); }  
WebFormsApp.Service1.set_defaultSucceededCallback = function(value) {  
WebFormsApp.Service1._staticInstance.set_defaultSucceededCallback(value); }  
WebFormsApp.Service1.get_defaultSucceededCallback = function() { return  
WebFormsApp.Service1._staticInstance.get_defaultSucceededCallback(); }  
WebFormsApp.Service1.set_defaultFailedCallback = function(value) {  
WebFormsApp.Service1._staticInstance.set_defaultFailedCallback(value); }  
WebFormsApp.Service1.get_defaultFailedCallback = function() { return  
WebFormsApp.Service1._staticInstance.get_defaultFailedCallback(); }  
WebFormsApp.Service1.set_path("http://127.0.0.1/webforms_test/Service1.svc");  
WebFormsApp.Service1.DoWork= function(onSuccess,onFailed,userContext)  
{WebFormsApp.Service1._staticInstance.DoWork(onSuccess,onFailed,userContext); }
```

نویسنده: ایمان اسلامی
تاریخ: ۸:۲۷ ۱۳۹۲/۱۲/۲۰

میخواستم بدونم در شرایط مساوی نصب یک برنامه asp.net روی iis از نظر performance بهتره یا روی apache

می دانیم بهینه سازی موتورهای جستجو (به انگلیسی: Search engine optimization (SEO)) که گاهی در فارسی به آن سئو نیز گفته می شود، عملیاتی است برای بهبود دید یک وب گاه یا یک صفحه وب، در صفحه نتایج موتورهای جستجو که می تواند طبیعی و یا الگوریتمی باشد. این عملیات برای وب مسترها یکی از عوامل مهم و حیاتی بست آوردن کاربران جدید از موتورهای جستجو است.

اگر چک لیست های SEO وب سایت ها را مشاهده کنیم، می توانیم آنها را در دو دسته کلی [بهینه سازی درونی و برونوی وب سایت](#) در نظر بگیریم :

Off-Page Optimization یا برونوی ، که بیشتر بر دوش مشاوران سئو و خود مدیران وب سایت است . ، فعالیت در شبکه اجتماعی و ...

On-Page Optimization یا درونی که بخش های مهمی از آن وظیفه مابر نامه نویس ها است . (H1 Tag , URL Naming) و اما در حوزه [الگوریتم جدید گوگل](#) (و +) به حساب می آید بر عهده مشاوران سئو و خود مدیران وب سایت می باشد] و همچنین [Content Optimization](#) که مهمترین عامل در الگوریتم های نسل جدید موتورهای جستجو [

در ادامه به ارائه چند راهکار جهت بهینه سازی برنامه های وب ASP.NET مان برای موتورهای جستجو می پردازیم:

1. متدهای برای ایجاد عنوان سایت

```
private const string SeparatorTitle = " - ";
private const int MaxLengthTitle = 60;
public static string GeneratePageTitle(params string[] crumbs)
{
    var title = "";
    for (int i = 0; i < crumbs.Length; i++)
    {
        title += string.Format(
            "{0}{1}",
            crumbs[i],
            (i < crumbs.Length - 1) ? SeparatorTitle : string.Empty
        );
    }
    title = title.Substring(0, title.Length <= MaxLengthTitle ? title.Length : MaxLengthTitle).Trim();
    return title;
}
```

نکته :

MaxLengthTitle پیشنهادی برای عنوان سایت 60 می باشد.

2. متدهای برای ایجاد متاتگ صفحات سایت

```
public enum CacheControlType
{
    [Description("public")]
    public,
    [Description("private")]
    private,
    [Description("no-cache")]
    nocache,
    [Description("no-store")]
    nostore
}
```

```

}

private const int MaxLengthTitle = 60;
private const int MaxLengthDescription = 170;
private const string FaviconPath = "~/cdn/ui/favicon.ico";
public static string GenerateMetaTag(string title, string description, bool allowIndexPage, bool
allowFollowLinks, string author = "", string lastmodified = "", string expires = "never", string
language = "fa", CacheControlType cacheControlType = CacheControlType._private)
{
    title = title.Substring(0, title.Length <= MaxLengthTitle ? title.Length :
MaxLengthTitle).Trim();
    description = description.Substring(0, description.Length <= MaxLengthDescription ?
description.Length : MaxLengthDescription).Trim();

    var meta = "";
    meta += string.Format("<title>{0}</title>\n", title);
    meta += string.Format("<link rel=\"shortcut icon\" href=\"{0}\"/>\n", FaviconPath);
    meta += string.Format("<meta http-equiv=\"content-language\" content=\"{0}\"/>\n", language);
    meta += string.Format("<meta http-equiv=\"content-type\" content=\"text/html; charset=utf-
8\"/>\n");
    meta += string.Format("<meta charset=\"utf-8\"/>\n");
    meta += string.Format("<meta name=\"description\" content=\"{0}\"/>\n", description);
    meta += string.Format("<meta http-equiv=\"Cache-control\" content=\"{0}\"/>\n",
EnumExtensions.EnumHelper<CacheControlType>.GetEnumDescription(cacheControlType.ToString()));
    meta += string.Format("<meta name=\"robots\" content=\"{0}, {1}\\" />\n", allowIndexPage ?
"index" : "noindex", allowFollowLinks ? "follow" : "nofollow");
    meta += string.Format("<meta name=\"expires\" content=\"{0}\"/>\n", expires);

    if (!string.IsNullOrEmpty(lastmodified))
        meta += string.Format("<meta name=\"last-modified\" content=\"{0}\"/>\n", lastmodified);

    if (!string.IsNullOrEmpty(author))
        meta += string.Format("<meta name=\"author\" content=\"{0}\"/>\n", author);

    //-----Google & Bing Doesn't Use Meta Keywords ...
    //meta += string.Format("<meta name=\"keywords\" content=\"{0}\"/>\n", keywords);

    return meta;
}

```

چند نکته :

MaxLengthDescription پیشنهادی برای متانگ توضیح سایت 170 می باشد.

آشنایی با متانگ ها (Meta tags) و کاربرد آنها در صفحات وب (HTML)

برای کاربرد allowFollowLinks و allowIndexPage هم می توانید به لینک بالا و بررسی متانگ robots بپردازید.
با توجه به اهمیت شبکه های اجتماعی متانگ های شبکه های اجتماعی (+ و +) را هم نباید از قلم انداخت.
برای دریافت Description نوع سفارشی CacheControlType از پروژه متدهای الحاقی علیرضا اسم رام استفاده کردم.

3. متدهای برای ایجاد Slug (اسلالگ آدرسی با مفهوم برای بکار بردن در URL ها است که دوستدار موتورهای جستجو می باشد)

```

private const int MaxLengthSlug = 45;
public static string GenerateSlug(string title)
{
    var slug = RemoveAccent(title).ToLower();
    slug = Regex.Replace(slug, @"[^a-z0-9-\u0600-\u06FF]", "-");
    slug = Regex.Replace(slug, @"\s+", "-").Trim();
    slug = Regex.Replace(slug, @"\-", "-");
    slug = slug.Substring(0, slug.Length <= MaxLengthSlug ? slug.Length : MaxLengthSlug).Trim();

    return slug;
}

private static string RemoveAccent(string text)
{
    var bytes = Encoding.UTF8.GetBytes(text);
    return Encoding.UTF8.GetString(bytes);
}

```

: نکته

MaxLengthSlug پیشنهادی برای عنوان سایت 45 می باشد.

نمونه ای از کاربرد توابع :

```
Head.InnerHtml = SEO.GenerateMetaTag
    (
        title: SEO.GeneratePageTitle(".NET Tips", "ASP.NET MVC
#1"),
        description: "آرشیو مطالب "ASP.NET MVC" با وجود فریم ورک پخته ای به نام ASP.NET web
        مطرح می شود این است: «برای چی؟». بنابراین تا به این سؤال اولین سؤالی که حین سوئیچ به
        ".پاسخ داده نشود، هر نوع بحث فنی در این مورد بی فایده است
        allowIndexPage: true,
        allowFollowLinks: true,
        author: "وحید نصیری",
        cacheControlType: SEO.CacheControlType._private
    );
```

: Page Source و خروجی در

```
<title>.NET Tips - آرشیو مطالب - ASP.NET MVC #1</title>
<link rel="shortcut icon" href="../../cdn/images/ui/favicon.ico"/>
<meta http-equiv="content-language" content="fa"/>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
<meta charset="utf-8"/>
<meta name="description" content="آرشیو مطالب "ASP.NET MVC" با وجود فریم ورک پخته ای به نام ASP.NET web forms,
        مطرح می شود این است: «برای چی؟». بن اولین سؤالی که حین سوئیچ به
        ..."/>
<meta http-equiv="Cache-control" content="private"/>
<meta name="robots" content="index, follow" />
<meta name="expires" content="never"/>
<meta name="author" content="وحید نصیری"/>
```

موفق باشید

نظرات خوانندگان

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۹/۰۸ ۱۵:۲۰

دوست من با این روش

```
Head.InnerHtml = SEO.GenerateMetaTag(....)
```

در صورتی که در قسمت head صفحه استایل یا متاتگهای دیگری تعریف شده باشد همگی حذف خواهد شد.

نویسنده: محسن نجف زاده
تاریخ: ۱۳۹۲/۰۹/۰۸ ۱۸:۴۵

بله صابر جان حتما همین طور است (برای مثال بیان شده است).
خودم به این صورت استفاده می کنم :

```
<head>
    <asp:Literal runat="server" ID="litHead"></asp:Literal>
    ....
</head>
```

و در قسمت code - behind

```
litHead.Text = SEO.GenerateMetaTag(...)
```

پ ن : چرا از کنترل Literal استفاده شده و از Label استفاده نشده ؟ به این دلیل که خروجی متن رندر شده Label توسط تگ <label> یا احاطه می شود. (+)

نویسنده: پوریا منفرد
تاریخ: ۱۳۹۲/۱۰/۱۴ ۱۲:۱۵

با سلام

امکان داره این خط کلاسش رو هم بزارید بnde دقیقا نتونستم با متدهای Label که گفتین این قسمت رو پیاده سازی کنم

```
EnumExtensions.EnumHelper<CacheControlType>.GetEnumDescription(cacheControlType.ToString()));
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۱۴ ۱۲:۲۳

مراجعه کنید به پروژه [DNT.Extensions](#) و همچنین [اینجا](#)

نویسنده: پوریا منفرد
تاریخ: ۱۳۹۲/۱۰/۱۴ ۱۳:۲۸

بنده یه مشکلی دارم : از MasterPage استفاده می کنم در ASP.net Webform و فرمی رو بهش Connect کردم که:
1- در MPage یک سری متاتگها هست + CSS + JavaScript +
2- وقتی در یک webForm متصل به Mpage کد :

```
Head.InnerHtml = SEO.GenerateMetaTag ....
```

رو قرار می دهم تمام تگ های Head حتی MasterPage هم پاک می شه
اینم می دونم این کلاس این کارو می کنه
راه حل چیه که تگ ها پاک نشه ؟ مخصوصا آدرس های CSS و JavaScript ؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۵۱ ۱۳۹۲/۱۰/۱۴

کمی بالاتر توضیح دادند : استفاده از یک Literal برای مثال.

نویسنده: مهرداد پاک دل
تاریخ: ۱۶:۱۶ ۱۳۹۲/۱۱/۰۲

من این توابع را خواندم که در مورد seo نوشته بودید ولی متاسفانه نحوه استفاده از آن هارا نفهمیدم یعنی کجا تعریف میشون و از کجا پارامترهای خود را دریافت می کنند. لطف کنید اگر امکان داره یه پروژه کوچک همه اینها استفاده شده باشه اگر هست بگذارید .
نکته دیگه شما عنوان هر پست را داخل url ها نمایش میدهید منظور اینکه جزوی از url سایت است این کار را چگونه انجام می دهید

mysite.info/post/1

dotnettips.info/post/1200/seo

نویسنده: محسن خان
تاریخ: ۱۶:۳۸ ۱۳۹۲/۱۱/۰۲

- پروژه Iris هست. به اون دقت کن در فایل های View .

- مثلاً متده Head.InnerHtml = SEO.GenerateMetaTag عنوان شده در این مطلب باید در Page_Load یک وب فرم فراخوانی شود. مطلب رو دارید. عنوان و متن و سایر مشخصات اون رو از دیتابیس دریافت کنید و بعد فقط یک جایگذاری است در متده Id شده.

- به اینکار Url rewrite و Routing می گن. بحث در MVC و وب فرمها کمی با هم فرق می کنه.

نویسنده: امین قادری
تاریخ: ۳:۳۵ ۱۳۹۳/۰۵/۰۹

لطفاً مثالی را در مورد نحوه استفاده از " متده برای ایجاد Slug " در عمل ارسال نمائید. با تشکر

نویسنده: موسوی
تاریخ: ۱۸:۹ ۱۳۹۳/۰۹/۲۲

سلام منم کدها رو تویه فایل cs به نام سئو دخیره کردم اما اینجا وقتی ران میکنم ارور میده

```
public enum CacheControlType
{
    [Description("public")]
    _public,
```

```
[Description("private")]
private,
[Description("no-cache")]
_nocache,
[Description("no-store")]
_nostore
}
```

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۹/۲۲ ۱۹:۵

« ارور می ده » برای رفع مشکل کافی نیست. چه خطایی می دهد دقیقا؟

نویسنده: موسوی
تاریخ: ۱۳۹۳/۰۹/۲۳ ۸:۴۰

درسته حق با شمامست. این ارور هست: یه نگاهی بندازین فکر کنم با فرم وورکم مشکل داره؟

Attribute 'Description' is not valid on this declaration type. It is only valid on 'method' declarations.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۲۳ ۹:۲۹

- ابتدا باید اسمبلی استاندارد System.ComponentModel.DataAnnotations پروژه اضافه شود.
- سپس باید از فضای نام مرتبط استفاده کنید (و نه از فضاهای نام مشابه)

```
enum MyEnum {
    [System.ComponentModel.Description("Blah")]
    MyValue
}
```

نویسنده: موسوی
تاریخ: ۱۳۹۳/۰۹/۲۳ ۱۵:۵۸

بازم تشکر به مشکل همین بود و طبق دستور شما استفاده کردم و در head از يه لیترال استفاده کردم و خیلی خوب جواب داد. زنده باشین

نویسنده: محمدرضا امینی
تاریخ: ۱۳۹۳/۱۲/۰۳ ۱۱:۵۹

سلام.

هنگام جستجو در گوگل گاهی اوقات سایتی باز میشه که خود اون سایت هم کلمه کلیدی شما رو جستجو کرده و نتایج رو به شما نشون میده.

مثلا یک فروشگاه خوب هنگامی که در گوگل قیمت رم لپ تاپ رو جستجو کنید چنین صفحه ای داره.
به نظر شما دوستان چطور میشه چنین صفحه ای در سایت خود داشته باشیم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۰۳ ۱۵:۷

- ابتدا یک پروفایل Google analytics ایجاد کنید و سپس اسکریپت آن را به سایت خودتان اضافه کنید.
- چند روز بعد که به آمار آن مراجعه کنید، میتوانید لیست جستجوهای گوگل منتهی به سایت خودتان را به همراه واژه های کلیدی مرتبط، گزارش گیری کنید.

Location		
Devices	<input type="checkbox"/>	1. (not provided)
Users Flow	<input type="checkbox"/>	2. dotnettips.info
Acquisition	<input type="checkbox"/>	3. بوت استریپ less
Overview	<input type="checkbox"/>	4. openid جیست
All Traffic	<input type="checkbox"/>	5. http://www.dotnettips.info/projectissues/
Channels	<input type="checkbox"/>	6. disablemvcresponseheader dotnettips
Source/Medium	<input type="checkbox"/>	7. web font cdn فارسی
Referrals	<input type="checkbox"/>	8. dotnettips

- بر اساس این واژه های کلیدی، برای محصولات خودتان برچسب درست کنید یا آنها را گروه بندی کنید. گوگل بر این اساس در دفعات آتی، نتایج جستجوی دقیق تری را به کاربران ارائه می دهد.

نویسنده: محسن نجف زاده
تاریخ: ۱۳۹۳/۱۲/۰۳ ۱۵:۲۵

- پیرو تکمیل صحبت آقای نصیری،
احتمالاً جستجو در وب سایت مورد نظر در دو صفحه مجزا طراحی شده:
 1. جستجوی های طبقه بندی شده بر اساس اطلاعات استخراجی شان از ابزارها ارزیابی وب سایت ها مانند [Google analytics](#) (راهنمای کامنت قبل)
 2. جستجوی لحظه ای در محتوای سایت این نکته در متاتگ های ایجاد شده نیز قابل مشاهده است:



www.

.com/Search/Category-Laptop-RAM/#/Category-Laptop-RAM/

```
<title> جستجو در گروه رم لپ تاپ </title>
<meta content="جستجو در گروه رم لپ تاپ" name="description">
<meta content="جستجو در گروه رم لپ تاپ" name="keywords">
```



www.

.com/Search/#/Keyword-/رم لپ تاپ-

```
<title> جستجو </title>
```

تعداد نتایج یافت شده متفاوت این دو صفحه (با یک عنوان جستجو) نشان از تایید مطلب دارد.

نویسنده: محمد رضا امینی
تاریخ: ۱۳۹۳/۱۲/۱۹ ۱۶:۵۱

برفرض مثال ما می خواهیم زمانی که از گوگل شخصی به سایت ما هدایت می شود با کلمه کلیدی ای که وارد سایت شده به طور خودکار در سایت جستجو کنیم.
کلا چطور میشه متوجه شد که کاربری از گوگل به سایت ما وارد شده و با چه جستجویی؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۱۹ ۱۹:۱۴

Request.UrlReferrer را آنالیز کنید. مقدار آن مشخص می کند که مثلا شخص از گوگل است یا سایت های جستجوی دیگر. بعد بر این اساس، کوئری استرینگ و آن را یافته و سپس شخص را به صفحه جستجو هدایت کنید:

```
// for Google
var urlReferrer = Request.UrlReferrer.ToString();
var query = HttpUtility.ParseQueryString(urlReferrer);
var searchQuery = query["q"];
```

[یک نمونه کاملتر](#)

نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۳/۱۲/۲۰ ۲۰:۲۹

یک سوال داشتم
اینکه این گونه سایتها از این روش برای افزایش بازدید استفاده می کنند یک نوع کلاه برداری در SEO به شمار نمیاد؟
چون که عموما این سایتها هیچ محتوای مرتبط ندارن و به خاطر افزایش بازدید اینکارو می کنن و باعث به هم ریختن نتایج جست و جو میشن تا جایی که گاهی اوقات می بینی چند صفحه اول تمام لینکها از این قبیل سایت هاست و گوگل هم بارها هشدار داده سایتها که کلاه برداری می کنند و محتوا ندارن رو جریمه می کنه ولی به تا به حال یک بار هم این حرف عملی نشده و همچنان همان سایتها هم به فعالیت ادامه میدن و باعث به هم ریختگی میشن

نویسنده: محسن خان
تاریخ: ۹:۵۶ ۱۳۹۳/۱۲/۲۱

هر کسی از ظن خود شد یار من ...

در مورد **glimpse** پیشتر مطالبی در سایت منتشر شده است:

آشنایی و بررسی ابزار Glimpse

بعد از آپلود سایت ما می‌توانیم دسترسی به تنظیمات خاص **glimpse** را تنها به کاربران عضو محدود کنیم:

```
<location path="Glimpse.axd" >
  <system.web>
    <authorization>
      <allow users="Administrator" />
      <deny users="*" />
    </authorization>
  </system.web>
</location>
```

یا می‌توانیم آنرا غیرفعال کنیم:

```
<glimpse defaultRuntimePolicy="Off" xdt:Transform="SetAttributes">
</glimpse>
```

همچنین می‌توانیم با پیاده سازی اینترفیس **IRuntimePolicy** سیاست‌های مختلف نمایش تب‌های **glimpse** را تعیین کنیم:

```
using Glimpse.AspNet.Extensions;
using Glimpse.Core.Extensibility;

namespace Test
{
    public class GlimpseSecurityPolicy:IRuntimePolicy
    {
        public RuntimePolicy Execute(IRuntimePolicyContext policyContext)
        {
            // You can perform a check like the one below to control Glimpse's permissions within your
            // application.
            // More information about RuntimePolicies can be found at http://getglimpse.com/Help/Custom-Runtime-
            // Policy
            var httpContext = policyContext.GetHttpContext();
            if (!httpContext.User.IsInRole("Administrator"))
            {
                return RuntimePolicy.Off;
            }

            return RuntimePolicy.On;
        }

        public RuntimeEvent ExecuteOn
        {
            get { return RuntimeEvent.EndRequest; }
        }
    }
}
```

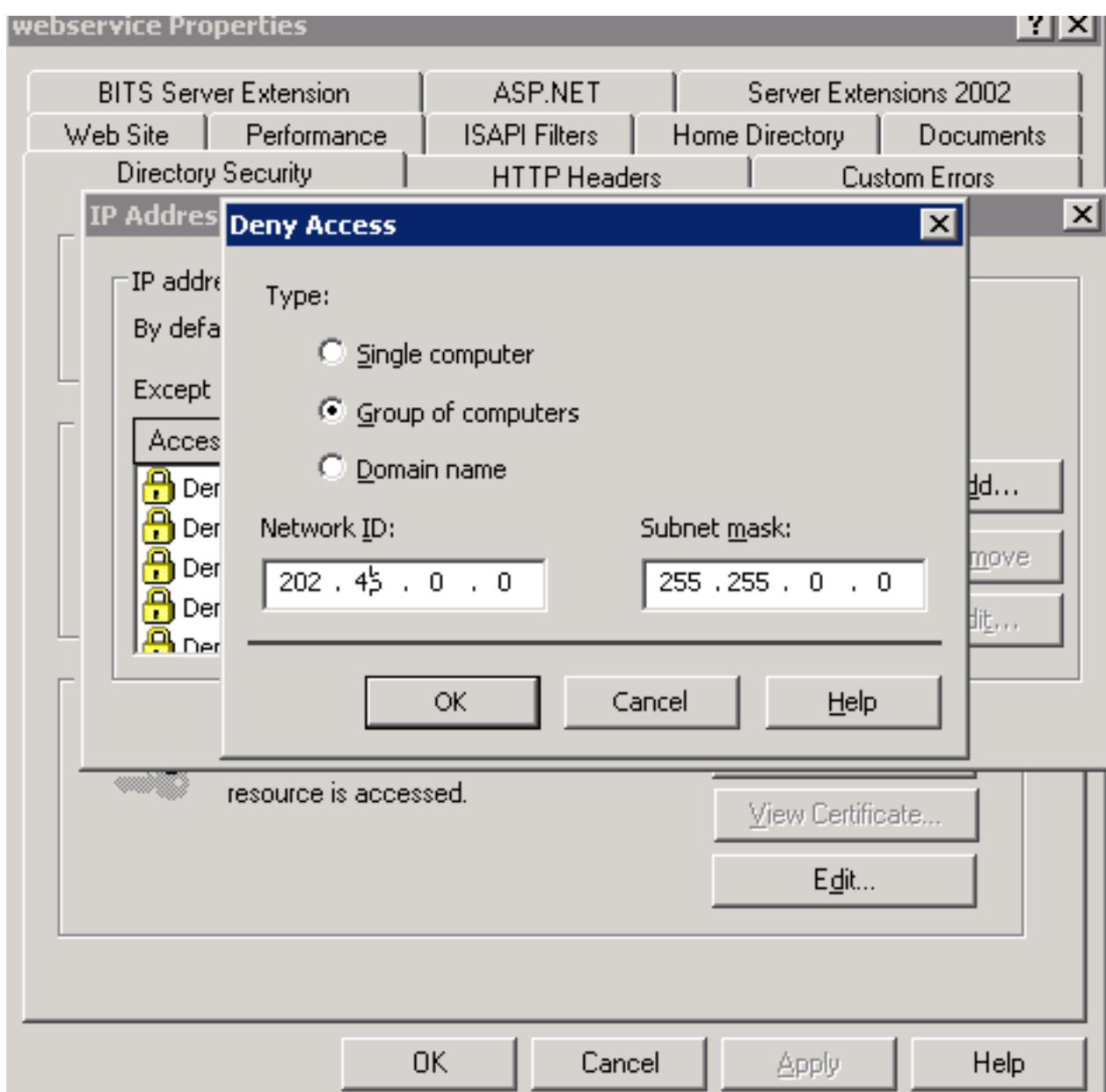
زمانیکه **glimpse** را از طریق **Nuget** نصب می‌کنید کلاس فوق به صورت اتوماتیک به پروژه اضافه می‌شود با این تفاوت که صورت کامنت شده است تنها کاری شما باید انجام بدید کدهای فوق را از حالت کامنت خارج کنید و **Role** مربوطه را جایگزین کنید.

نکته: کلاس فوق نیاز به **Registar** شدن ندارد و تشخیص آن توسط **Glimpse** به صورت خودکار انجام می‌شود.

یک سری از ربات‌ها مدام سایت‌ها را برای یافتن یک سری از اسکریپت‌های خاص اسکن می‌کنند. IP‌های آن‌ها نیز عموماً متعلق است به چین و هسایگان آن. مشکلی که با این ربات‌ها وجود دارد این است که از یک IP خاص نشات نمی‌گیرند و به نظر صدھا سرور آلوده را جهت مقاصد خود مورد استفاده قرار می‌دهند. به همین جهت نیاز است بتوان یک بازه‌ی IP را در IIS بست.

IIS یک بازه‌ی IP در 6

در IIS 6 باید به خواص وب سایت و برگه‌ی آن مراجعه کرد. سپس در این قسمت، در حین افزودن IP مد نظر، باید گزینه‌ی [Group of computers](#) را انتخاب نمود.



در اینجا برای مثال برای بستن IP‌هایی که با 194 شروع می‌شوند، باید 194.0.0.0 را وارد کرد و در این حالت subnet mask را نیز باید 255.0.0.0 انتخاب کرد. با این subnet mask خاص، اعلام می‌کنیم که فقط اولین قسمت IP وارد شده برای ما اهمیت دارد و سه قسمت بعدی خیر. به این ترتیب تمام IP‌های شروع شده با 194 با هر سه جزء دیگر دلخواهی، بسته خواهند شد.

بستن یک بازه‌ی IP در IIS‌های 7 به بعد

در IIS‌های 7 به بعد نیاز است مراحل زیر را طی کرده و نقش « [IP and Domain Restrictions](#) » را نصب کنید.

Administrative Tools -> Server Manager -> expand Roles
-> Web Server (IIS) -> Role Services -> Add Role Services -> select IP and Domain Restrictions

پس از آن با استفاده از تنظیمات ذیل در فایل web.config می‌توان یک IP و یا بازه‌ای از IP‌ها را بست:

```
<system.webServer>
  <security>
    <ipSecurity>
      <add ipAddress="192.168.100.1" />
      <add ipAddress="169.254.0.0" subnetMask="255.255.0.0" />
    </ipSecurity>
  </security>
</system.webServer>
```

البته علاوه بر نصب نقش یاد شده، باید Feature Delegation نیز جهت استفاده از آن فعال گردد:

IIS 7 -> root server -> Feature Delegation -> IP and Domain Restrictions
-> Change the delegation to Read/Write

نظرات خوانندگان

نویسنده: آرمان فرقانی
تاریخ: ۱۳۹۲/۰۹/۱۷ ۱۲:۴۶

ضمن تشکر بفرمایید چرا از طریق فایروال دسترسی IP‌های یاد شده را مسدود نمی‌کنید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۷ ۱۲:۴۷

این هم یک روش است؛ البته برای کسی که ادمین است. اگر ادمین نباشد و ادمین قبلاً قابلیت ذکر شده مخصوص 7 IIS را افزوده باشد، کاربران یک هاست اشتراکی هم می‌توانند راسا و بدون نیاز به ادمین، فقط با ویرایش کردن فایل web.config، اقدام کنند.

نویسنده: آرمان فرقانی
تاریخ: ۱۳۹۲/۰۹/۱۷ ۱۳:۲

بله. البته منابع سیستمی که اسکن‌های یاد شده مصرف می‌کنند، معمولاً مسدود نمودن آنها را در حیطه وظایف مدیر سرور قرار می‌دهد و البته در مورد 6 IIS دسترسی به وب سرور لازم است. و احتمالاً مسدود نمودن از طریق فایروال از نظر سربار و مصرف منابع ارجح است. از طرفی باید در نظر داشت که همان سرورهای چینی علاقه مند به اسکن سایتها علاقه مند به حملات DDOS و ... هم هستند که الزاماً از مسیر IIS نمی‌گذرد.

در هر صورت ممنون از بیان مطلب فوق که به هر حال دانستن آن بهتر از ندانستن است. سوال کردم چون گفتم شاید دلیل خاصی دارد که این روش را بیان فرمودید.

یکی از مسائلی که همیشه برای وب سایت، از نظر موتورهای جستجو بسیار مهم است، مقوله SEO بودن آدرس هاست. خوبشخانه در ASP.NET MVC این مقوله به خوبی پوشش داده شده است. برای مثال این قابلیت در ASP.NET Webforms توسعه بسته نرم افزاری به نام [ASP.NET FriendlyUrls](#) توسعه دهنده ASP.NET ارائه شده است. در این زمینه قبل آقای [کاویانی](#) در همین سایت در صفحه [مروی بر قابلیت جدید ASP.NET FriendlyUrls](#) به این مقوله پرداخته اند و با مثال، نحوه استفاده از این بسته نرم افزاری را آموزش داده اند.

یکی از مسائلی که در ASP.NET Webforms با آن مواجه بودیم انتقال دائمی آدرس های صفحات یا همان Permanent Redirect بود. در واقع زمانیکه شما در حالت عادی از بسته نرم افزاری مذکور استفاده می کنید، این افزونه باعث SEO شدن آدرس های صفحات شما می گردد. با در نظر گرفتن اینکه شما مطلب [مروی بر قابلیت جدید ASP.NET FriendlyUrls](#) را مطالعه کرده اید، به فرض مثال کد زیر:

```
public static class RouteConfig
{
    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.EnableFriendlyUrls();
        routes.MapPageRoute("Sitemap", "sitemap.xml", "~/sitemap.aspx");
    }
}
```

و رجیستر کردن آن در RouteTable سایت به صورت زیر در فایل Global

```
protected void Application_Start(object sender, EventArgs e)
{
    RouteConfig.RegisterRoutes(RouteTable.Routes);
}
```

باعث می شود که با زدن آدرس <http://www.dotnettips.info/sitemap.aspx> آدرس <http://www.dotnettips.info/sitemap.xml> لود شود؛ اما در نوار آدرس، همان آدرس sitemap.xml باقی خواهد ماند. در این روش در صورتیکه کاربر آدرس <http://www.dotnettips.info/sitemap.aspx> را در مرورگر خود تایپ کند باز هم سایت کار می کند (آدرس مورد نظر در دسترس است) و فاجعه زمانی شروع می شود که موتور جستجو، با محتوی تکراری در سایت در دو آدرس مختلف مواجه شود (پست آقای نصیری را در [بهبود SEO در ASP.NET MVC](#) بررسی کنید). یا زمانیکه شما صفحات قدیمی دارید و می خواهید آنها را به آدرس های جدید منتقل کنید. برای رفع این مشکل طوری که هر زمان آدرس قدیمی هم وارد سیستم مسیریابی سایت شود به صورت اتوماتیک به آدرس جدید به صورت دائم منتقل شود، می توانید کد بالا را به صورت زیر اصلاح نمایید:

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.EnableFriendlyUrls(new FriendlyUrlSettings()
    {
        AutoRedirectMode = RedirectMode.Permanent,
        ResolverCachingMode = ResolverCachingMode.Dynamic
    });
    routes.MapPageRoute("Sitemap", "sitemap.xml", "~/sitemap.aspx");
}
```

با این تنظیمات، سیستم مسیریابی ملزم به انتقال دائمی آدرس، به آدرس جدید خواهد شد. یعنی شما حتی اگر آدرس <http://www.dotnettips.info/sitemap.aspx> را وارد نمایید مسیریاب سایت برای آدرس موردنظر، وضعیت Moved 301 را مقدار دهی کرده و فقط آدرس <http://www.dotnettips.info/sitemap.xml> به عنوان آدرس اصلی در نظر گرفته خواهد شد.

نظرات خوانندگان

نویسنده: طاهر
تاریخ: ۱۴:۱۸ ۱۳۹۲/۰۹/۱۲

سلام

چه تفاوتی با [ASP.NET FriendlyUrls](#) که در دات نت 4 به وب فرم اضافه شده دارد؟

نویسنده: صابر فتح الهی
تاریخ: ۱۵:۳۹ ۱۳۹۲/۰۹/۱۲

از همون سیستم Routing دات نت استفاده می‌کنند با این تفاوت که کار باهاش راحت‌تر،

نویسنده: سالار
تاریخ: ۲۲:۴۱ ۱۳۹۳/۰۵/۲۹

سلام؛ ممنون به خاطر مطلب مفیدی که قرار دادید. من این تنظیمات رو اعمال کردم ولی بازم با درخواست هر دو آدرس ، صفحه بارگذاری میشه. در واقع باید هر دو آدرس رو اگر تایپ کردیم مشکلی پیش نیاد درسته؟ با 4.5 webforms کار میکنم. ممنون میشم بگید مشکل از چیه

نویسنده: صابر فتح الهی
تاریخ: ۱۴:۲۱ ۱۳۹۳/۰۵/۳۱

لطفا نمونه کد خودتون بذارید بررسی کنم

نویسنده: محسن خان
تاریخ: ۱۸:۱۱ ۱۳۹۳/۰۵/۳۱

در مثال پیوست شده زمانیکه sitemap.xml معمولی فراخوانی میشه به sitemap.xml هدایت نمیشه (وضعیت Moved 301 در مرورگر قابل مشاهده نیست) : [FriendlyUrlsTests.zip](#) Permanently

نویسنده: غلامرضا
تاریخ: ۲۰:۳۷ ۱۳۹۳/۰۵/۳۱

ممنون به خاطر پاسختون . پس دقیقا به این صورت هستش که هر دو آدرس رو اگر درخواست کنیم صفحه مورد نظر نمایش داده میشه؟ ولی مسیر یاب گوگل این رو مسیر تکراری به حساب نمیاره؟ خطأ رو نمایش نمیده پس؟

یکی از مواردی را که به کرات در ارجاع دهنده‌های سایت مشاهده می‌کنم، درج صفحات سایت به صورت `iframe` داخل یک سری سایت‌های تبلیغاتی بسیار سطحی است. سؤال: چگونه می‌توان جلوی این حرکت نامطلوب را گرفت؟ برای این کار نیاز است اسکریپت ذیل را به ابتدای اسکریپت‌های سایت خود اضافه کنید:

```
function defrm() {
    document.write = '';
    window.top.location = window.self.location;
    setTimeout(function () {
        document.body.innerHTML = '';
    }, 0);
    window.self.onload = function (evt) {
        document.body.innerHTML = '';
    };
}
if (window.top !== window.self) {
    try {
        if (window.top.location.host)
            { /* will throw */ }
        else {
            defrm(); /* chrome */
        }
    } catch (ex) {
        defrm(); /* everyone else */
    }
}
```

اگر شیء `window.top` با شیء `window.self` یکی نبود (یعنی `window` رویی نبود)، بنابراین صفحه‌ی جاری، داخل `iframe` قرار گرفته است. همچنین برای اطمینان بیشتر می‌توان `window.top.location.host` را نیز فراخوانی کرد. مرورگرهای جدید امکان دسترسی به دومین والد را از طریق یک `iframe` نمی‌دهند و همینجا یک استثناء صادر خواهد شد. در مرحله بعد، با فراخوانی `window.top.location = window.self.location` و سایر کدهایی که مشاهده می‌کنید، کل صفحه را به صورت خودکار به سایت خودمان هدایت خواهیم کرد.

این مورد را نیز می‌توان به مجموعه [best practices](#) اجباری تهیه یک سایت اضافه کرد.

نظرات خوانندگان

نویسنده: حامد سبزیان
تاریخ: ۹:۴ ۱۳۹۲/۰۹/۲۴

استفاده از `X-FRAME-OPTIONS` در هدر درخواست با مقدار `SAMEORIGIN` با `DENY` کافیت نمیکنه؟

نویسنده: وحید نصیری
تاریخ: ۹:۲۰ ۱۳۹۲/۰۹/۲۴

خوبه : فقط `redirect` بحث شده در مطلب را نداره

نویسنده: امید قربانی
تاریخ: ۹:۱۹ ۱۳۹۲/۰۹/۲۷

با سلام؛ اگر بخواهیم این مورد را که اشاره کردید رو دور بزنیم چطور باید عمل کنیم منظورم اینه که اگر سایتی همچین قابلیتی رو اضافه کرده بود چطور از بین بپریمش ؟

نویسنده: وحید نصیری
تاریخ: ۹:۲۶ ۱۳۹۲/۰۹/۲۷

حداقل با استفاده از جاوا اسکریپت، به دلایل امنیتی، امکان دسترسی به محتویات یک `iframe` بارگذاری شده از سایت دیگری را ندارید؛ مگر اینکه سمت سرور، محتویات آن صفحه را دریافت و در ادامه این صفحه‌ی محلی شده را نشان دهید.

نویسنده: behrouz
تاریخ: ۲۲:۱۰ ۱۳۹۲/۱۲/۲۸

سلام

آیا سمت سرور راهی هست که بتوان تشخیص داد صفحاتی که درخواست داده می‌شوند از درون یک `Iframe` فرخانی می‌شوند یا خیر(به طور خاص در `(ASP.NET MVC)`)

نویسنده: وحید نصیری
تاریخ: ۲۲:۵۹ ۱۳۹۲/۱۲/۲۸

خیر. راه حل تشخیص سمت سرور، ندارد. هیچ هدر خاصی برای مشخص سازی درخواست اجرای یک صفحه از درون یک `IFrame` طراحی نشده است. حداقل می‌توان `Request.UrlReferrer` را بررسی کرد (ولی اطمینانی به آن نیست. چون عوامل زیادی در تغییر آن دخیل هستند؛ از مرورگر تا فایروال و غیره)

مرورگرهای جدید تحت زیر مجموعه‌ای به نام Content Security Policy، قابلیت‌های توکاری را اضافه کرده‌اند تا حملاتی مانند XSS را حتی در برنامه‌ی وبی که برای این نوع حملات تمھیداتی را درنظر نگرفته است، خشی کنند. این قابلیت‌ها به صورت پیش فرض فعال نبوده و نیاز است برنامه نویس صراحتاً درخواست فعال شدن آن‌ها را از طریق افزودن تعدادی هدر مشخص به Response، ارائه دهد. در ادامه این هدرها را بررسی خواهیم کرد.

غیرفعال کردن اجرای اسکریپت‌های inline

عمده‌ی حملات XSS زمانی قابلیت اجرا پیدا می‌کنند که مهاجم بتواند به طریقی (ورودی‌های اعتبارسنجی نشده)، اسکریپتی را به درون صفحه‌ی جاری تزریق کند. بنابراین اگر ما به مرورگر اعلام کنیم که دیگر اسکریپت‌های inline را پردازش نکن، سایت را به حد زیادی در مقابل حملات XSS مقاوم کرده‌ایم. این قابلیت به صورت پیش فرض خاموش است؛ چون به طور قطع فعال سازی آن بسیاری از سایت‌هایی را که عادت کرده‌اند اسکریپت‌های خود را داخل صفحات وب مدفون کنند، از کار می‌اندازد. این نوع سایت‌ها باید به روز شده و اسکریپت‌ها را از طریق فایل‌های خارجی `js`، به سایت و صفحات خود الحاق کنند.

برای فعال سازی این قابلیت، فقط کافی است هدرهای زیر به Response اضافه شوند:

```
Content-Security-Policy: script-src 'self'  
X-WebKit-CSP: script-src 'self'  
X-Content-Security-Policy: script-src 'self'
```

سطر اول به زودی تبدیل به یک استاندارد W3 خواهد شد؛ اما فعلًا فقط توسط کروم 25 به بعد پشتیبانی می‌شود. سطر دوم توسط مرورگرهایی که از موتور WebKit استفاده می‌کنند، پشتیبانی می‌شود و سطر سوم مخصوص فایرفاکس است و IE 10 به بعد. بعد از فعال شدن این قابلیت، فقط اسکریپت‌هایی که از طریق دومین شما به صفحه الحاق شده‌اند، قابلیت اجرا را خواهند یافت و کلیه اسکریپت‌های مدفون شده داخل صفحات، دیگر اجرا نخواهد شد. در این حالت اگر از CDN برای الحاق اسکریپتی استفاده می‌کنید، مثلاً مانند الحاق `jQuery` به صفحه، نیاز است مسیر آن را صراحتاً در این هدر ذکر کنید:

```
Content-Security-Policy: script-src 'self' https://youcdn.com  
X-WebKit-CSP: script-src 'self' https://youcdn.com  
X-Content-Security-Policy: script-src 'self' https://youcdn.com
```

علاوه بر آن حتی می‌شود پردازش تمام منابع مورد استفاده را نیز مانند تصاویر، شیوه‌نامه‌ها، فایل‌های فلش و غیره، به دو مین جاری محدود کرد:

```
Content-Security-Policy: default-src 'self' https://youcdn.com  
X-WebKit-CSP: default-src 'self' https://youcdn.com  
X-Content-Security-Policy: default-src 'self' https://youcdn.com
```

بدیهی است پس از آشنایی با این مورد، احتمالاً در پروژه‌های جدید خود از آن استفاده کنید (چون inline script‌های فعلی شما را کاملاً از کار می‌اندازد).

نحوه اضافه کردن هدرهای Content Security Policy به برنامه‌های ASP.NET

روشی که با هر دو برنامه‌های وب فرم و MVC کار می‌کند، تهیه یک HTTP module است؛ به شرح ذیل:

```
using System;  
using System.Web;  
  
namespace AntiXssHeaders  
{
```

```

public class SecurityHeadersConstants
{
    public static readonly string XXssProtectionHeader = "X-XSS-Protection";
    public static readonly string XFrameOptionsHeader = "X-Frame-Options";
    public static readonly string XWebKitCspHeader = "X-WebKit-CSP";
    public static readonly string XContentSecurityPolicyHeader = "X-Content-Security-Policy";
    public static readonly string ContentSecurityPolicyHeader = "Content-Security-Policy";
    public static readonly string XContentTypeOptionsHeader = "X-Content-Type-Options";
}

public class ContentSecurityPolicyModule : IHttpModule
{
    public void Dispose()
    { }

    public void Init(HttpApplication app)
    {
        app.BeginRequest += AppBeginRequest;
    }

    void AppBeginRequest(object sender, EventArgs e)
    {
        var app = (HttpApplication)sender;
        var response = app.Context.Response;
        setHeaders(response);
    }

    private static void setHeaders(HttpResponse response)
    {
        response.Headers.Set(SecurityHeadersConstants.XFrameOptionsHeader, "SameOrigin");

        // For IE 8+
        response.Headers.Set(SecurityHeadersConstants.XXssProtectionHeader, "1; mode=block");
        response.Headers.Set(SecurityHeadersConstants.XContentTypeOptionsHeader, "nosniff");

        //todo: Add /Home/Report --> public JsonResult Report() { return Json(true); }

        const string cspValue = "default-src 'self';";
        // For Chrome 16+
        response.Headers.Set(SecurityHeadersConstants.XWebKitCspHeader, cspValue);

        // For Firefox 4+
        response.Headers.Set(SecurityHeadersConstants.XContentSecurityPolicyHeader, cspValue);
        response.Headers.Set(SecurityHeadersConstants.ContentSecurityPolicyHeader, cspValue);
    }
}
}

```

و یا در برنامه‌های ASP.NET MVC می‌توان یک فیلتر جدید را تعریف کرد و سپس آن را به صورت عمومی معرفی نمود:

```

//// RegisterGlobalFilters -> filters.Add(new ContentSecurityPolicyFilterAttribute());
public class ContentSecurityPolicyFilterAttribute : ActionFilterAttribute
{
    public override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        var response = filterContext.HttpContext.Response;
        response.AddHeader("Content-Security-Policy", "script-src 'self'");
        // the rest ...
        base.OnActionExecuting(filterContext);
    }
}

```

در مارژول تهیه شده چند مورد دیگر را نیز مشاهده می‌کنید:

الف) XSS-Protection مربوط است به IE 8 به بعد

ب) تنظیم هدر X-Frame-Options به SameOrigin سبب می‌شود تا صفحات سایت شما دیگر توسط Iframe‌ها در سایت‌های دیگر قابل نمایش نباشد و فقط در سایت جاری بتوان صفحه‌ای را از همان دومین در صورت نیاز توسط Iframe‌ها نمایش داد.

ج) تنظیم mime-type X-Content-Type-Options به nosniff سبب می‌شود تا IE سعی نکند با اجرای یک محتوا سعی در تشخیص آن کند و به این ترتیب امنیت دسترسی و مشاهده اشیاء قرار گرفته در صفحه (و یا تزریق شده توسط مهاجمین) به شدت بالا خواهد رفت.

برای مطالعه بیشتر

[Security through HTTP response headers](#)

پروژه‌ی کاملی مخصوص افزودن هدرهای یاد شده

[/https://nwebsec.codeplex.com](https://nwebsec.codeplex.com)

یک نکته تكميلي

توصيه شده است تا ديگر از روال رويدادگردن [PreSendRequestHeaders](#) برای ارسال هدرها استفاده نکنيد؛ چون با پردازش‌های غيرهمzman تداخل ايجاد می‌کند.

نظرات خوانندگان

نویسنده: محمد رعیت پیشه
تاریخ: ۹:۲۳ ۱۳۹۲/۰۹/۳۰

آیا اگر اجرای Inline غیر فعال شود و پردازش تمام منابع مورد استفاده به دامین جاری محدود شود مهاجم نمی‌تواند یک ارجاع به اسکریپت مخرب خود به صفحه اضافه کند؟

نویسنده: وحید نصیری
تاریخ: ۹:۳۷ ۱۳۹۲/۰۹/۳۰

اجرا نمی‌شود؛ مگر اینکه به هدر ارسالی مانند [CDN](#) ذکر شده در متن، آدرس آن اضافه و صراحتاً مجوز استفاده از دومینی دیگر توسط برنامه صادر شود.

نویسنده: ناصر طاهری
تاریخ: ۱۲:۲۲ ۱۳۹۲/۰۹/۳۰

یعنی کلیه اسکریپتها حتی اسکریپت هایی مانند :

```
<script type="text/javascript">
    $(document).ready(function () {
        PopupForm.ShowForm({
            renderFormUrl : "/postreply/renderreplyform",
            ....
        });
    });
</script>
```

اجازه‌ی اجرا ندارند؟ و باید در یک فایل جدا به صفحه تزریق شوند؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۱۸ ۱۳۹۲/۰۹/۳۰

بله. این مثال را در کروم اجرا کنید:

[AntiXssHeaders.zip](#)

در صفحه اول آن

```
<script type="text/javascript">
    alert('test');
</script>
```

درج شده

نویسنده: ناصر طاهری
تاریخ: ۱۳:۳۷ ۱۳۹۲/۰۹/۳۰

ممnon از مثالتون. یک پروژه فروشگاه دارم که داخل هر صفحه‌ی آن پر است از اسکریپت‌های به قول شما مذفون شده. حتی فکر پیدا کردن و منتقل کردن هر کدام از اونها به داخل یک فایل خارجی وحشت آور است.

نویسنده: رسول آذری
تاریخ: ۱۳:۴۷ ۱۳۹۲/۱۰/۰۴

با سلام . تشکر

من با مرورگر ie9 تست کردم . ولی کد اجرا شد. مگه نگفتنی واسه ie8 به بعد؟
و اینکه بعد از استفاده از این روش چطور میتونم از فایل‌های js و jquery خودم توی برنامه استفاده کنم؟

نویسنده: **وحید نصیری**
تاریخ: ۱۴:۱۸ ۱۳۹۲/۱۰/۰۴

- رفتار IE در مورد [X-XSS-Protection](#) متفاوت است. یعنی باید واقعاً تشخیص بدهد که اسکریپت در حال اجرا یک حمله محسوب می‌شود. [یک مثال دریافت کوکی برای امتحان](#)
- مانند قبل. در متن ذکر شده: « سایتها باید به روز شده و اسکریپتها را از طریق فایل‌های خارجی js ، به سایت و صفحات خود الحق کنند »

نویسنده: **رسول آذری**
تاریخ: ۱۴:۳۵ ۱۳۹۲/۱۰/۰۴

ممnon از پاسخ تون.
یه سری از کدها را میشه توی فایل خارجی قرار داد.
ولی کدهای جاوایی که گریدویو تویید میکنه، چی کار میشه کرد؟
با استفاده از این روش رویدادهایی مثل SelectedIndexChanged در dropdownlist ، با تشکر
با این افته. آیا راهی برای این مشکل وجود داره؟

نویسنده: **وحید نصیری**
تاریخ: ۱۵:۲ ۱۳۹۲/۱۰/۰۴

- به MVC کوچ کنید تا کنترل کاملی بر روی عناصر اضافه شونده به صفحه داشته باشید.
- با استفاده از jQuery به هر عنصری در صفحه می‌توان رویدادهای خاصی را [انتساب داد](#) یا حذف کرد. انجام اینکار از طریق یک فایل js الحق شده به صفحه نیز میسر است.

در نگارش‌های قبلی [ASP.NET Web forms](#) اگر نیاز به ارسال محتوای HTML ای و وجود داشت، می‌بایستی کل سیستم اعتبارسنجی حداقل یک صفحه را غیرفعال کرد. برای مثال:

```
<%@ Page Language="C#" ValidateRequest="false" %>
```

این نقیصه‌ی همه‌ی هیچ، در ASP.NET وجود ندارد و می‌توان به ازای یک خاصیت خاص، اعتبارسنجی پیش فرض را با اعمال ویژگی AllowHtml موقتاً غیرفعال کرد؛ اما مابقی فیلد‌ها و خاصیت‌های فرم همچنان تحت نظر سیستم اعتبارسنجی‌های ورودی قرار خواهد داشت و به این ترتیب امکان ورود اطلاعات خطرناک، خصوصاً از لحاظ مباحث XSS، حداقل در آن فیلد‌ها وجود نخواهد داشت.

در ASP.NET 4.5 مفهوم جدیدی به نام Deferred validation معرفی شده‌است تا رفتار Web forms را نیز در برابر ورودی‌های ارسال شده همانند MVC کند. به این معنا که اگر جهت دسترسی به مقدار کوئری استرینگ lastName همانند قبل عمل کنید:

```
Request["lastName"]
```

و کاربر ورودی خطرناکی را وارد کرده باشد، همچنان استثنای A potentially dangerous request.form value was detected صادر می‌شود.

اما اگر در ASP.NET 4.5، کوئری استرینگ را به نحو ذیل دریافت کنید:

```
Request.Unvalidated.QueryString["lastName"]
```

به صورت خودکار سیستم اعتبارسنجی غیرفعال شده و امکان دسترسی به محتوای اصلی کوئری استرینگ lastName را خواهد داشت. به این ترتیب دیگر نیازی نخواهد داشت تا اعتبارسنجی کل صفحه را برای دسترسی به یک مقدار خاص غیرفعال نماید. برای دسترسی به مقادیر اعتبارسنجی نشده فیلد‌های یک فرم ارسالی نیز می‌توان به صورت ذیل عمل کرد:

```
Request.Unvalidated.Form[txtData1.UniqueID]
```

کلاس [Request.Unvalidated](#) شامل خاصیت‌های Cookies, Files, Headers و امثال آن نیز می‌باشد. در اینجا اگر دقت کرده باشید از UniqueID بجای Id استفاده شده‌است؛ از این جهت که مجموعه Form، حاوی Id های واقعی دریافت شده از کاربر مانند ct100\$MainContent\$MessageText می‌باشد.

روش دوم، استفاده از خاصیت جدید [ValidateRequestMode](#) یک کنترل سمت سرور است و در اینجا نیز می‌توان صرفاً اعتبارسنجی یک کنترل را بجای کل صفحه، غیرفعال کرد:

```
<asp:TextBox ID="txtASPN" ValidateRequestMode="Disabled" runat="server" />
```

تنظیم خاصیت ذکر شده برای کنترل‌های سمت سرور ضروری است. از این جهت که در حین تشکیل ViewState از محتوای این کنترل‌ها نیز استفاده می‌شود. اینجا است که اگر ValidateRequestMode غیرفعال نشده باشد، باز همان خطای ورودی خطرناک را دریافت خواهید کرد.

البته برای فعل سازی اعتبارسنجی با تاخیر نیاز است اصلاح ذیل را نیز به web.config برنامه اعمال کنید ([مقدار پیش فرض آن](#) است):

```
<httpRuntime targetFramework="4.5" requestValidationMode="4.5" />
```

سیستم ASP.NET 2.0 بهمراه ASP.NET Membership در سال 2005 معرفی شد، و از آن زمان تا بحال تغییرات زیادی در چگونگی مدیریت احراز هویت و اختیارات کاربران توسط اپلیکیشن‌های وب بوجود آمده است. ASP.NET Identity نگاهی تازه است به آنچه که سیستم Membership هنگام تولید اپلیکیشن‌های مدرن برای وب، موبایل و تبلت باید باشد.

پیش زمینه: سیستم عضویت در ASP.NET

ASP.NET Membership

ASP.NET Membership طراحی شده بود تا نیازهای سیستم عضویت وب سایت‌ها را تامین کند، نیازهایی که در سال 2005 رایج بود و شامل مواردی مانند مدل احراز هویت فرم، و یک پایگاه داده SQL Server برای ذخیره اطلاعات کاربران و پروفایل هایشان می‌شد. امروزه گزینه‌های بسیار بیشتری برای ذخیره داده‌های وب اپلیکیشن‌ها وجود دارد، و اکثر توسعه دهنده‌اند از اطلاعات شبکه‌های اجتماعی نیز برای احراز هویت و تعیین سطوح دسترسی کاربرانشان استفاده کنند. محدودیت‌های طراحی سیستم ASP.NET Membership گذر از این تحول را دشوار می‌کند:

الگوی پایگاه داده آن برای SQL Server طراحی شده است، و قادر به تغییرش هم نیست. می‌توانید اطلاعات پروفایل را اضافه کنید، اما تمام داده‌ها در یک جدول دیگر ذخیره می‌شوند، که دسترسی به آنها نیز مشکل‌تر است، تنها راه دسترسی Profile Provider API خواهد بود.

سیستم تامین کننده (Provider System) امکان تغییر منبع داده‌ها را به شما می‌دهد، مثلاً می‌توانید از بانک‌های اطلاعاتی MySQL یا Oracle استفاده کنید. اما تمام سیستم بر اساس پیش فرض هایی طراحی شده است که تنها برای بانک‌های اطلاعاتی relational درست هستند. می‌توانید تامین کننده (Provider) ای بنویسید که داده‌های سیستم عضویت را در منبعی به غیر از دیتابیس‌های relational ذخیره می‌کند؛ مثلاً Windows Azure Storage Tables. اما در این صورت باید مقادیر زیادی کد بنویسید. مقادیر زیادی درست نیستند. باید بنویسید، برای متدهایی که به دیتابیس‌های NoSQL مربوط نیستند.

از آنجایی که سیستم ورود/خروج سایت بر اساس مدل Forms Authentication کار می‌کند، سیستم عضویت نمی‌تواند از OWIN استفاده کند. OWIN شامل کامپوننت‌هایی برای احراز هویت است که شامل سرویس‌های خارجی هم می‌شود (مانند Microsoft Accounts، Facebook، Google، Twitter Organizational). همچنین امکان ورود به سیستم توسط حساب‌های کاربری سازمانی (Accounts) نیز وجود دارد مانند Windows Azure Active Directory و Active Directory. این کتابخانه از CORS و OAuth 2.0، JWT و OWIN استفاده کرد. این کتابخانه از نیز پشتیبانی می‌کند.

ASP.NET Simple Membership

به عنوان یک سیستم عضویت، برای فریم ورک Web Pages توسعه داده شد. این سیستم با ASP.NET simple membership Visual Studio 2010 SP1 و WebMatrix انتشار یافت. هدف از توسعه این سیستم، آسان کردن پروسه افزودن سیستم عضویت به یک اپلیکیشن Web Pages بود. این سیستم پروسه کلی کار را آسان‌تر کرد، اما هنوز مشکلات ASP.NET Membership را نیز داشت. محدودیت‌های نیز وجود دارند:

ذخیره داده‌های سیستم عضویت در بانک‌های اطلاعاتی non-relational مشکل است.

نمی‌توانید از آن در کنار OWIN استفاده کنید.

با فراهم کننده‌های موجود ASP.NET Membership بخوبی کار نمی‌کند. توسعه پذیر هم نیست.

ASP.NET Universal Providers

برای ذخیره سازی اطلاعات سیستم عضویت در Windows Azure SQL Database توسعه پیدا کردند. با SQL Server Compact هم بخوبی کار می‌کنند. این تامین کننده‌ها بر اساس Entity Framework Code First ساخته شده بودند و بدین معنا بود که داده‌های سیستم عضویت را می‌توان در هر منبع داده‌ای که توسط EF پشتیبانی می‌شود ذخیره کرد. با انتشار این تامین کننده‌ها الگوی دیتابیس سیستم عضویت نیز بسیار سبک‌تر و بهتر شد. اما این سیستم بر پایه زیر ساخت ASP.NET

نوشته شده است، بنابراین محدودیت‌های پیشین مانند محدودیت‌های `SqlMembershipProvider` هنوز وجود دارند. به بیان دیگر، این سیستم‌ها همچنان برای بانک‌های اطلاعاتی `relational` طراحی شده اند، پس سفارشی سازی اطلاعات کاربران و پروفایل‌ها هنوز مشکل است. در آخر آنکه این تامین کننده‌ها هنوز از مدل احراز هویت فرم استفاده می‌کنند.

ASP.NET Identity همانطور که داستان سیستم عضویت ASP.NET طی سالیان تغییر و رشد کرده است، تیم ASP.NET نیز آموخته‌های زیادی از بازخوردهای مشتریان شان بدست آورده اند. این پیش فرض که کاربران شما توسط یک نام کاربری و کلمه عبور که در اپلیکیشن خودتان هم ثبت شده است به سایت وارد خواهند شد، دیگر معتبر نیست. دنیای وب اجتماعی شده است. کاربران از طریق وب سایتها و شبکه‌های اجتماعی متعددی با یکدیگر در تماس هستند، خیلی از وقت بصورت زنده! شبکه‌های مانند Facebook و Twitter. همانطور که توسعه نرم افزارهای تحت وب رشد کرده است، الگوها و مدل‌های پیاده سازی نیز تغییر و رشد کرده اند. امکان Unit Testing روی کد اپلیکیشن‌ها، یکی از مهم‌ترین دلوایپسی‌های توسعه دهنده‌گان شده است. در سال 2008 تیم ASP.NET فریم ورک جدیدی را بر اساس الگوی (MVC) Model-View-Controller اضافه کردند. هدف آن کمک به توسعه دهنده‌گان، برای تولید برنامه‌های ASP.NET با قابلیت Unit Testing بهتر بود. توسعه دهنده‌گانی که می‌خواستند کد اپلیکیشن‌های خود را Unit Test کنند، همین امکان را برای سیستم عضویت نیز می‌خواستند.

با در نظر گرفتن تغییراتی که در توسعه اپلیکیشن‌های وب بوجود آمده ASP.NET Identity با اهداف زیر متولد شد:

یک سیستم هویت واحد (One ASP.NET Identity system)

سیستم ASP.NET Identity می‌تواند در تمام فریم ورک‌های مشتق از ASP.NET استفاده شود. مانند Web API و SignalR و Pages، Web API

از این سیستم می‌توانید در تولید اپلیکیشن‌های وب، موبایل، استور (Store) و یا اپلیکیشن‌های ترکیبی استفاده کنید.

سادگی تزریق داده‌های پروفایل درباره کاربران

روی الگوی دیتابیس برای اطلاعات کاربران و پروفایل‌ها کنترل کامل دارید. مثلاً می‌توانید به سادگی یک فیلد، برای تاریخ تولد در نظر بگیرید که کاربران هنگام ثبت نام در سایت باید آن را وارد کنند.

کنترل ذخیره سازی/واکشی اطلاعات

بصورت پیش فرض ASP.NET Identity تمام اطلاعات کاربران را در یک دیتابیس ذخیره می‌کند. تمام مکانیزم‌های دسترسی به داده‌ها توسط EF Code First کار می‌کنند.

از آنجا که روی الگوی دیتابیس، کنترل کامل دارید، تغییر نام جداول و یا نوع داده فیلد‌های کلیدی و غیره ساده است. استفاده از مکانیزم‌های دیگر برای مدیریت داده‌های آن ساده است، مانند SharePoint، Windows Azure Storage Table و دیتابیس‌های NoSQL.

تست پذیری

ASP.NET Identity تست پذیری اپلیکیشن وب شما را بیشتر می‌کند. می‌توانید برای تمام قسمت‌هایی که از ASP.NET Identity استفاده می‌کنند تست بنویسید.

تامین کننده نقش (Role Provider)

تامین کننده‌ای وجود دارد که به شما امکان محدود کردن سطوح دسترسی بر اساس نقش را می‌دهد. بسادگی می‌توانید نقش‌های جدید مانند "Admin" بسازید و بخش‌های مختلف اپلیکیشن خود را محدود کنید.

Claims Based

ASP.NET Identity از امکان احراز هویت بر اساس Claims نیز پشتیبانی می‌کند. در این مدل، هویت کاربر بر اساس دسته‌ای از اختیارات او شناسایی می‌شود. با استفاده از این روش توسعه دهنده‌گان برای تعریف هویت کاربران، آزادی عمل بیشتری نسبت به مدل Roles دارند. مدل نقش‌ها تنها یک مقدار منطقی (bool) است؛ یا عضو یک نقش هستید یا خیر، در حالیکه با استفاده از روش Claims می‌توانید اطلاعات بسیار ریز و دقیقی از هویت کاربر در دست داشته باشید.

تامین کننده‌گان اجتماعی

به راحتی می‌توانید از تامین کنندگان دیگری مانند Microsoft, Facebook, Twitter, Google و غیره استفاده کنید و اطلاعات مربوط به کاربران را در اپلیکیشن خود ذخیره کنید.

Windows Azure Active Directory

برای اطلاعات بیشتر به [این لینک](#) مراجعه کنید.

یکپارچگی با OWIN

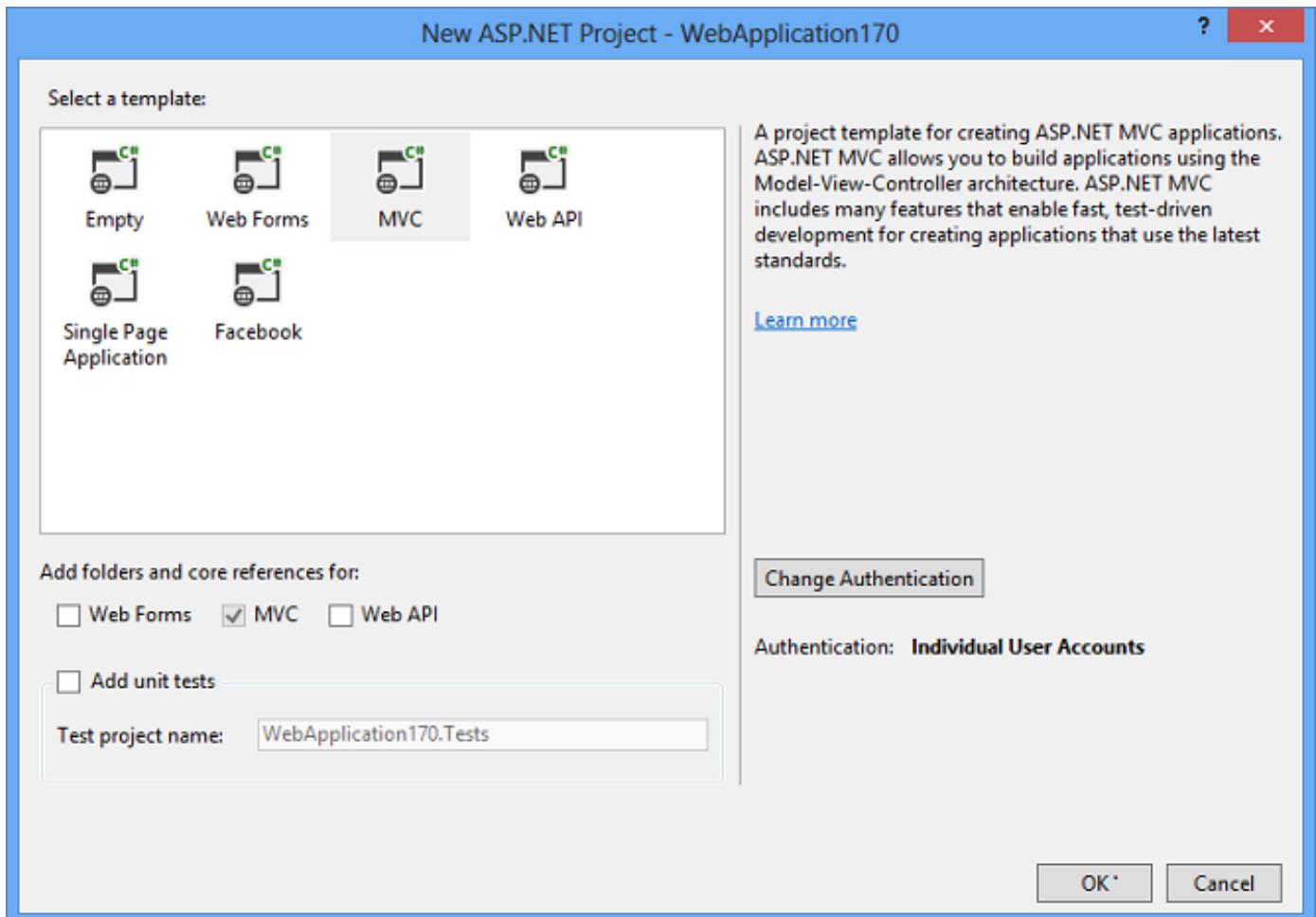
ASP.NET Identity بر اساس OWIN توسعه پیدا کرده است، بنابراین از هر میزبانی که از OWIN پشتیبانی می‌کند می‌توانید استفاده کنید. همچنین هیچ وابستگی ای به System.Web وجود ندارد. ASP.NET Identity یک فریم ورک کامل و مستقل برای OWIN است و می‌تواند در هر اپلیکیشنی که روی OWIN میزبانی شده استفاده شود. Forms از OWIN برای ورود/خروج کاربران در سایت استفاده می‌کند. این بدین معنا است که بجای استفاده از Authentication برای تولید یک کوکی، از OWIN CookieAuthentication استفاده می‌شود.

NuGet پکیج

ASP.NET Identity در قالب یک بسته NuGet توزیع می‌شود. این بسته در قالب پروژه‌های Web API و ASP.NET MVC، Web Forms با Visual Studio 2013 منتشر شدند گنجانده شده است. توزیع این فریم ورک در قالب یک بسته NuGet این امکان را به تیم ASP.NET می‌دهد تا امکانات جدیدی توسعه دهند، باگ‌ها را برطرف کنند و نتیجه را بصورت چاپک به توسعه دهنده‌گان عرضه کنند.

شروع کار با ASP.NET Identity

ASP.NET Identity در قالب پروژه‌های SPA و ASP.NET MVC، Web API و ASP.NET MVC، Web Forms که بهمراه Visual Studio 2013 منتشر شده اند استفاده می‌شود. در ادامه به اختصار خواهیم دید که چگونه ASP.NET Identity کار می‌کند. یک پروژه جدید ASP.NET MVC با تنظیمات Individual User Accounts بسازید.



پروژه ایجاد شده شامل سه بسته می‌شود که مربوط به ASP.NET Identity هستند:

این بسته شامل پیاده سازی Entity Framework با ASP.NET Identity می‌شود، که تمام داده‌های مربوطه را در یک دیتابیس SQL Server ذخیره می‌کند. این بسته محتوی تمام interface‌های ASP.NET Identity است. با استفاده از این بسته می‌توانید پیاده سازی دیگری از ASP.NET Identity بسازید که منبع داده متفاوتی را هدف قرار می‌دهد. مثلا Windows Azure Storage Table و دیتابیس‌های NoSQL.

این بسته امکان استفاده از احراز هویت OWIN را در اپلیکیشن‌های ASP.NET فراهم می‌کند. هنگام تولید کوکی‌ها از OWIN Cookie Authentication استفاده خواهد شد.

اپلیکیشن را اجرا کرده و روی لینک Register کلیک کنید تا یک حساب کاربری جدید ایجاد کنید.

The screenshot shows a web browser window with a blue header bar. The title bar reads "Register - My ASP.NET Ap". The address bar shows the URL "localhost:1234/Account/Register". The main content area is titled "Application name" and features a large "Register." heading. Below it, the text "Create a new account." is displayed. There are three input fields: "User name", "Password", and "Confirm password", each with a corresponding text input box. A "Register" button is located below the "Confirm password" field. At the bottom of the page, the copyright notice "© 2013 - My ASP.NET Application" is visible.

Register - My ASP.NET Ap

localhost:1234/Account/Register

Application name

Register.

Create a new account.

User name

Password

Confirm password

Register

© 2013 - My ASP.NET Application

هنگامیکه بر روی دکمه‌ی Register کلیک شود، کنترلر Account، اکشن متده‌ی Register را فراخوانی می‌کند تا حساب کاربری جدیدی با استفاده از API ASP.NET Identity ساخته شود.

```
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Register(RegisterViewModel model)
{
    if (ModelState.IsValid)
    {
        var user = new ApplicationUser() { UserName = model.UserName };
        var result = await UserManager.CreateAsync(user, model.Password);
        if (result.Succeeded)
        {
            await SignInAsync(user, isPersistent: false);
            return RedirectToAction("Index", "Home");
        }
        else
        {
            AddErrors(result);
        }
    }

    // If we got this far, something failed, redisplay form
    return View(model);
}
```

اگر حساب کاربری با موفقیت ایجاد شود، کاربر توسط فراخوانی متده‌ی SignInAsync به سایت وارد می‌شود.

```
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Register(RegisterViewModel model)
{
    if (ModelState.IsValid)
    {
        var user = new ApplicationUser() { UserName = model.UserName };
        var result = await UserManager.CreateAsync(user, model.Password);
        if (result.Succeeded)
        {
            await SignInAsync(user, isPersistent: false);
            return RedirectToAction("Index", "Home");
        }
        else
        {
            AddErrors(result);
        }
    }

    // If we got this far, something failed, redisplay form
    return View(model);
}
```

```
private async Task SignInAsync(ApplicationUser user, bool isPersistent)
{
    AuthenticationManager.SignOut(DefaultAuthenticationTypes.ExternalCookie);

    var identity = await UserManager.CreateIdentityAsync(
        user, DefaultAuthenticationTypes.ApplicationCookie);

    AuthenticationManager.SignIn(
        new AuthenticationProperties() {
            IsPersistent = isPersistent
        }, identity);
}
```

از آنجا که OWIN Cookie Authentication و ASP.NET Identity هستند، فریم ورک، انتظار آبجکتی از نوع Claims-based را خواهد داشت. این آبجکت تمامی اطلاعات لازم برای تشخیص هویت کاربر را در بر دارد. مثلاً اینکه کاربر

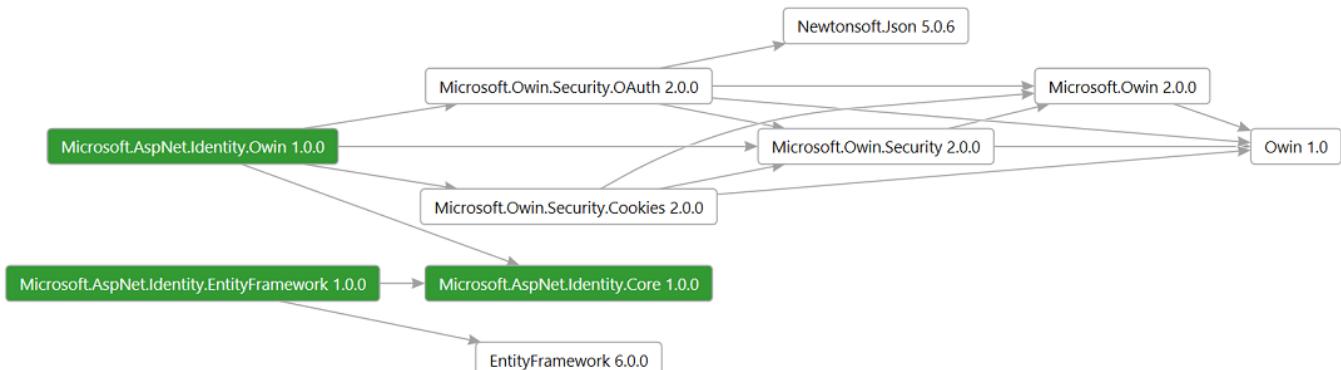
مورد نظر به چه نقش هایی تعلق دارد؟ و اطلاعاتی از این قبیل. در این مرحله میتوانید Claims های بیشتری را به کاربر بیافزایید. کلیک کردن روی لینک Log off در سایت، اکشن متده LogOff در کنترلر Account را اجرا میکند.

```
// POST: /Account/LogOff
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult LogOff()
{
    AuthenticationManager.SignOut();
    return RedirectToAction("Index", "Home");
}
```

همانطور که مشاهده میکنید برای ورود/خروج کاربران از AuthenticationManager استفاده میشود که متعلق به OWIN است. متده [FormsAuthentication.SignOut](#) همتای متده SignOut است.

کامپونت های ASP.NET Identity

تصویر زیر اجزای تشکیل دهنده ASP.NET Identity را نمایش میدهد. بسته هایی که با رنگ سبز نشان داده شده اند سیستم کلی ASP.NET Identity را میسازند. مابقی بسته ها وابستگی هایی هستند که برای استفاده از ASP.NET Identity در اپلیکیشن های ASP.NET لازم اند.



دو پکیج دیگر نیز وجود دارند که به آنها اشاره نشد:

این بسته امکان استفاده از مدل احراز هویت مبتنی بر کوکی (Cookie-based) را فراهم میکند. مدلی مانند سیستم [Microsoft.Security.Owin.Cookies](#).ASP.NET Forms Authentication را فراهم میکند. ممکن است ابزار و افزونه هایی نیز توسعه پیدا کنند. اما در یک نگاه کلی میتوان گفت مهاجرت بین این دو فریم ورک زیاد ساده نیست. تفاوت های فنی و ساختاری زیادی وجود دارند، مثلاً الگوی دیتابیس ها برای ذخیره اطلاعات کاربران، مبتنی بودن بر فریم

مهاجرت از ASP.NET Identity به Membership

تیم ASP.NET و مايكروسافت هنوز راهنمایی رسمي، برای این مقوله ارائه نکرده اند. گرچه پست های وبلاگ ها و منابع مختلفی وجود دارند که از جنبه های مختلفی به این مقوله پرداخته اند. اميدواریم تا در آینده نزدیک مايكروسافت راهنمایی های لازم را منتشر کند، ممکن است ابزار و افزونه هایی نیز توسعه پیدا کنند. اما در یک نگاه کلی میتوان گفت مهاجرت بین این دو فریم ورک زیاد ساده نیست. تفاوت های فنی و ساختاری زیادی وجود دارند، مثلاً الگوی دیتابیس ها برای ذخیره اطلاعات کاربران، مبتنی بودن بر فریم

ورک OWIN و غیره. اگر قصد اجرای پروژه جدیدی را دارید پیشنهاد می‌کنم از فریم ورک جدید مایکروسافت ASP.NET Identity استفاده کنید.

[Create an ASP.NET MVC 5 App with Facebook and Google OAuth2 and OpenID Sign-on](#) قدم‌های بعدی

در این مقاله خواهید دید چگونه اطلاعات پروفایل را اضافه کنید و چطور از ASP.NET Identity برای احراز هویت کاربران توسط Deploy a Secure ASP.NET MVC app with Membership, OAuth, and SQL Database to a Google و Facebook استفاده کنید.

[Windows Azure Web Site](#)

[Individual User Accounts](#)

[Organizational Accounts](#)

[Customizing profile information in ASP.NET Identity in VS 2013 templates](#)

[Get more information from Social providers used in the VS 2013 project templates](#)

<https://github.com/rustd/AspnetIdentitySample>

پروژه نمونه ASP.NET Identity می‌تواند مفید باشد. در این پروژه نحوه کارکردن با کاربران و نقش‌ها و همچنین نیازهای مدیریتی را بیشتر بررسی می‌کند. این پروژه در VS 2013 می‌باشد و برای نصب در IIS Express نیاز به نصب ندارد. این پروژه شامل یک پروژه ASP.NET MVC 5، یک پروژه Windows Azure Web Site و یک پروژه SQL Database می‌باشد. این پروژه می‌تواند برای ایجاد یک سایت آنلاین مفید باشد.

نظرات خوانندگان

نوبت‌دهنده: ناظم
تاریخ: ۱۳۹۲/۱۰/۱۴ ۱۱:۴۶

سلام دوست عزیز
به خاطر این مطلب بسیار خوبتون سپاسگذارم.
آیا کتاب یا راهنمای جامعی برای کسی مثل من که اصلا تجربه کار با هیچ کدام از این سیستم‌های مدیریت کاربر را ندارم سراغ دارین؟ لینکهای بالا همسنون موردنی هستن

نوبت‌دهنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۱۴ ۱۲:۳۰

مباحث ابتدایی Forms Authentication در ASP.NET ۱.x مربوط است به [دوره‌ای در این مورد](#). همچنین در MVC هم قابل استفاده است ([کتابی در این مورد](#) و [مباحث membership](#)). مباحث ۲.x در ASP.NET مربوط است به [کتابی در این مورد](#).

نوبت‌دهنده: مهران
تاریخ: ۱۳۹۲/۱۰/۱۴ ۲۱:۳۹

سلام
با تشکر از مطالب و ارائه خوبتون؛
خواستم بدونم چطور میشه، کاربر را بر اساس عملیات کنترلر (Actions) تعیین دسترسی کرد؟

نوبت‌دهنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۴ ۲۱:۴۴

تا حال با کتاب یا دوره جامعی درباره ASP.NET Identity مواجه نشدم، اگر منبع مناسبی پیدا کنم به اشتراک میدارم. در چند پست آتی بیشتر درباره این فریم ورک صحبت خواهم کرد و مثال‌های عملی نیز در نظر خواهم گرفت

نوبت‌دهنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۴ ۲۳:۲۷

سوالتون رو دقیقا متوجه نشدم. از خاصیت Authorize میتوانید استفاده کنید، که قابل اعمال بر روی تک تک اکشن‌های کنترلر است. خاصیت AllowAnonymous برای دسترسی عمومی استفاده می‌شود. برای اطلاعات بیشتر درباره نحوه استفاده از OWIN و ساختار کلی ASP.NET Identity لطفا به لینک‌های ضمیمه شده مراجعه کنید.

```
[Authorize]
public controller account
{
    public ActionResult Index() { }
    public ActionResult Manage() { }

    [AllowAnonymous]
    public ActionResult Info() { }
}

[Authorize(Roles="Admin")]
public controller admin
{
    public ActionResult Index() { }
}
```

نوبت‌دهنده: مهران
تاریخ: ۱۳۹۲/۱۰/۱۵ ۰:۱

فکر میکنم با مطالعه بیشتر Claims Based که در مطلب اشاره کردید، به راه حل مورد نظر برسم.

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۵ ۲:۱۶

درسته، درباره Claims-based authorization هم یک یا دو پست مینویسم.

نویسنده: وحید
تاریخ: ۱۳۹۲/۱۰/۲۰ ۲:۱۸

با تشکر از مطلب مفیدتون سوالی داشتم
یک جا گفتید " مثلا الگوی دیتابیس‌ها برای ذخیره اطلاعات کاربران " منظورتون از دیتابیس Table هست؟
چرا از کد زیر task برای خروجی استفاده کردید

```
async Task<ActionResult> Register
```

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۲۱ ۶:۴۹

منظور دیتابیس سیستم عضویت است، همانطور که گفته شد این دیتابیس توسط EF ساخته میشود، بنابراین جداول، فیلدها و دیگر موارد را میتوانید سفارشی کنید.

همانطور که از امضای این متدها مشخص است، عملیات بصورت Async پردازش میشوند. برای اطلاعات بیشتر به [این لینک نمونه](#) مراجعه کنید.

نویسنده: Programmer
تاریخ: ۱۳۹۲/۱۰/۲۳ ۱۷:۱۸

با عرض سلام و تشکر بابت پست‌های مفید که یقیناً خیلی برash زحمت میکشید.

اگر ممکنه کمی در مورد ساختار Identity و کلاس‌ها و اینترفیس‌ها و نحوه کار با آنها بصورتی که بتونیم خودمون هم Custom Implement اش رو انجام بدیم توضیح بدید.

اینکه اینترفیس‌هایی چون IUser و IUserStore و IUserPasswordStore و IUserSecurityStampStore و در طرف دیگه IdentityUser و UserStore و UserManager مطابق آموزش‌هایی که در سایت هست در مورد MVC و تعریف لایه‌های مختلف و سرویس‌های مورد نیاز، چطور باید از Identity استفاده کرد که هماهنگ با اون مطالب باشه؟

ممnon

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۲۳ ۲۰:۴

در پست‌های آتی پیاده سازی یک تامین کننده (Provider) برای MySQL را بررسی میکنم. برای اطلاعات بیشتر به [Implementing a Custom MySQL ASP.NET Identity Storage Provider](#) مراجعه کنید.

هنوز مستندات کامل و رسمی برای این فریم ورک عرضه نشده اما مطالب مفید زیادی در اینترنت وجود دارند. چند لینک در زیر لیست شده:

[ASP.NET Core Identity](#)

The good, the bad and the ugly of ASP.NET Identity

درباره تطابق با آموزش‌های سایت: دقیقاً متوجه نشدم منظورتون کدوم الگوها است، اما چند نکته تكمیلی: بصورت پیش فرض وقتی ASP.NET Identity به پروژه اضافه می‌شود کلاسی بنام ApplicationDbContext ایجاد می‌شود که بعنوان DbContext پیش فرض برای دیتابیس عضویت استفاده خواهد شد. اگر موجودیت جدیدی برای پروفایل کاربران بسازید باید عنوان یک DbSet<T> به این کلاس افزوده بشو. اگر نیازی به تفکیک دیتابیس سیستم عضویت و دیتابیس اپلیکیشن نیست، بهتره از یک DbContext استفاده کنید. لایه بندی مدل‌ها، سرویس‌ها و کد دسترسی به داده‌ها هم ساده است چرا که کل سیستم توسط EF Code First مدیریت می‌شود. بنابراین استفاده از الگوهای رایج مانند تزریق وابستگی‌ها و غیره مشابه دیگر سناریوهای EF Code First است.

نویسنده: کامران سادی
تاریخ: ۱۳۹۲/۱۰/۲۹ ۱۷:۱

با سلام.
بنده می‌خواستم بدونم آیا در .NET ورژن 4 هم میتوانیم استفاده کنیم؟
با ویژوال استادیو 2012

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۲۹ ۱۸:۴۰

با نصب پکیج‌های مربوط به ASP.NET Identity و غیرفعال کردن Forms Auth می‌توانید همچین کاری بکنید اما توصیه نمی‌شود. سیستم Identity اکثر عملیاتش را بصورت Async انجام میدهد که نیاز به .NET 4.5 دارد. دلایل دیگه ای هم وجود دارد که اگر یک جستجوی ساده در اینترنت بکنید مطالب خوبی در این باره پیدا می‌کنید، مثلاً لینک زیر:

<http://stackoverflow.com/questions/19237285/using-asp-net-identity-in-mvc-4>

نویسنده: سوران
تاریخ: ۱۳۹۲/۱۲/۰۱ ۱۲:۳۷

با تشکر از مطالب آموزنده شما ،
من در ارتباط با ارث بری از کلاس IdentityUser یک سوال داشتم. با توجه به نمونه کدهای تمپلیت vs2013 یک کلاس ApplicationUser از کلاس IdentityUser ارث بری می‌کند و DbContext هم مربوط به این کلاس می‌شود، یعنی به صورت زیر ApplicationDbContext : IdentityDbContext<ApplicationUser>

حال سوال من اینه که اگه چند کلاس داشته باشیم که بخوایم از IdentityUser ارث بری داشته باشند، چطور می‌شون اونها را در یک DbContext استفاده کرد؟
اگر مقاله یا مثالی در این مورد معرفی کنید ممنون می‌شم .
با تشکر

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۲/۰۳ ۵:۲۴

کلاس کاربر:

```
public class AppUser : IdentityUser
{
    public string Email { get; set; }
    public string ConfirmationToken { get; set; }
    public bool IsConfirmed { get; set; }

    public virtual UserProfile Profile { get; set; }
}
```

کلاس پروفایل کاربر:

```
public class UserProfile
{
    public int Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }

    public DateTime? Birthday { get; set; }

    public byte[] Avatar { get; set; }
}
```

کلاس کانتکست دیتابیس:

```
public class SampleDbContext : IdentityDbContext
{
    public SampleDbContext() : base("DefaultConnection") { }

    static SampleDbContext()
    {
        Database.SetInitializer(new DropCreateDatabaseIfModelChanges<SampleDbContext>());
    }

    public DbSet<UserProfile> UserProfiles { get; set; }
    public DbSet<Customer> Customers { get; set; }
    public DbSet<Product> Products { get; set; }
    ...
}
```

این کلاس‌ها می‌توانن تو لایه دیگری مثل Domain Models تعریف بشن.

نویسنده: رضا گرمارودی
تاریخ: ۹:۲۱ ۱۳۹۲/۱۲/۰۳

سلام؛ Identity در کار با SQLCE مشکل داره! هنگام چک کردن نام کاربری و کلمه عبور پیغامی مبنی بر استفاده از تابع Lowercase میدهد که گویا در SqlCe به شیوه دیگری باید صدا زده شود!

نویسنده: وحید نصیری
تاریخ: ۹:۳۶ ۱۳۹۲/۱۲/۰۳

این پژوهش سورس باز هست. مشکلات آن را برای رفع در اینجا گزارش کنید. نحوه گزارش مشکل هم باید کمی فنی باشد.
حداقل جزئیات exception و stack trace آن باید گزارش شوند.

نویسنده: احمد
تاریخ: ۱۰:۲۸ ۱۳۹۳/۰۲/۲۳

سلام
آیا در پروژه‌های windows application که از wcf استفاده می‌کنند هم می‌توانیم از این سیستم استفاده کنیم؟

نویسنده: داریوش حمیدی
تاریخ: ۲۰:۴ ۱۳۹۳/۰۷/۱۸

سلام ; در کلاس `IdentityUser` این خصوصیت دو به چی اشاره دارند ؟
1-SecurityStamp

این مقدار `Random` تغیر میکنید وقتی که اطلاعات کاربری تغیر پیدا میکند مثل تغیر رمز عبور ..
2-TwoFactorEnabled

نویسنده: وحید نصیری
تاریخ: ۱۳:۴۶ ۱۳۹۳/۰۹/۲۹

« [اعمال تزریق وابستگی‌ها به مثال رسمی](#) »

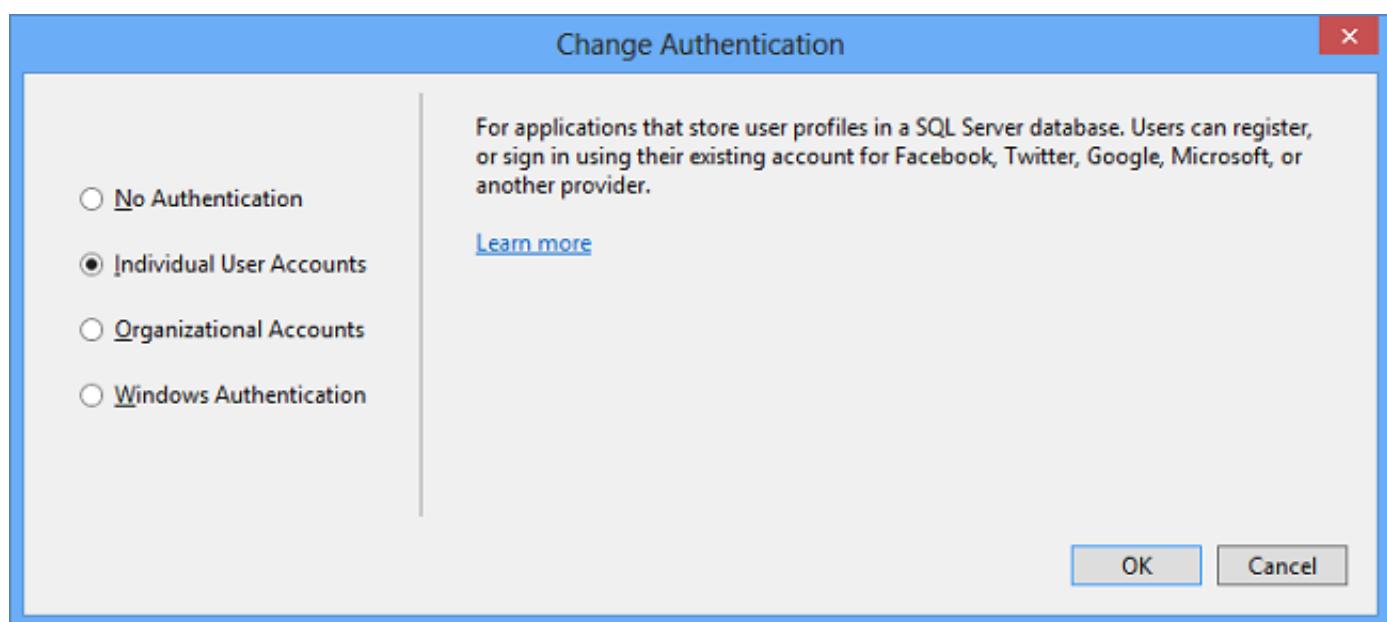
ویژوال استودیو 2013 چندین گزینه برای احراز هویت در قالب‌های پیش فرض پروژه‌های ASP.NET Web Forms, MVC, Web API ارائه می‌کند:

[No Authentication](#)

[Individual User Accounts](#)

[Organizational Accounts](#)

[Windows Authentication](#)



No Authentication

اگر گزینه **No Authentication** را انتخاب کنید، پروژه ایجاد شده صفحاتی را برای ورود به سایت نخواهد ساخت. همچنین رابط کاربری ای برای نمایش کاربر فعلی، کلاس‌های موجودیت‌ها برای یک دیتابیس عضویت و رشته‌های اتصالی نیز وجود نخواهد داشت.

[Individual User Accounts](#)

اگر گزینه **Individual User Accounts** را انتخاب کنید، اپلیکیشن شما برای استفاده از ASP.NET Identity (که پیش از این با نام ASP.NET Membership شناخته می‌شد) پیکربندی می‌شود. کاربران را قادر می‌سازد تا با ساختن حساب کاربری جدیدی در سایت و یا با استفاده از تامین کننده‌های ثالثی مانند Facebook, Google و غیره به سایت وارد شوند. این فریم ورک برای ذخیره‌ی داده‌های پروفایل کاربران، بصورت پیش فرض از یک دیتابیس SQL Server LocalDB استفاده می‌کند که می‌توانید بعداً آنرا بر روی Windows Azure SQL Database یا SQL Server نیز منتشر کنید. این قابلیت‌ها در Visual Studio 2013 در نسخه قبلی نیز وجود داشتند، اما کد سیستم عضویت آن مجدداً بازنویسی شده است. این بازنویسی دارای مزایای زیر است:

سیستم عضویت جدید بجای استفاده از مژول ASP.NET Forms Authentication بر پایه OWIN نوشته شده است. این بدین معنا است که از یک مکانیزم احراز هویت واحد می‌توانید در اپلیکیشن‌های ASP.NET Web Forms, MVC, Web API و SignalR استفاده کنید. سیستم عضویت جدید توسط Entity Framework Code First مدیریت می‌شود و شامل تمامی کلاس‌هایی است که نماینده جداول و موجودیت‌ها هستند. این بدین معنا است که روی الگوی دیتابیس کنترل کامل دارید. سفارشی سازی و تغییر اطلاعات کاربران و پروفایل‌هایشان بسیار ساده‌تر است، تنها لازم است برای اعمال تغییرات از Code First Migrations استفاده کنید.

سیستم عضویت جدید بصورت خودکار در تمام قالب‌های پروژه پیش فرض، نصب و پیاده سازی می‌شود. این امکان برای تمام پروژه‌هایی که دات نت فریم ورک 4.5 را هدف قرار می‌دهند وجود دارد. ASP.NET Identity هنگام تولید وب سایت‌های اینترنتی که اکثر کاربرانشان خارجی (External) هستند گزینه خوبی است. اگر سازمان شما از Active Directory و یا Office 365 استفاده می‌کند و می‌خواهید پروژه‌تان قادر باشد تا احراز هویت کارمندان و شرکای تجاری تان را مدیریت کند، **Organizational Accounts** گزینه بهتری است.

برای اطلاعات بیشتر درباره Individual User Accounts به لینک‌های زیر مراجعه کنید:

[asp.net/identity](#)

[Create an ASP.NET MVC 5 App with Facebook and Google OAuth2 and OpenID Sign-on](#)

[Web API - External Authentication Services](#)

[Adding External Logins to your ASP.NET application in Visual Studio 2013](#)

Organizational Accounts

اگر گزینه Windows Identity Foundation را انتخاب کنید پروژه ایجاد شده برای استفاده از (WIF) پیکربندی خواهد شد. این فریم ورک برای احراز هویت کاربران از Windows Azure Active Directory (WAAD) استفاده می‌کند که شامل Office 365 نیز می‌شود.

Windows Authentication

اگر گزینه Windows Authentication را انتخاب کنید، پروژه ایجاد شده برای استفاده از Windows Authentication IIS Module پیکربندی خواهد شد. چنین اپلیکیشنی نام دامنه و نام کاربری را نمایش خواهد که یا از Active Directory می‌آید، یا از یک ماشین محلی (local machine). اما رابط کاربری ای برای ورود به سیستم وجود ندارد؛ چرا که اینگونه اپلیکیشن‌ها برای سایت‌های اینترنتی (Intranet) استفاده خواهند شد.

یک راه دیگر انتخاب گزینه Organizational Accounts زیر شاخه On-Premises است. این گزینه بجای استفاده از مژول Windows Identity Foundation برای احراز هویت استفاده می‌کند. انجام چند مرحله دستی برای پیکربندی این گزینه لازم است، اما WIF امکاناتی را عرضه می‌کند که در مژول احراز هویت ویندوز وجود ندارند. برای مثال، هنگام استفاده از WIF می‌توانید تنظیمات لازم را در Active Directory انجام دهید تا قادر به واکنشی اطلاعات پوشش‌ها باشید (data querying).

گزینه‌های احراز هویت Organizational Accounts

دیالوگ Configure Authentication گزینه‌های متعددی برای احراز هویت توسط Windows Azure Active Directory و Windows Server Active Directory (including Office 365) در اختیارتان می‌گذارد:

[Cloud - Single Organization](#)

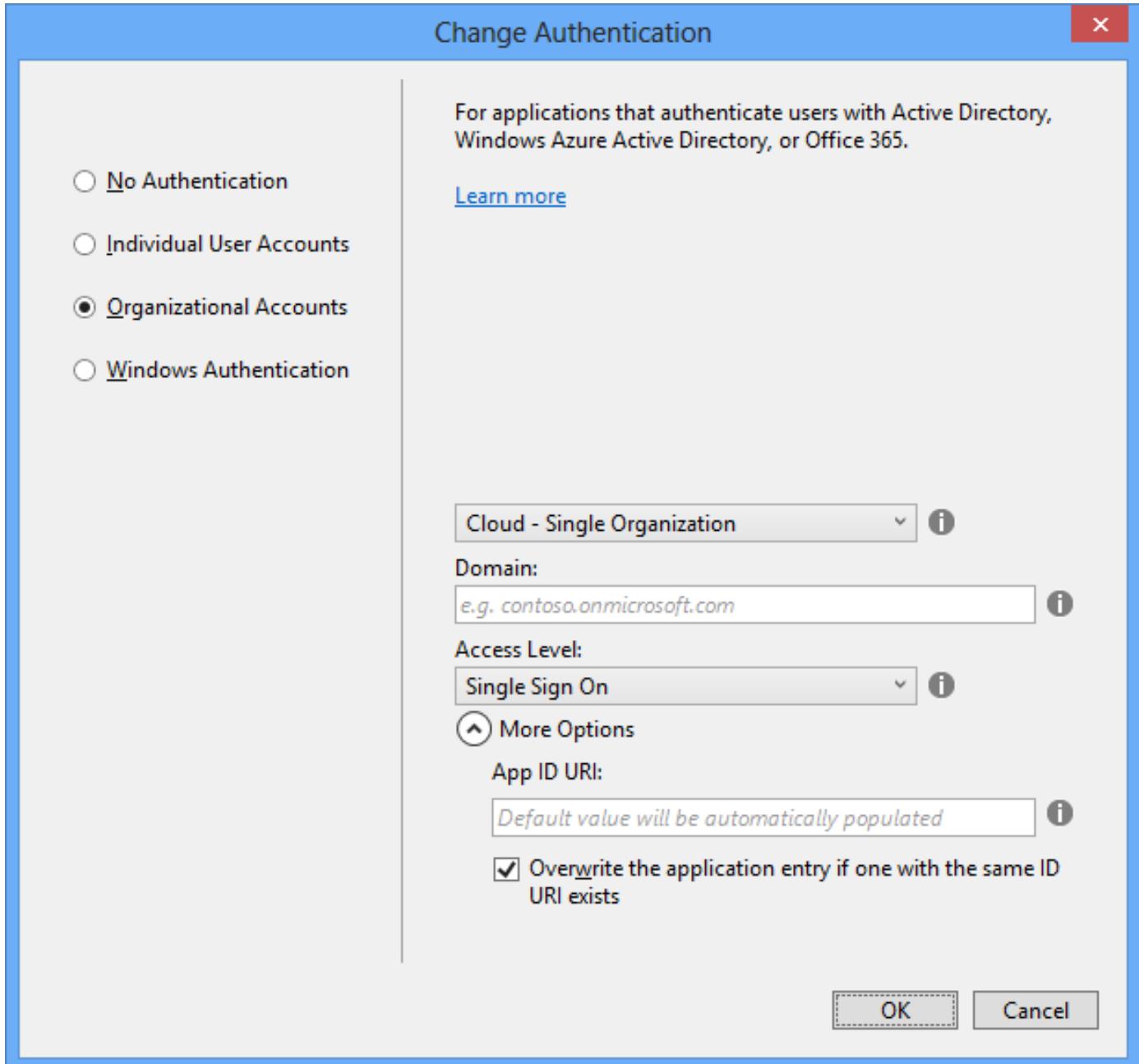
[Cloud - Multi Organization](#)

On-Premises

اگر می‌خواهید یکی از گزینه‌های WAAD را امتحان کنید اما حساب کاربری ای ندارید، روی [این لینک](#) کلیک کنید تا ثبت نام کنید.

نکته: اگر یکی از گزینه‌های WAAW را انتخاب کنید، باید اطلاعات هویتی (Credentials) یک مدیر کل را وارد کنید. برای نسخه نهایی Visual Studio 2013 برنامه‌هایی وجود دارد تا دیگر نیازی نباشد چنان مرافقی را تکمیل کنید. در این صورت ویژوال استودیو تنظیماتی را نمایش خواهد داد که یک مدیر می‌تواند بعداً از آنها استفاده کند تا اپلیکیشن را بصورت دستی در WAAW پیکربندی کند.

Cloud - Single Organization Authentication



از این گزینه برای احراز هویت کاربرانی استفاده کنید که در قالب یک [OWIN Tenant](#) تعریف می‌شوند. برای مثال سایتی با نام [Company.com](#) که برای کارمندان این سازمان از طریق [company.onmicrosoft.com](#) قابل دسترسی خواهد بود. نمی‌توانید [WAAD](#) را طوری پیکربندی کنید که کاربران tenant که دیگر نیز به اپلیکیشن شما دسترسی داشته باشند.

Domain

نام دامنه‌ای در WAAW که می‌خواهید اپلیکیشن را برای آن پیکربندی کنید، مثل [company.onmicrosoft.com](#). اگر از [custom domain](#)

ها استفاده می‌کنید مانند company.onmicrosoft.com بجای company.com می‌توانید این اطلاعات را اینجا وارد کنید.

سطح دسترسی

اگر اپلیکیشن نیاز به کوئری گرفتن یا بروز رسانی اطلاعات پوشش‌ها (directory information) را توسط Graph API دارد، از گزینه‌های Single Sign-On, Read and Write Directory Data و Single Sign-On, Read Directory Data استفاده کنید. در غیر اینصورت گزینه Single Sign-On را رها کنید. برای اطلاعات بیشتر به [Using the Graph API to Application Access Levels](#) و [Single Sign-On](#) مراجعه کنید.

[Query Windows Azure AD](#)

Application ID URI

بصورت پیش فرض، قالب پروژه یک شناسه application ID URI برای شما تولید می‌کند، که این کار با الحاق نام پروژه شما به نام دامنه WAAD صورت می‌گیرد. برای مثال، اگر نام پروژه Example باشد و نام دامنه contoso.onmicrosoft.com، شناسه خروجی More می‌شود. اگر می‌خواهید بصورت دستی این فیلد را مقدار دهی کنید، گزینه [https://contoso.onmicrosoft.com/Example](#) را انتخاب کنید. این شناسه باید با [https://options](#) شروع شود.

بصورت پیش فرض، اگر اپلیکیشنی که در WAAD تهیه و تدارک دیده شده است، شناسه‌ای یکسان با شناسه موجود در پروژه Visual Studio داشته باشد، پروژه شما به اپلیکیشن موجود در WAAD متصل خواهد شد. اگر می‌خواهید تدارکات جدیدی بینید تیک گزینه Overwrite the application entry if one with the same ID already exists را حذف کنید.

اگر تیک این گزینه حذف شده باشد، و ویژوال استودیو اپلیکیشنی با شناسه‌ای یکسان را پیدا کند، عددی به آخر URI اضافه خواهد شد. مثلاً فرض کنید نام پروژه Example است و اپلیکیشنی نیز با شناسه [https://contoso.onmicrosoft.com/Example](#) وجود دارد. در این صورت اپلیکیشن جدیدی با شناسه‌ای مانند [https://contoso.onmicrosoft.com/Example_20130619330903](#) ایجاد می‌شود.

تهیه و تدارک اپلیکیشن در WAAD

برای آنکه یک اپلیکیشن WAAD ایجاد کنید و یا پروژه را به یک اپلیکیشن موجود متصل کنید، ویژوال استودیو به اطلاعات ورود یک مدیر کل برای دامنه مورد نظر، نیاز دارد. هنگامی که در دیالوگ **Configure Authentication** روی OK کلیک می‌کنید، اطلاعات ورود یک مدیر کل از شما درخواست می‌شود و نهایتاً هنگامیکه روی **Create Project** کلیک می‌کنید، ویژوال استودیو اپلیکیشن شما را در WAAD پیکربندی می‌کند.

برای اطلاعات بیشتر درباره نحوه استفاده از مدل احراز هویت Cloud - Single Organization به لینک‌های زیر مراجعه فرمایید:

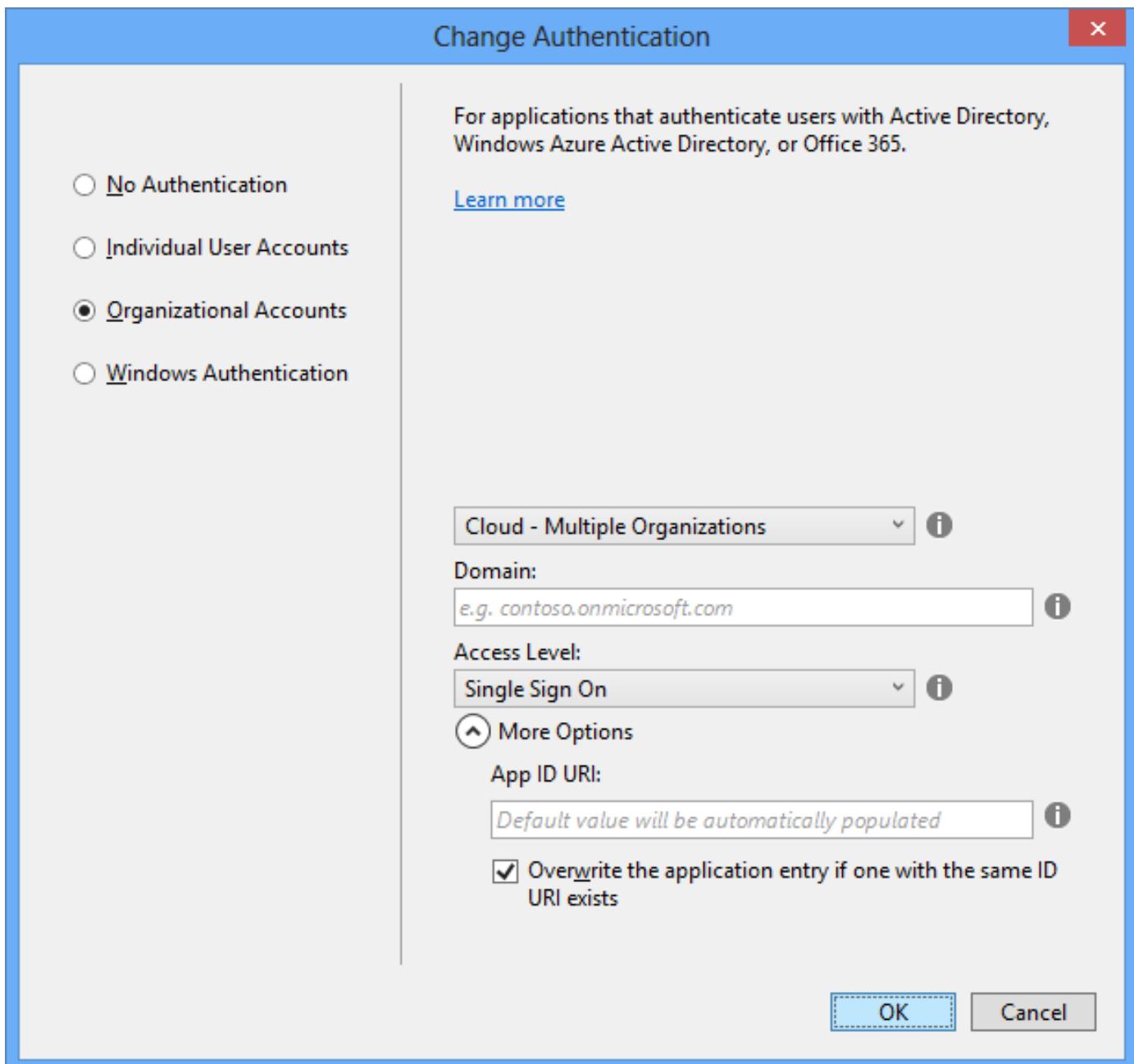
[Windows Azure Authentication](#)

[Adding Sign-On to Your Web Application Using Windows Azure AD](#)

[Developing ASP.NET Apps with Windows Azure Active Directory](#)

مقالات مذکور برای ویژوال استودیو 2013 بروز رسانی نشده اند. برخی از مراحلی که در این مقالات بصورت دستی باید انجام شوند در Visual Studio 2013 مکانیزه شده است.

[Cloud - Multi Organization Authentication](#)

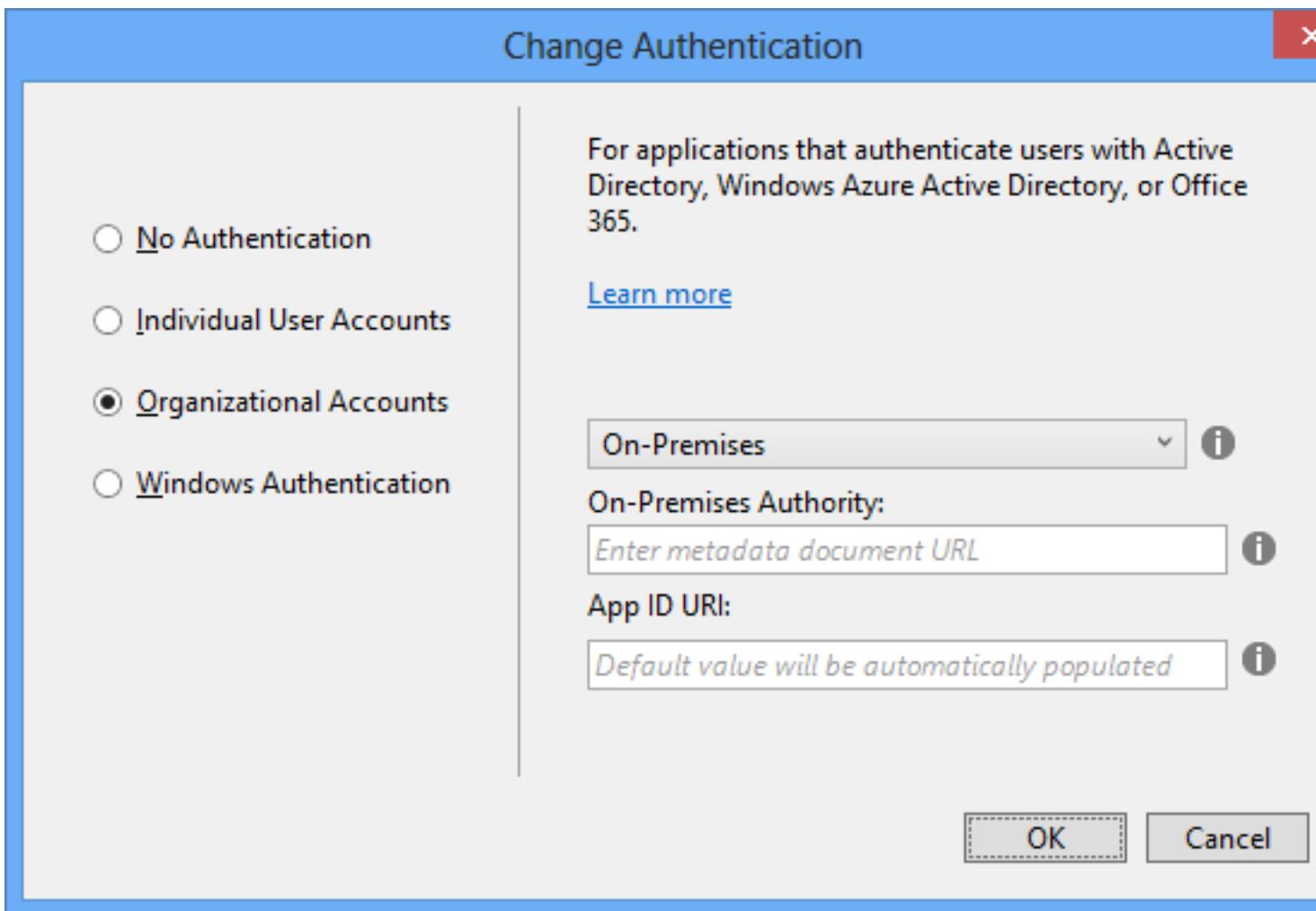


از این گزینه برای احراز هویت کاربرانی استفاده کنید که در tenant WAAD های متعددی تعریف شده‌اند. برای مثال، نام سایت fabrikam.onmicrosoft.com است و برای کارمندان دو سازمان از طریق آدرس‌های contoso.onmicrosoft.com و contoso.onmicrosoft.com قابل دسترسی خواهد بود. نحوه پیکربندی این مدل نیز مانند قسمت قبلی است.

برای اطلاعات بیشتر درباره احراز هویت Cloud - Multi Organization به لینک‌های زیر مراجعه کنید:
[Easy Web App Integration with Windows Azure Active Directory, ASP.NET & Visual Studio](#)

[Developing Multi-Tenant Web Applications with Windows Azure AD](#)

On-Premises Organizational Accounts



این گزینه را هنگامی انتخاب کنید که کاربران در Windows Server Active Directory (AD) تعریف شده اند و نمی‌خواهید از WAAD استفاده کنید. از این مدل برای ایجاد وب سایت‌های اینترنت و اینترانت می‌توانید استفاده کنید. برای یک وب سایت اینترنتی از Active Directory Federation Services (ADFS) استفاده کنید.

برای یک وب سایت اینترنتی، می‌توانید کلا این گزینه را رها کنید و از Windows Authentication استفاده کنید. در صورت استفاده از گزینه Windows Authentication لازم نیست تا آدرس سند متادیتا (metadata document URL) را فراهم کنید، همچنین توجه داشته باشید که Windows Authentication امکان کوئری گرفتن از پوشش‌ها و کنترل سطوح دسترسی در Active Directory Federation Services (ADFS) را ندارد.

On-Premises Authority

آدرس سند متادیتا. این سند اطلاعاتی درباره مختصات Authority دارد که اپلیکیشن از آنها برای به پیش بردن روند احراز هویت و ورود به سایت استفاده می‌کند.

Application ID URI

یک شناسه منحصر به فرد که AD از آن برای شناسایی اپلیکیشن استفاده می‌کند. می‌توانید این فیلد را خالی رها کنید تا ویژوال استودیو بصورت خودکار اپلیکیشنی بدین منظور بسازد.

در این مقاله با مدل‌های مختلف احراز هویت در اپلیکیشن‌های Visual Studio 2013 آشنا شدید و برخی تغییرات و امکانات جدید نیز بررسی شدند. برای اطلاعات تکمیلی به [ASP.NET and Web Tools for Visual Studio 2013 Release Notes](#) مراجعه کنید.

نظرات خوانندگان

نویسنده: هومن
تاریخ: ۱۳۹۳/۰۳/۱۹ ۱۹:۵۴

سلام

- بابت مطلب خوبتون ممنون. من یه سوال داشتم، در پورتال‌های سازمانی و در حالتی که نیاز نداریم پالیسی رو از اکتیو بخونیم، بهتره از حالت آخر یعنی Windows Authentication استفاده کنیم درسته؟ من از این حالت استفاده کردم، و یک مشکل و یک سوال برآم پیش آمد...
 - مشکلم اینه که در سیستمی که با یوزر اکتیو بالا اومده خوب کار میکنه و یوزر رو تشخیص میده، اما روی لپتاپم چون لوکال هست و طبیعیه اکتیو دایرکتوری موجود نیست نمیتونه یوزر رو تشخیص بده (پنجره لاگین مرورگر را باز میکند) و من نمیتونم برنامemo گسترش بدم و برای تست نیاز به اجرای برنامه دارم، این مشکلو چجوری برطرف کنم؟
 - سوالم اینه که ما هیچ کجا اسم دامین یا آی پی سرور رو به برنامه نمیدیم، پس سیستم احراز هویت چجوری دامین رو تشخیص میده و باهاش کار میکنه ؟

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۳/۲۰ ۹:۱۸

اگر از IE استفاده کنید، مشکلی نباید باشه. چون IE با سیستم اعتبارسنجی مبتنی بر ویندوز یکپارچه هست. اگر با IE صفحه لاگین مرورگر باز میشه، به تنظیمات امنیتی اون مراجعه کنید و سایت رو در قسمت trusted sites اضافه کنید:

<http://support.microsoft.com/kb/258063>

سمت سرور هم باید در تنظیمات IIS، گزینه‌ی اعتبارسنجی مبتنی بر ویندوز فعال باشه:

<http://www.asp.net/mvc/tutorials/older-versions/security/authenticating-users-with-windows-authentication-cs>

ابزار ASP.NET برای Windows Azure Active Directory فعال کردن احراز هویت در وب اپلیکیشن هایی که روی Windows Azure میزبانی شده اند را ساده‌تر می‌کند. می‌توانید از Windows Azure Authentication برای احراز هویت کاربران Office 365 استفاده کنید، حساب‌های کاربری را از On-Premise Active Directory خود همگام سازی (Sync) کنید و یا از یک دامنه سفارشی Windows Azure Active Directory بهره ببرید. فعال سازی Windows Azure Authentication شما را طوری پیکربندی می‌کند تا تمامی کاربران را با استفاده از یک Windows Azure Active Directory tenant احراز هویت کنند. این مقاله ساختن یک اپلیکیشن ASP.NET بر اساس organizational accounts را که بر اساس Windows Azure پیکربندی شده و بر روی Active Directory میزبانی می‌شود، مرور می‌کند.

پیش نیاز ها

Visual Studio 2013 RC با Visual Studio Express 2013 RC for Web

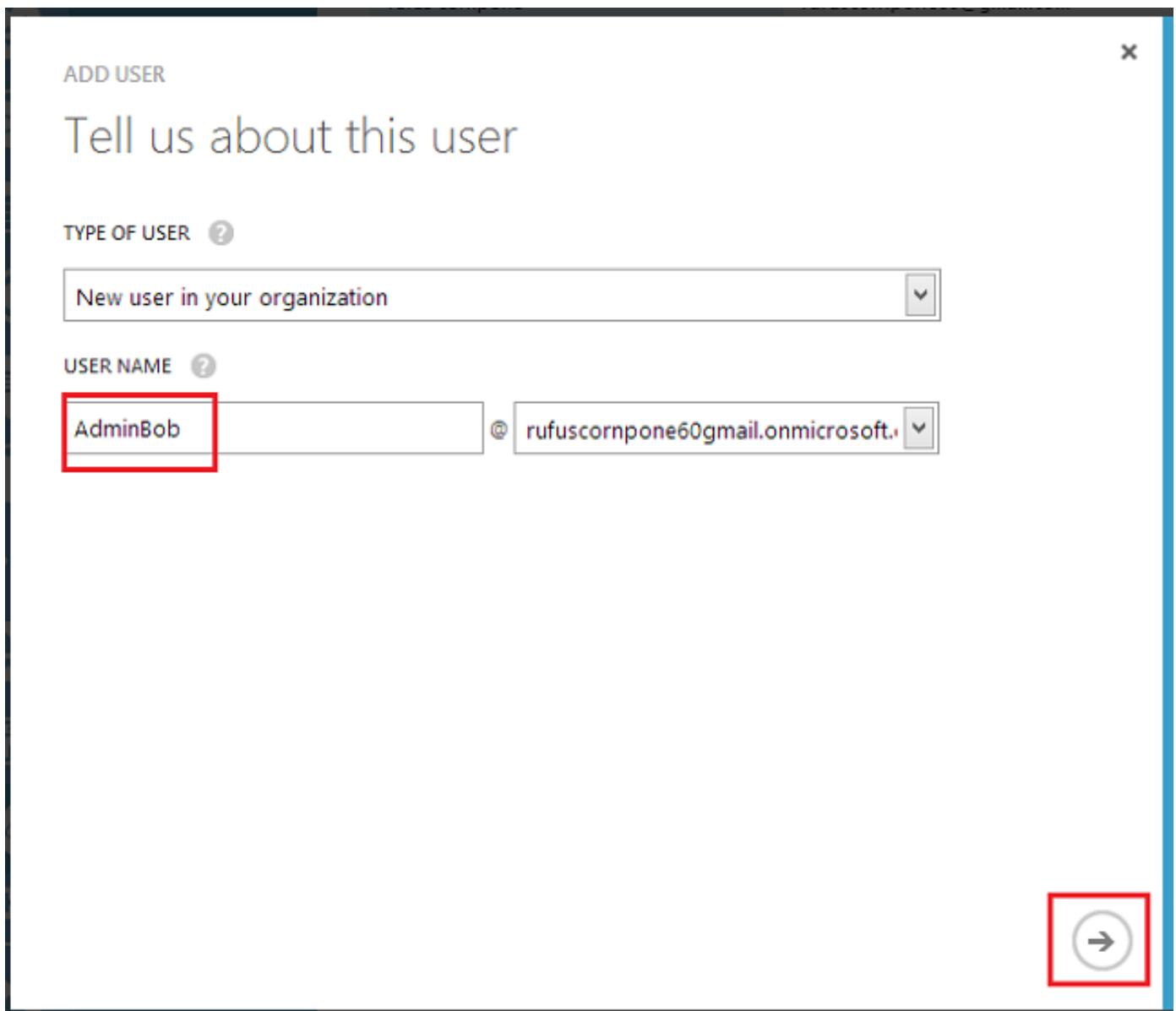
یک حساب کاربری در Windows Azure. می‌توانید یک [حساب رایگان بسازید](#).

یک مدیر کلی به حساب کاربری Active Directory خود اضافه کنید
وارد [Windows Azure Portal](#) شوید.

یک حساب کاربری (AD) Windows Azure Active Directory (AD) انتخاب یا ایجاد کنید. اگر قبل از این کاربری ساخته اید از همان استفاده کنید در غیر اینصورت یک حساب جدید ایجاد کنید. مشترکین Windows Azure یک AD پیش فرض با نام Default خواهند داشت. [Directory](#) در حساب کاربری AD خود یک کاربر جدید در نقش Global Administrator بسازید. اکانت AD خود را انتخاب کنید و Add User را کلیک کنید. برای اطلاعات کامل‌تر به [Managing Windows Azure AD from the Windows Azure Portal 1- Sign Up with an Organizational Account](#) مراجعه کنید.

The screenshot shows the Windows Azure Active Directory management portal. The URL in the browser is <https://manage.windowsazure.com/#Workspaces/Active>. The interface includes a sidebar with icons for Compute, Mobile, Cloud Services, Database, and Storage. A large blue button labeled "Default Directory" is visible. The main area is titled "default directory". It features a navigation bar with "USERS", "APPLICATIONS", "DOMAINS", and "DIRECTORIES". Below this is a table with columns "DISPLAY NAME" and "USER NAME". Two users are listed: "joe admin" (User Name: admin1@rufuscorpone.com) and "rufus corpone" (User Name: rufuscorpone). At the bottom right, there are "ADD USER" and "MANAGE PASSWORD" buttons, with "ADD USER" highlighted by a yellow box.

یک نام کاربری انتخاب کرده و به مرحله بعد بروید.



نام کاربری را وارد کنید و نقش **Global Administrator** را به آن اختصاص دهید. مدیران کلی به یک آدرس ایمیل متناسب هم نیاز دارند. به مرحله بعد بروید.

ADD USER

user profile

FIRST NAME LAST NAME

Bob Admin

DISPLAY NAME

Bob the Admin

ROLE ?

Global Administrator

ALTERNATE EMAIL ADDRESS

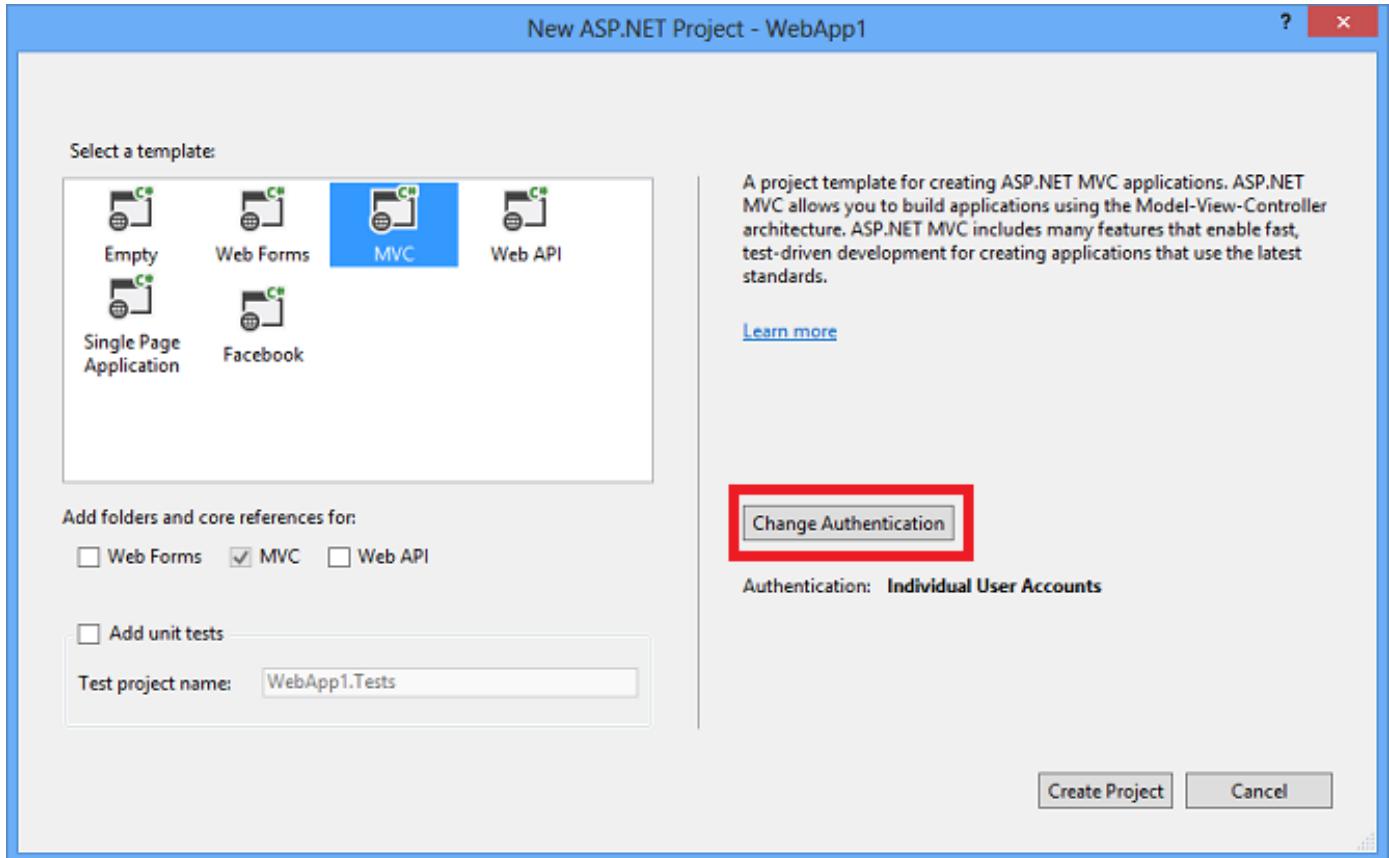
bob@contoso.com

Require Multi-factor Authentication [PREVIEW](#)

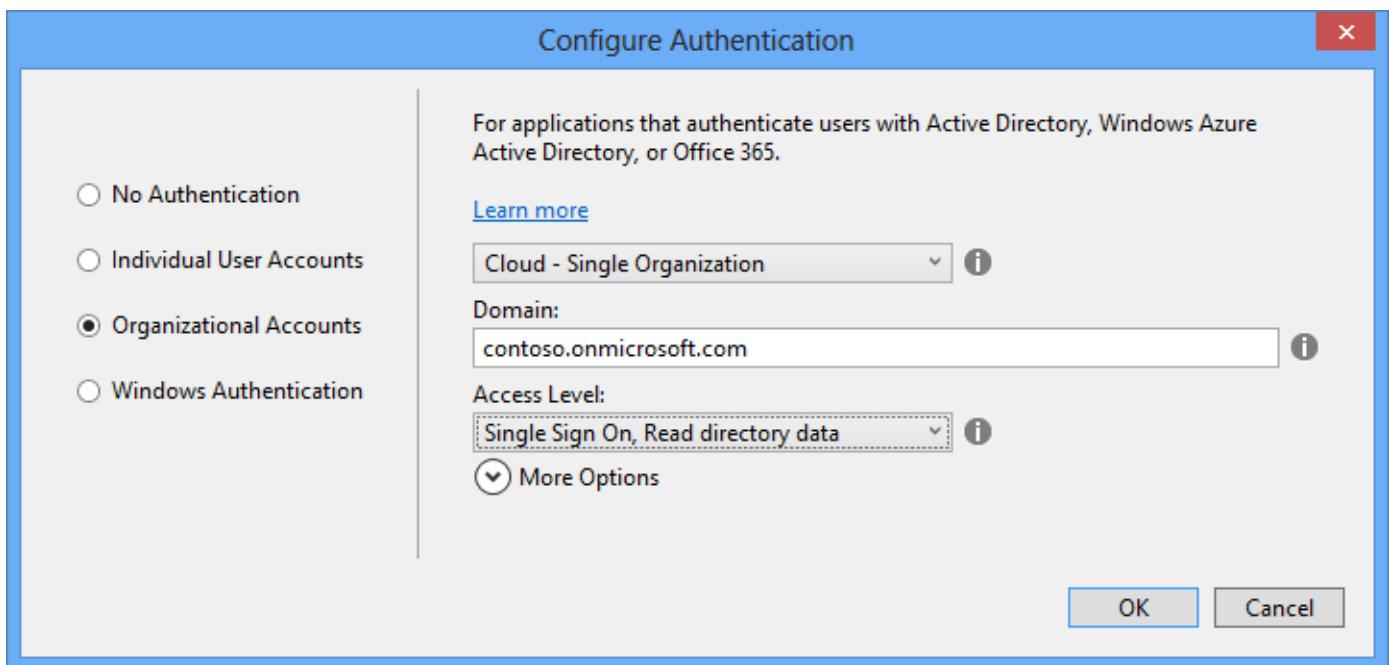


بر روی **Create** کلیک کنید و کلمه‌ی عبور موقتی را کپی کنید. بس از اولین ورود باید کلمه عبور را تغییر دهید.

در ویژوال استودیو یک پروژه جدید ASP.NET Web Forms یا MVC بسازید و روی **Change Authentication** کلیک کنید.



گزینه Single Sign On, Read directory را انتخاب کنید. نام دامنه خود را وارد کنید و سپس گزینه Organizational Accounts را انتخاب کنید. به مرحله بعد بروید.



نکته: در قسمت More Options می توانید قلمرو اپلیکیشن (Application ID URI) را تنظیم کنید. تنظیمات پیش فرض برای اکثر

کاربران مناسب است اما در صورت لزوم می‌توانید آنها را ویرایش کنید، مثلاً از طریق Windows Azure Portal دامنه‌های سفارشی خودتان را تنظیم کنید.

اگر گزینه **Overwrite** را انتخاب کنید اپلیکیشن جدیدی در Windows Azure برای شما ساخته خواهد شد. در غیر اینصورت فریم ورک سعی می‌کند اپلیکیشنی با شناسه یکسان پیدا کند (در پست [متدهای احراز هویت در VS 2013](#) به تنظیمات این قسمت پرداخته شده).

اطلاعات مدیر کلی دامنه در Active Directory خود را وارد کنید (Credentials) و پروژه را با کلیک کردن بر روی **Create Project** بسازید.

با کلیدهای ترکیبی Ctrl + F5، اپلیکیشن را اجرا کنید. مرورگر شما باید یک اخطار SSL Certificate به شما بدهد. این بدین دلیل است که مدرک استفاده شده توسط IIS Express مورد اعتماد (trusted) نیست. این اخطار را بپذیرید و اجازه اجرا را به آن بدهید. پس از آنکه اپلیکیشن خود را روی Windows Azure منتشر کردید، این پیغام دیگر تولید نمی‌شود؛ چرا که Certificate‌های استفاده شده trusted هستند.

با حساب کاربری سازمانی (organizational account) که ایجاد کرده‌اید، وارد شوید.

The screenshot shows a Microsoft sign-in page titled "Sign in" with the URL "https://login.microsoftonline.com". The page includes fields for email and password, a "Keep me signed in" checkbox, and a "Sign in" button. It also features links for account recovery and a note about organizational accounts.

Sign in

Sign in with your organizational account

admin1@rufuscornpone60gmail.onmicrosoft.com

.....

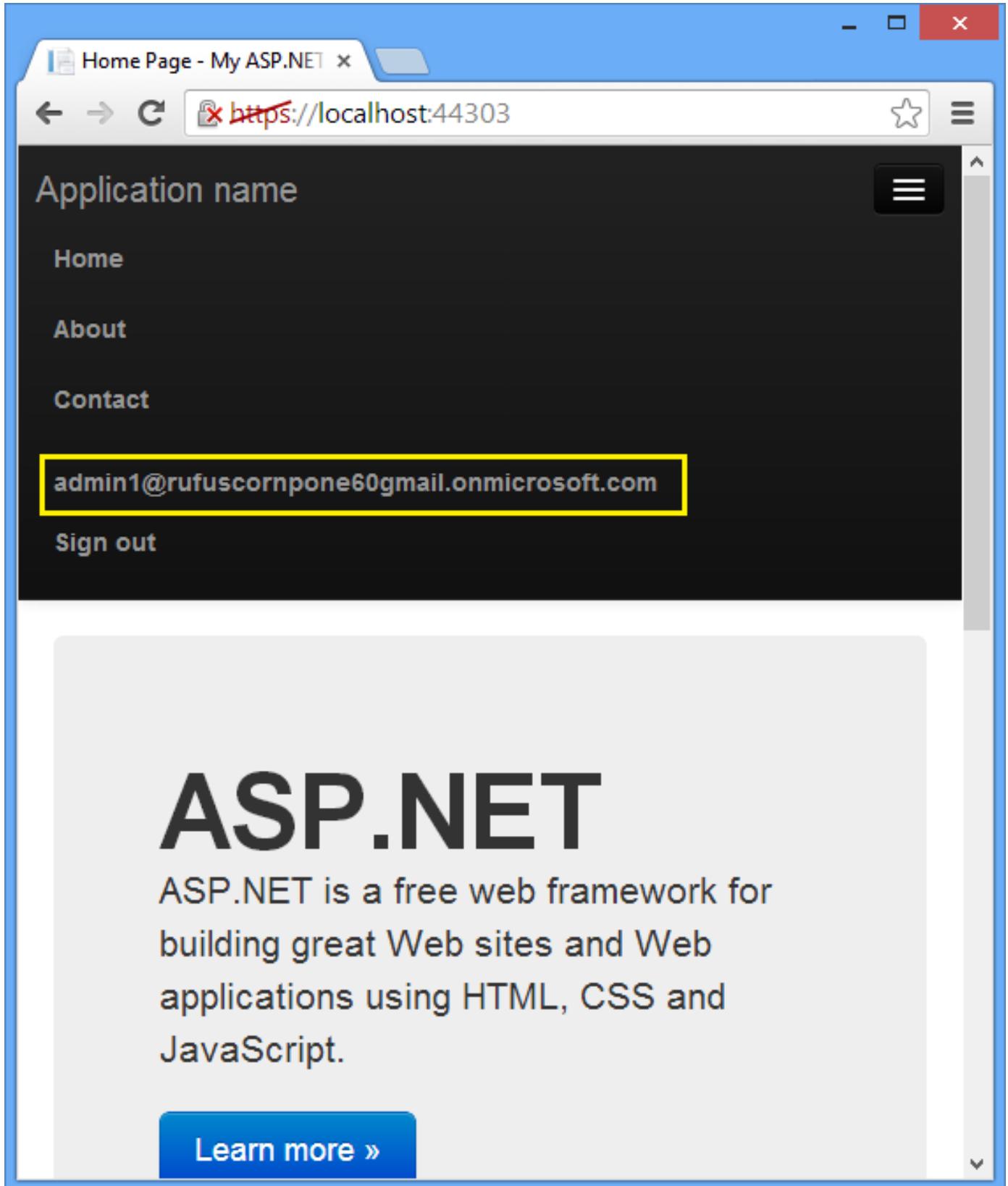
Keep me signed in

Sign in

Can't access your account?

 Organizational accounts that work here can be used anywhere you see this icon. © 2013 Microsoft Legal Privacy Feedback

همانطور که مشاهده می‌کنید هم اکنون به سایت وارد شده اید.



توزيع اپلیکیشن روی Windows Azure

در Windows Azure Portal یک وب سایت را به همراه یک دیتابیس، ایجاد کنید. در پانل سمت چپ صفحه روی **Websites** کلیک کنید و بعد **New** را انتخاب کنید. سپس گزینه **Custom Create** را برگزینید.

NEW WEB SITE - CUSTOM CREATE

Create Web Site

URL

Contoso8  .azurewebsites.net

REGION

West US 

DATABASE

Create a free 20 MB SQL database 

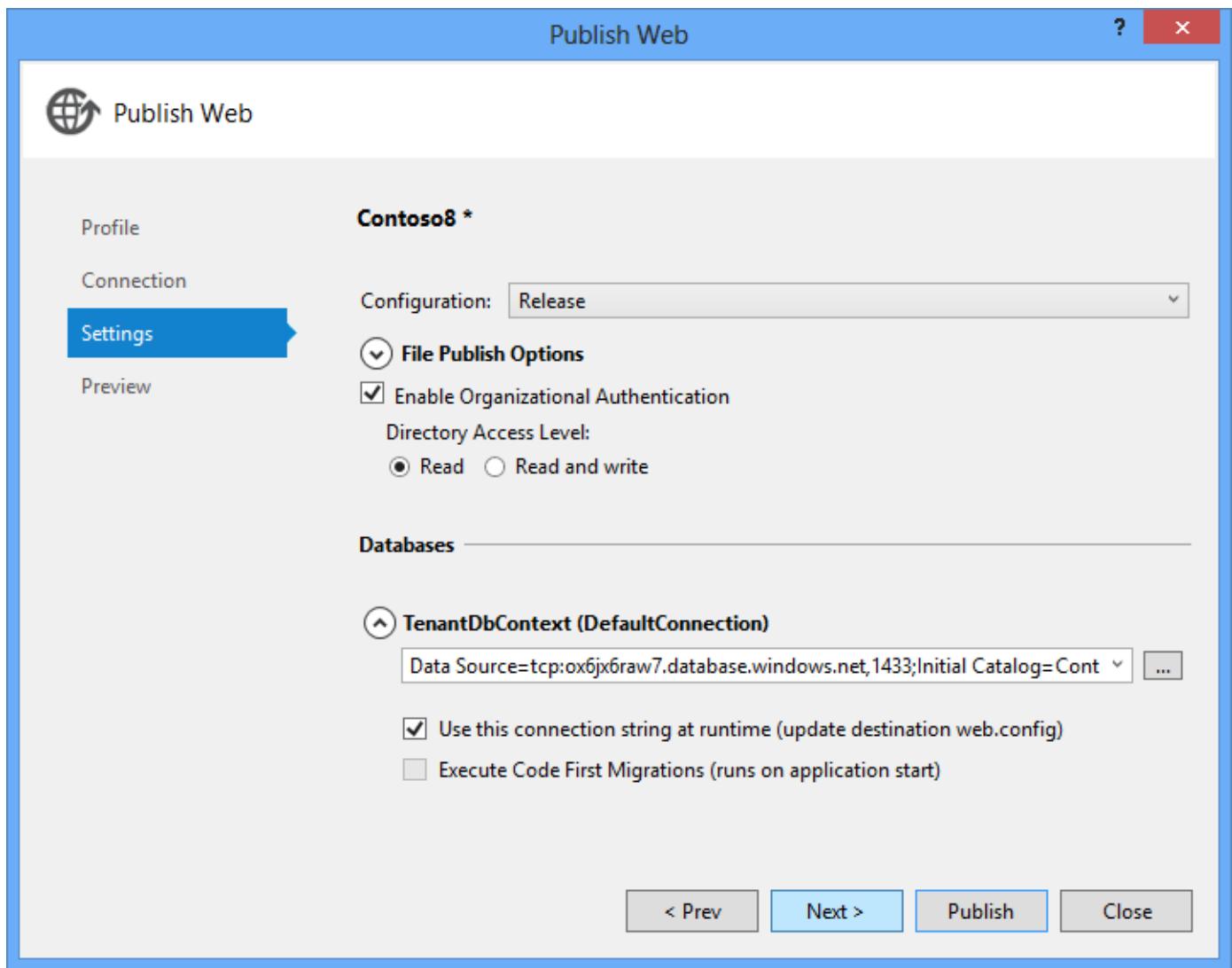
DB CONNECTION STRING NAME 

DefaultConnection

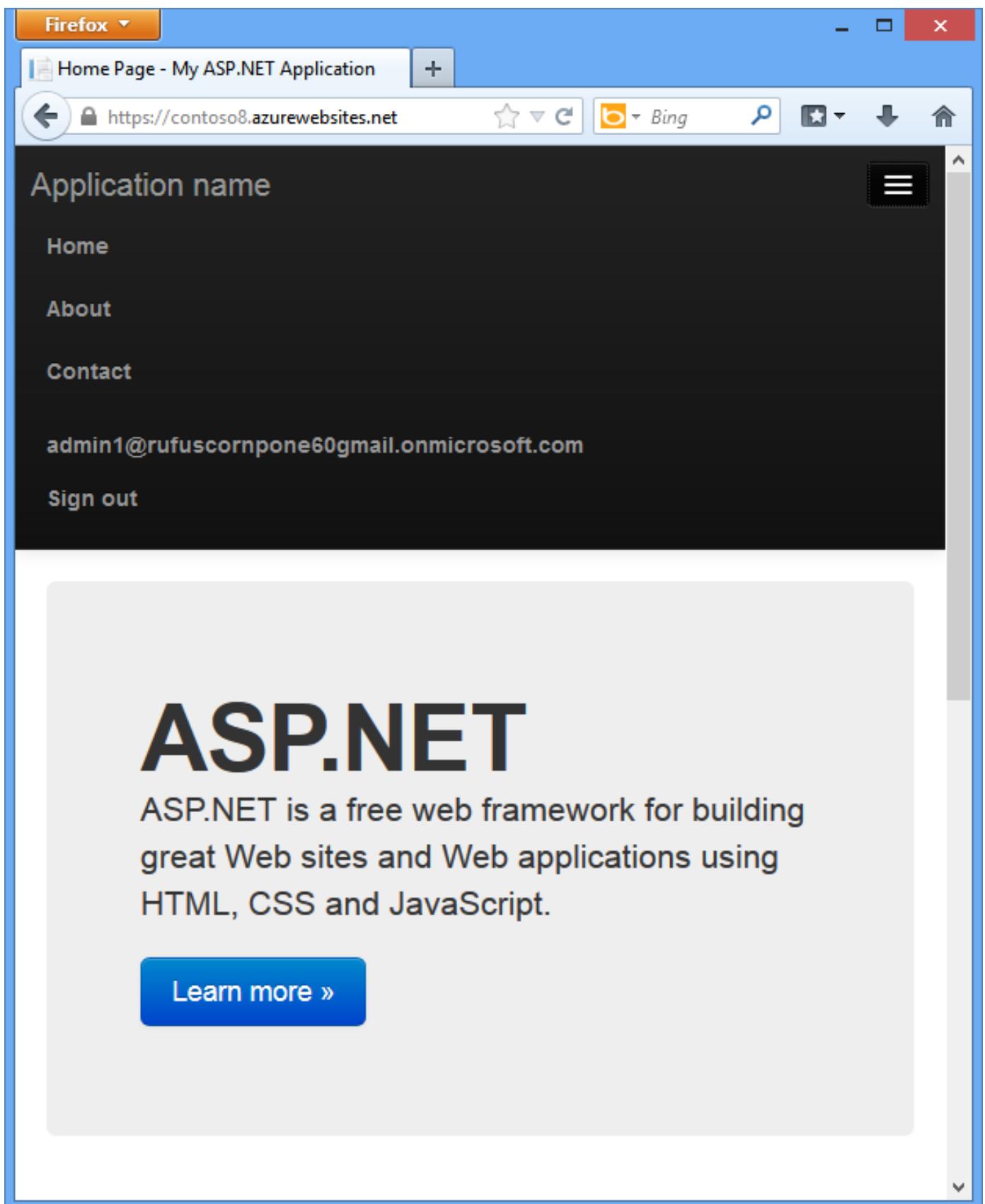
Publish from source control 

 2

اپلیکیشن را روی Windows Azure منتشر کنید. روی پروژه کلیک راست کرده و **Publish** را انتخاب کنید. در مرحله تنظیمات (Settings) مشاهده می‌کنید که احراز هویت حساب‌های سازمانی (organizational accounts) فعال است. همچنین سطح دسترسی به خواندن تنظیم شده است. در قسمت Database رشته اتصال دیتابیس را تنظیم کنید.



حال به وب سایت Windows Azure خود بروید و توسط حساب کاربری ایجاد شده وارد سایت شوید. در این مرحله دیگر نباید خطای امنیتی SSL را دریافت کنید.



خواندن اطلاعات پروفایل کاربران توسط Graph API

قالب پروژه ویژوال استودیو برای UserProfile یک متد و نما بنام organizational accounts به پروژه اضافه کرده است.

[Authorize]

```
public async Task<ActionResult> UserProfile()
{
    string tenantId = ClaimsPrincipal.Current.FindFirst(TenantSchema).Value;
    // Get a token for calling the Windows Azure Active Directory Graph
    AuthenticationContext authContext = new AuthenticationContext(String.Format(CultureInfo.InvariantCulture, LoginUrl, tenantId));
    ClientCredential credential = new ClientCredential(AppPrincipalId, AppKey);
    AuthenticationResult assertionCredential = authContext.AcquireToken(GraphUrl, credential);
    string authHeader = assertionCredential.CreateAuthorizationHeader();
    string requestUrl = String.Format(
        CultureInfo.InvariantCulture,
        GraphUserUrl,
        HttpUtility.UrlEncode(tenantId),
        HttpUtility.UrlEncode(User.Identity.Name));

    HttpClient client = new HttpClient();
    HttpRequestMessage request = new HttpRequestMessage(HttpMethod.Get, requestUrl);
    request.Headers.TryAddWithoutValidation("Authorization", authHeader);
    HttpResponseMessage response = await client.SendAsync(request);
    string responseString = await response.Content.ReadAsStringAsync();
    UserProfile profile = JsonConvert.DeserializeObject<UserProfile>(responseString);

    return View(profile);
}
```

کلیک کردن لینک **UserProfile** اطلاعات پروفایل کاربر جاری را نمایش می‌دهد.

The screenshot shows a web browser window with the title "User Profile - My ASP.NET". The URL in the address bar is <https://con7.azurewebsites.net/Home/UserProfile>. The page content includes:

- Application name: Home
- UserProfile
- Contact
- Email: Bob@rufuscornpone60@gmail.onmicrosoft.com
- Sign out

Below this, there is a table showing user details:

Display Name	Bob Smith User
First Name	Bob
Last Name	Smith

At the bottom left, there is a copyright notice: © 2013 - My ASP.NET Application.

اطلاعات بیشتر

[Managing Windows Azure AD from the Windows Azure Portal 1- Sign Up with an Organizational Account](#)

[Adding Sign-On to Your Web Application Using Windows Azure AD](#)

[Using the Graph API to Query Windows Azure AD](#)

نظرات خوانندگان

نویسنده: حسین
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۲:۲۰

خیلی ممنون از خدمات شما
چند وقت پیش قصد داشتم ثبت نام کنم، زمانی که میخواهید حساب کاربری ایجاد کنید نیاز به شماره موبایل داره تا اجازه بده
مرا حل ثبت نام کامل بشه ولی اصلاً امکان انتخاب کشور ایران وجود ندارد و موفق نشدم

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۶ ۱۳:۱۴

از [حساب‌های کاربری مایکروسافت](#) می‌تونید استفاده کنید.

نویسنده: بهروز
تاریخ: ۱۳۹۲/۱۰/۱۹ ۱۲:۱

ممنون بابت مطلب خوبتون.

لطفاً یکم شفاف‌تر بگین که چطور برآ تو حساب کاربری ایجاد کنیم؟ من هر کاری می‌کنم این صفحه می‌باد!
<https://manage.windowsazure.com/Error/NoSubscriptions>

نویسنده: کامران سادین
تاریخ: ۱۳۹۲/۱۱/۱۲ ۴:۲۲

سلام. من شماره موبایل هم دادم اما اطلاعات مستر کارت و ویزا کارت می‌خواهد!

مدتی قبل سؤال مهمی در سایت مطرح شد، به شرح زیر:

«من از کنترل‌های تلریک استفاده می‌کنم که یک سری اسکریپت را بصورت

`http://localhost:1244/WebResource.axd?d=ak1E6L8AEfPEgIS3T-`

`oXc6mevPfbpi6VRp_ZTP2nBVrnt5UL0FYD3GNWRrDHwANC3VDQ1L8dLAa5g35dzgHyuzAgAguIpYrf-_NXIJwNNu0YRSnH3-`

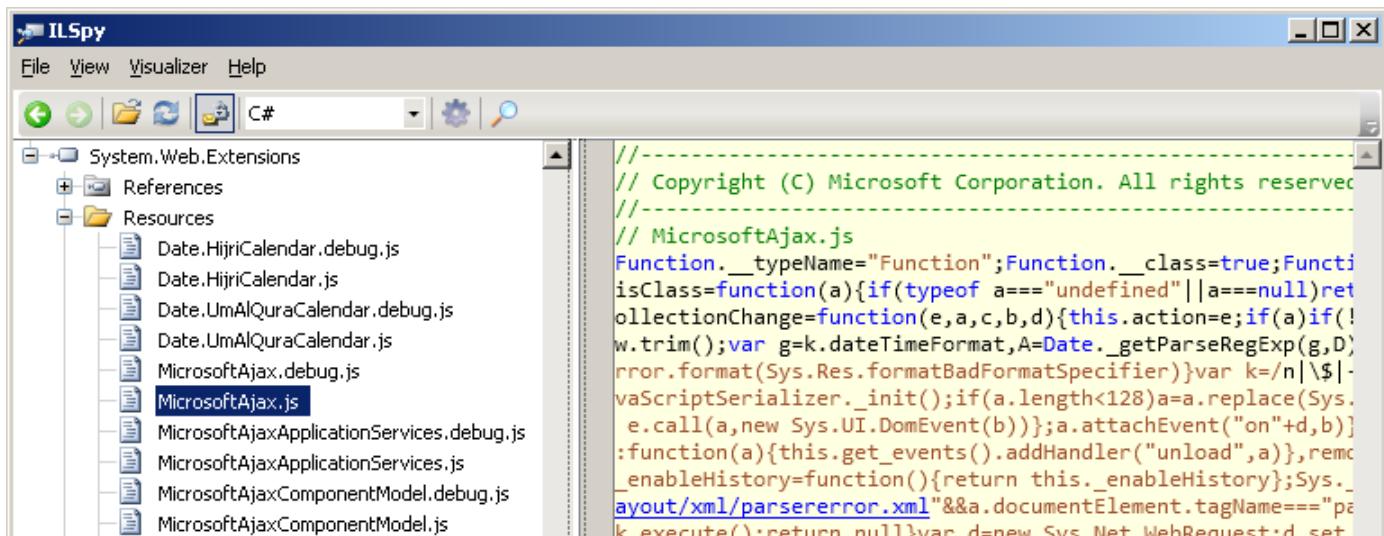
`MgKMfnwKBKF_Lk2E5oeIcLL78uD1Q0se_GxQ2&t=635231470568640000`

به فرم تزریق می‌کند و بعضی وقت‌ها داخلش xp و یا یک سری دستورات اس‌کیوال تولید می‌شوند. در این حالت این مسیرها توسط ISA Server در شبکه داخلی حمله تشخیص داده شده و بلاک خواهند شد و عملاً برنامه از کار می‌افتد. آیا راهی برای خلاصی از دست آن‌ها هست؟»

پاسخ: بله. از دات نت ۳ و نیم به بعد، امکان جایگزینی کامل اسکریپت‌های خودکار مدفون شده در اسمبلی‌ها با فایل‌های استاتیک پیش‌بینی شده‌است که در ادامه نحوه استخراج و کار با آن‌ها را بررسی خواهیم کرد.

الف) یافتن اسکریپت‌های مدفون در اسمبلی‌ها

در ابتدا اسمبلی حاوی کنترل‌های وب فرم مدنظر خود را باید توسط برنامه‌های ILSpy یا Reflector یا امثال آن‌ها گشوده و نام دقیق منبع و همچنین محتوای آن فایل اسکریپت را استخراج کنید. برای مثال:



در این تصویر، اسمبلی استاندارد System.Web.Extensions مورد بررسی قرار گرفته است. برای نمونه اگر بخواهید اسکریپت‌های متناظر با UpdatePanel و ScriptManager را با معادل‌های استاتیک آن‌ها جایگزین کنید، باید دو فایل MicrosoftAjaxWebForms.js و MicrosoftAjax.js را از این اسمبلی استخراج نمائید. (برنامه‌های یاد شده امکان ذخیره سازی منابع را نیز می‌دهند)

ب) وارد کردن ASP.NET به استفاده از نسخه‌ی استاتیک منابع

```
<asp:ScriptManager ID="Scriptmanager1" runat="server">
    <Scripts>
```

```
<asp:ScriptReference Name="MicrosoftAjaxWebForms.js" Assembly="System.Web.Extensions"
    Path("~/staticJS1.js" />
<asp:ScriptReference Name="MicrosoftAjax.js" Assembly="System.Web.Extensions"
Path "~/staticJS2.js" />
</Scripts>
</asp:ScriptManager>
```

فرض کنید دو اسکریپت یاد شده را در فایل‌های staticJS1.js و staticJS2.js در ریشه‌ی سایت خود ذخیره کرده‌اید. اکنون یک ScriptManager را به صفحه اضافه کرده و مطابق کدهای فوق، اسکریپت‌های مدفون شده در اسمبلی System.Web.Extensions را به این فایل‌های استاتیک هدایت کنید. همانطور که عنوان شد نام این مداخل باید دقیقاً با نام موجود در اسمبلی یکی باشد؛ در غیر اینصورت با خطای ذیل مواجه خواهد شد:

The assembly 'System.Web.Extensions' does not contain a Web resource that has the name 'xyz.js'.
Make sure that the resource name is spelled correctly.
Make sure that the application references the correct version of an ASP.NET AJAX Framework assembly.

اکنون اگر برنامه را اجرا کنید (با فرض قرار داشتن یک UpdatePanel و ScriptManager در سورس صفحه، شاهد آدرس‌های طولانی ScriptResource.axd و WebResource.axd نخواهید بود. به صورت خودکار از دو فایل استاتیک تنظیم شده، استفاده می‌شود):

```
<script src="staticJS1.js" type="text/javascript"></script>
<script src="staticJS2.js" type="text/javascript"></script>
```

بدیهی است در صورت نیاز، باید تعاریف سایر اسکریپت‌های مدفون در اسمبلی یاد شده یا اسمبلی System.Web از طریق ScriptManager اضافه کرد. در مورد کنترل‌های ثالث نیز وضع به همین صورت است و استاندارد آن تفاوتی نمی‌کند.

یک نکته‌ی تکمیلی

در مطلب « [ASP.NET 4.5 ScriptManager Improvements in WebForms](#) » مشاهده خواهید کرد که از 4.5 ASP.NET به بعد، طی دو بسته‌ی نیوگت که هر از چندگاهی به روز می‌شوند، کلیه اسکریپت‌های System.Web.Extensions و System.Web از این اسمبلی‌ها نیز قابل دریافت بوده و با استفاده از سیستم bunding & minification می‌توان آن‌ها را فشرده و یکی کرد.

نظرات خوانندگان

نویسنده: سجاد ف
تاریخ: ۱۳۹۲/۱۰/۱۷ ۱۰:۳۱

با تشکر رفراوان. دلم نیومد که بدون ارسال نظر تشکر کنم

نویسنده: آتوسا فتوحی
تاریخ: ۱۳۹۲/۱۰/۱۸ ۹:۳

با سلام و خسته نباشین.

من هم تو پروژه ام به وفور از کامپوننت‌های تلریک استفاده میکنم و این مشکلو دارم. وقتی پروژه رو اجرا میکنم آدرس صفحه از [http://localhost:24434/\(S\(zes4wsco5fczysrbals0een1\)\)/default.aspx](http://localhost:24434/(S(zes4wsco5fczysrbals0een1))/default.aspx) به <http://localhost:24434/default.aspx> تغییر پیدا میکنه.

واسه همه صفحه‌ها اینجوری هست. یعنی علتش همین که شما گفتین میتونه باشه ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۱۸ ۹:۲۱

خیر. قسمتی که به URL اضافه شده، Session ID کاربر هست و زمان استفاده از حالت [cookie less session](#) رخ می‌دهد. از لحاظ امنیتی مشکل دارد و توصیه نمی‌شود.

نویسنده: لیبرتاد
تاریخ: ۱۳۹۲/۱۰/۱۸ ۱۰:۶

کلا" به خاطر همین چیزاست که استفاده از MVC توصیه میشه

نویسنده: بهزاد
تاریخ: ۱۳۹۲/۱۰/۱۸ ۱۲:۴۳

ما برای خودمون یک سیستم ساده نوشته ایم که نه تنها این کار رو می‌کنه بلکه خیلی حرfe ایتر و ساده‌تر این کار رو می‌کنه، مثلاً حتی باهاش میشه تمام فایل‌های یک فریم ورک مثل Bootstrap یا jQuery رو توی اسملی گذاشت و بعد به سادگی بهشون ارجاع داد

روش کار مون اینطوریه که توی اسملی که می‌خواهیم فایلهاش روی وب قابل دسترس باشند، یک attribute تعريف می‌کنیم، توی این تعريف، پسوند فایل‌های مجاز، پیشوند نشانی شون و بعضی چیزای دیگه رو تعیین می‌کنیم در مرحله بعدی، توی فایل web.config یک هندر اضافه می‌کنیم و نهايیتا فایلی رو که می‌خواهیم آدرس بدیم، با استفاده از یک تابع خاص آدرس می‌ديم، نهايیتا نشانی به صورت زير توليد میشه:

/Dynamic/Farayan.Web.JQuery/jquery-1.5.js

اگر مایل باشین، می‌تونم فایل هاش رو برآتون توی همین سایت بذارم، از دات نت هم بهتره!

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۱۸ ۱۲:۵۷

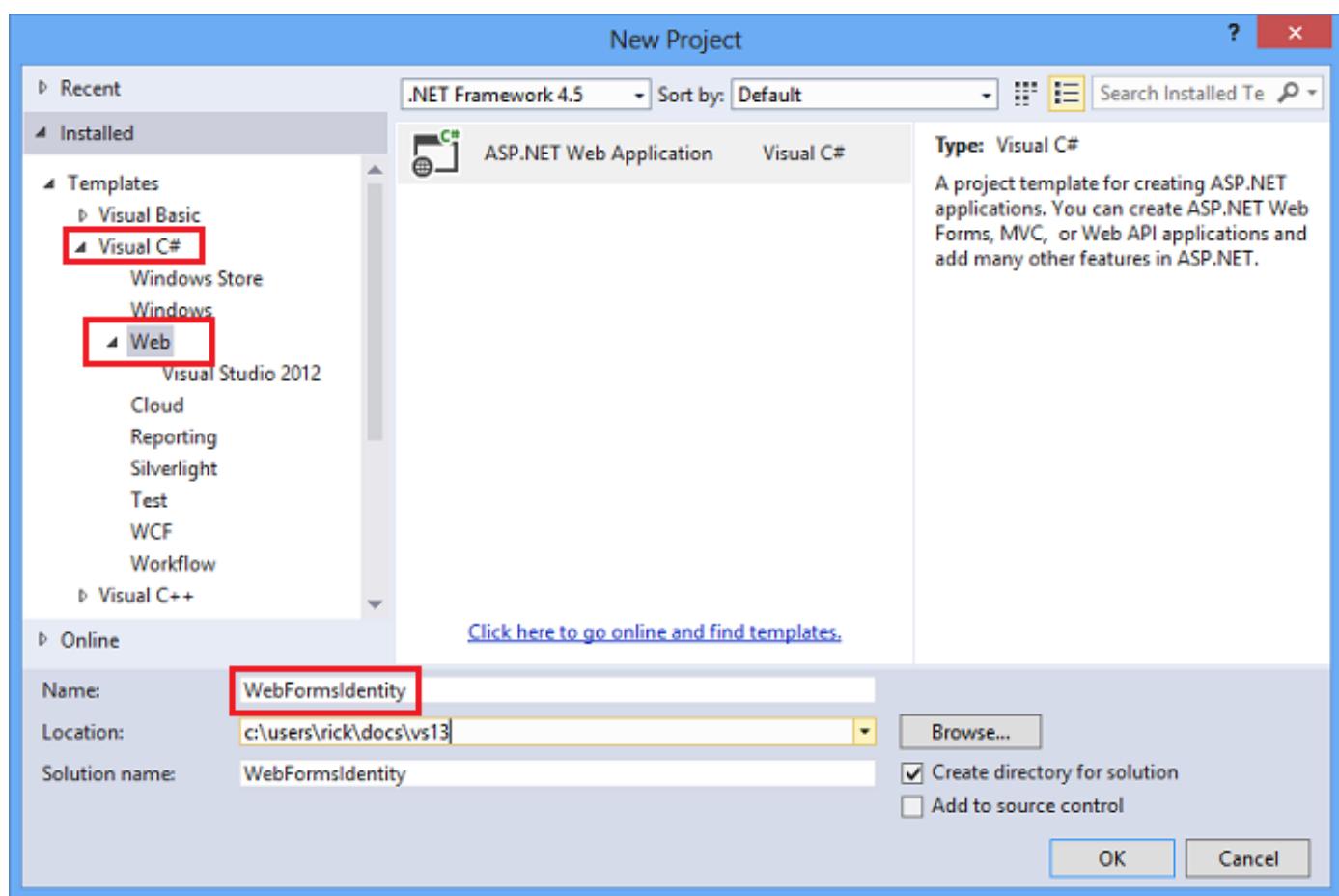
روزهای اولی که همه می‌رن سراغ وب فرم، دوست دارند همه چیز را داخل اسملی‌ها قرار دهند. فکر می‌کنند اینطوری بهتر است. بعد متوجه می‌شوند که به روز رسانی آن‌ها سخت می‌شود، WebResource.axd های طولانی مشکل‌زا درست می‌کند (مطلوب جاری) و از همه مهم‌تر تعداد ارجاعاتی که در یک صفحه اضافه می‌شوند، زیاد هست و روی کارآیی سایت تاثیر منفی می‌گذارد (تعداد رفت و برگشت‌های زیادی را به سرور برای دریافت فایل‌های هر صفحه ایجاد می‌کند). بعد به این نتیجه می‌رسند که بد

نیست این فایل‌ها را با هم یکی کنیم؟ (داخل یک اسمبلی گذاشتن به معنای یکی کردن فایل‌ها نیست) فشرده سازی خود فایل‌ها با حذف فوائل یا کوتاه کردن نام متغیرها چطور؟ اگر در این بین، سرور این‌ها را به صورت gzip ارائه دهد که خیلی خوب خواهد شد. اگر هدر کش کردن به مدت یکسال را هم در سمت کلاینت اضافه کنیم که عالی و به علاوه اگر فایلی در سمت سرور به روز شد، به صورت خودکار این کش دیگر قابل استفاده نباشد و به روز شود. اینجا است که سیستم [bundling](#) & [minification](#) دات نت متولد می‌شود. هم در MVC قابل استفاده است و هم در وب فرم‌ها. بنابراین طراحی سیستمی بهینه جهت ارائه اسکریپت‌ها و شیوه‌نامه‌ها، فراتر است از صرفا قرار دادن چند فایل در یک اسمبلی و ارائه خام آن‌ها.

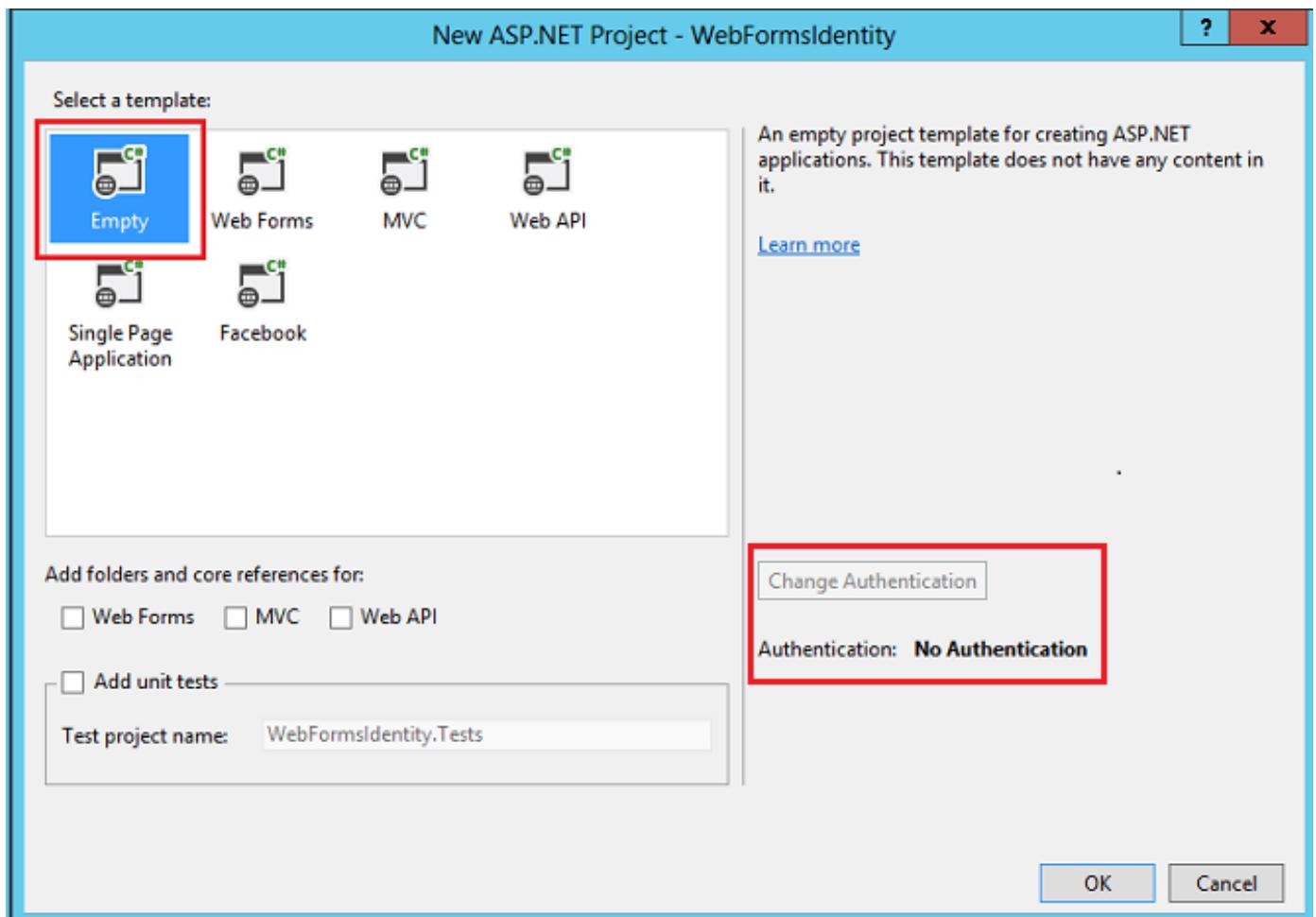
با نصب و اجرای [Visual Studio 2013](#) یا [Visual Studio 2013 Express for Web](#) شروع کنید.

یک پروژه جدید بسازید (از صفحه شروع یا منوی فایل)

گزینه C# و سپس **ASP.NET Web Application** را انتخاب کنید. نام پروژه را به "WebFormsIdentity" تغییر داده و OK کنید.

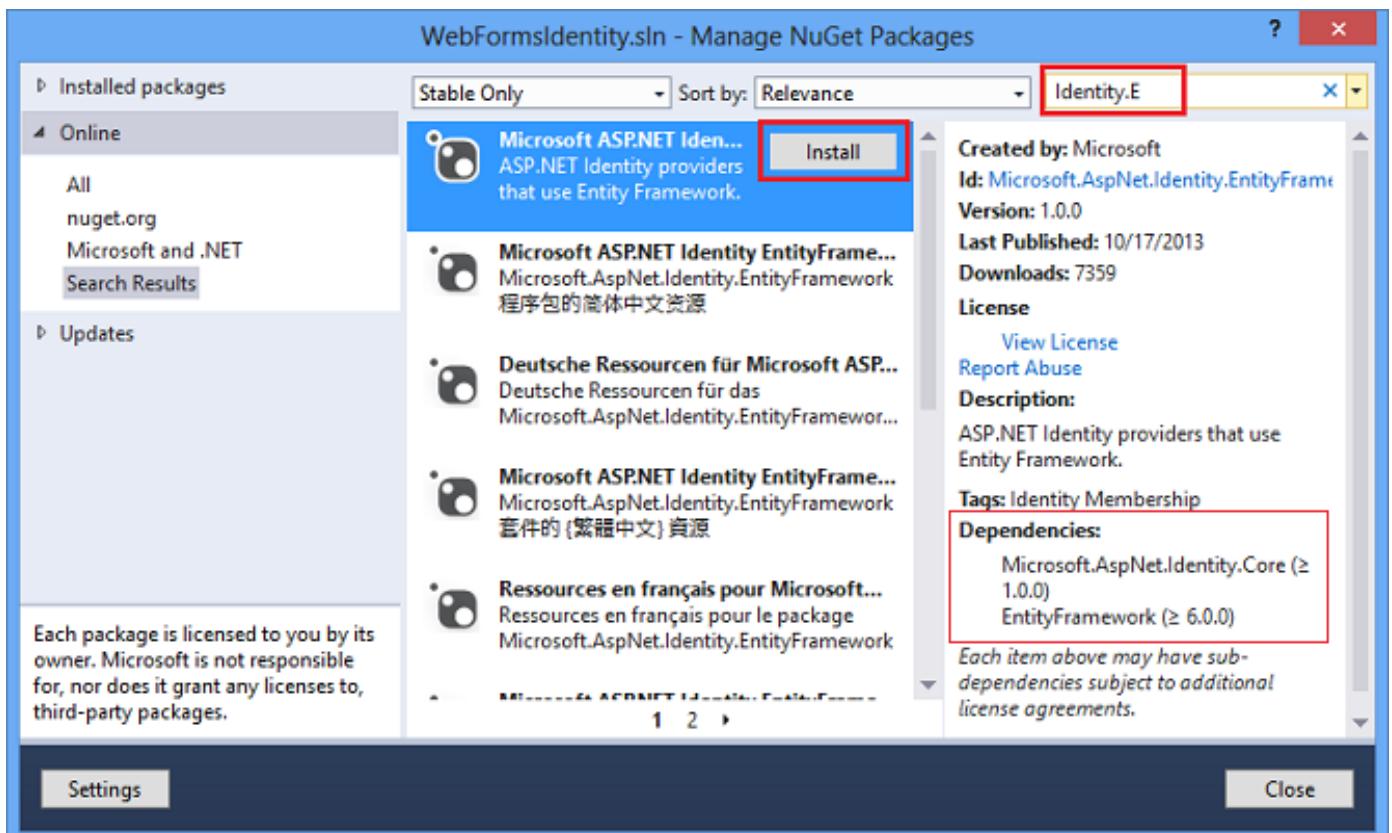


در دیالوگ جدید ASP.NET گزینه **Empty** را انتخاب کنید.



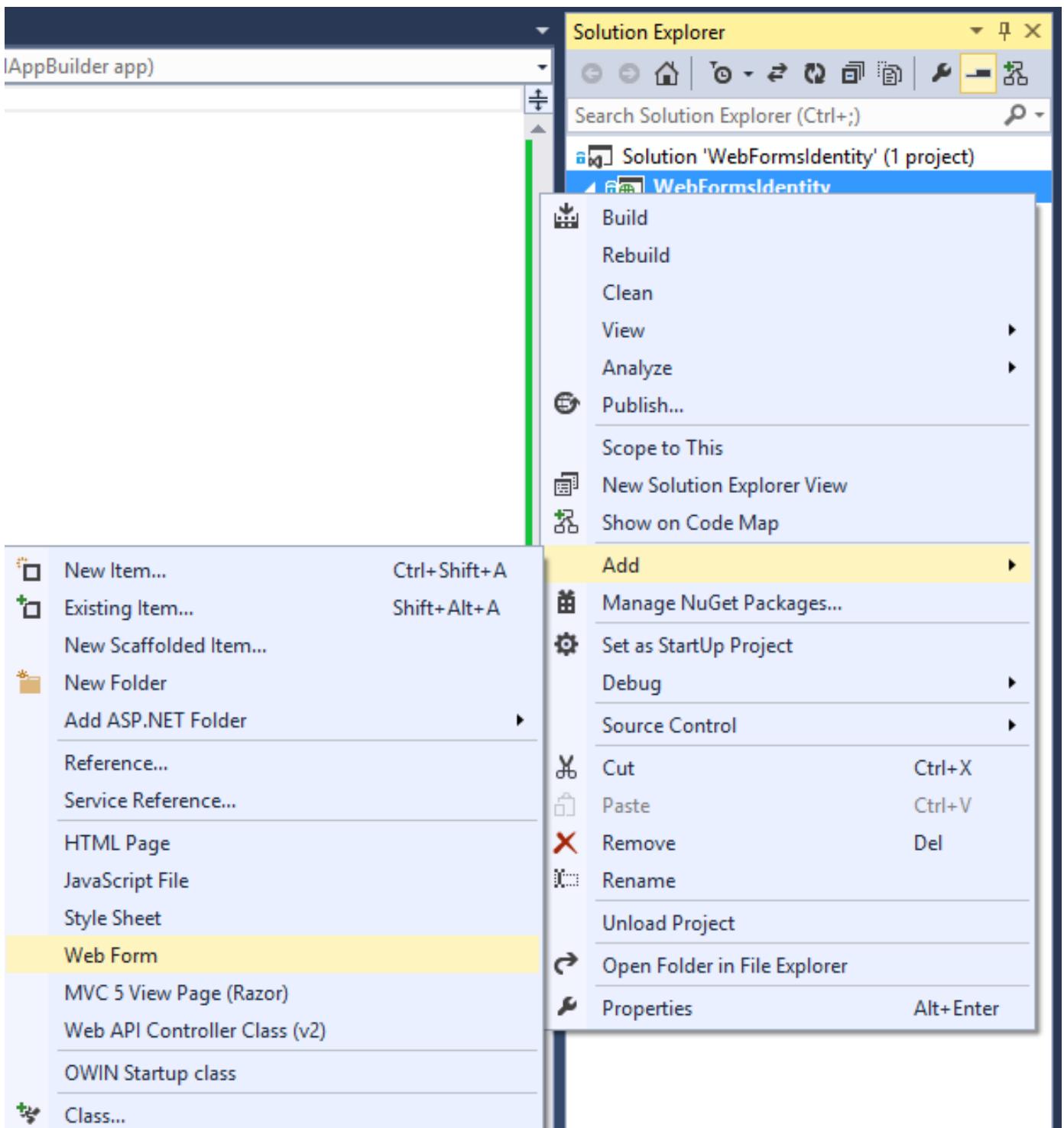
دقت کنید که دکمه **Change Authentication** غیرفعال است و هیچ پشتیبانی ای برای احراز هویت در این قالب پروژه وجود ندارد.

افزودن پکیج‌های ASP.NET Identity به پروژه
روی نام پروژه کلیک راست کنید و گزینه **Manage NuGet Packages** را انتخاب کنید. در قسمت جستجوی دیالوگ باز شده عبارت "Identity.E" را وارد کرده و این پکیج را نصب کنید.



دقت کنید که نصب کردن این پکیج وابستگی‌ها را نیز بصورت خودکار نصب می‌کند: ASP.NET Identity Core و Entity Framework.

افزودن فرم‌های وب لازم برای ثبت نام کاربران
یک فرم وب جدید بسازید.



در دیالوگ باز شده نام فرم را به **Register** تغییر داده و تایید کنید.

فایل ایجاد شده جدید را باز کرده و کد Markup آن را با قطعه کد زیر جایگزین کنید.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Register.aspx.cs"
Inherits="WebFormsIdentity.Register" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
```

```
</head>
<body style="background-color: #f0f0f0">
    <form id="form1" runat="server">
        <div>
            <h4 style="text-align: center; margin-bottom: 10px">Register a new user</h4>
            <hr />
            <p>
                <asp:Literal runat="server" ID="StatusMessage" />
            </p>
            <div style="margin-bottom: 10px">
                <asp:Label runat="server" AssociatedControlID="UserName">User name</asp:Label>
                <div>
                    <asp:TextBox runat="server" ID="UserName" />
                </div>
            </div>
            <div style="margin-bottom: 10px">
                <asp:Label runat="server" AssociatedControlID="Password">Password</asp:Label>
                <div>
                    <asp:TextBox runat="server" ID="Password" TextMode="Password" />
                </div>
            </div>
            <div style="margin-bottom: 10px">
                <asp:Label runat="server" AssociatedControlID="ConfirmPassword">Confirm password</asp:Label>
                <div>
                    <asp:TextBox runat="server" ID="ConfirmPassword" TextMode="Password" />
                </div>
            </div>
            <div>
                <asp:Button runat="server" OnClick="CreateUser_Click" Text="Register" />
            </div>
        </div>
    </form>
</body>
</html>
```

این تنها یک نسخه ساده شده Register.aspx است که از چند فیلد فرم و دکمه‌ای برای ارسال آنها به سرور استفاده می‌کند.

فایل کد این فرم را باز کرده و محتویات آن را با قطعه کد زیر جایگزین کنید.

```
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using System;
using System.Linq;

namespace WebFormsIdentity
{
    public partial class Register : System.Web.UI.Page
    {
        protected void CreateUser_Click(object sender, EventArgs e)
        {
            // Default UserStore constructor uses the default connection string named: DefaultConnection
            var userStore = new UserStore<IdentityUser>();
            var manager = new UserManager<IdentityUser>(userStore);

            var user = new IdentityUser() { UserName = UserName.Text };
            IdentityResult result = manager.Create(user, Password.Text);

            if (result.Succeeded)
            {
                StatusMessage.Text = string.Format("User {0} was created successfully!", user.UserName);
            }
            else
            {
                StatusMessage.Text = result.Errors.FirstOrDefault();
            }
        }
    }
}
```

کد این فرم نیز نسخه ای ساده شده است. فایلی که بصورت خودکار توسط VS برای شما ایجاد می‌شود متفاوت است.

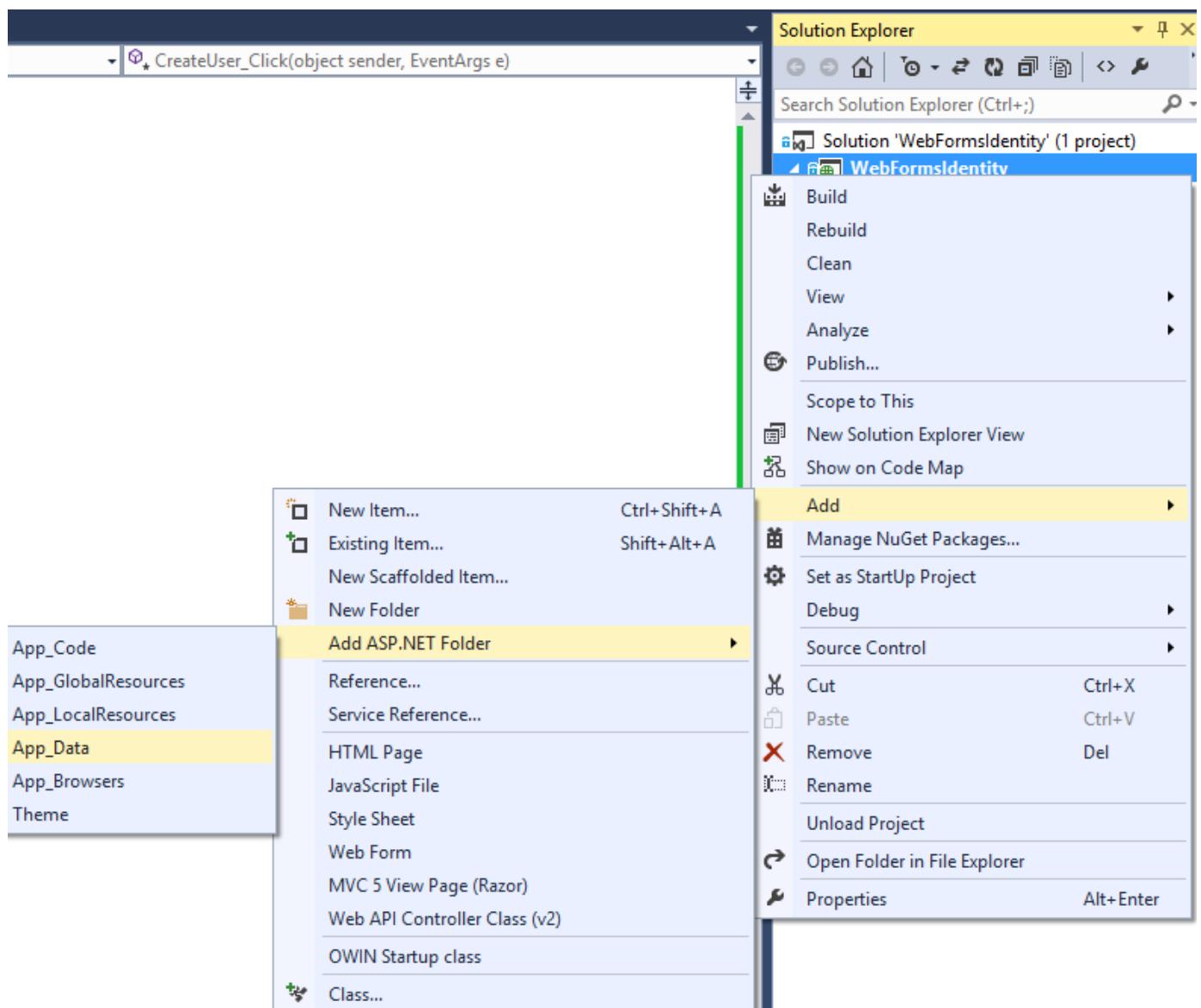
کلاس `IdentityUser` پیاده سازی پیش فرض EntityFramework از قرارداد `IUser` است. قرارداد `IUser` تعریفات حداقلی یک کاربر در ASP.NET Identity Core را در بر می‌گیرد.

کلاس `UserStore` پیاده سازی پیش فرض EF از یک فروشگاه کاربر (user store) است. این کلاس چند قرارداد اساسی `.IUserRoleStore`, `IUserStore`, `IUserLoginStore`, `IUserClaimStore` و `Identity Core` را پیاده سازی می‌کند:

کلاس `UserManager` دسترسی به API‌های مربوط به کاربران را فراهم می‌کند. این کلاس تمامی تغییرات را بصورت خودکار در `UserStore` ذخیره می‌کند.

کلاس `IdentityResult` نتیجه یک عملیات هویتی را معرفی می‌کند (identity operations).

پوشه `App_Data` را به پروژه خود اضافه کنید.



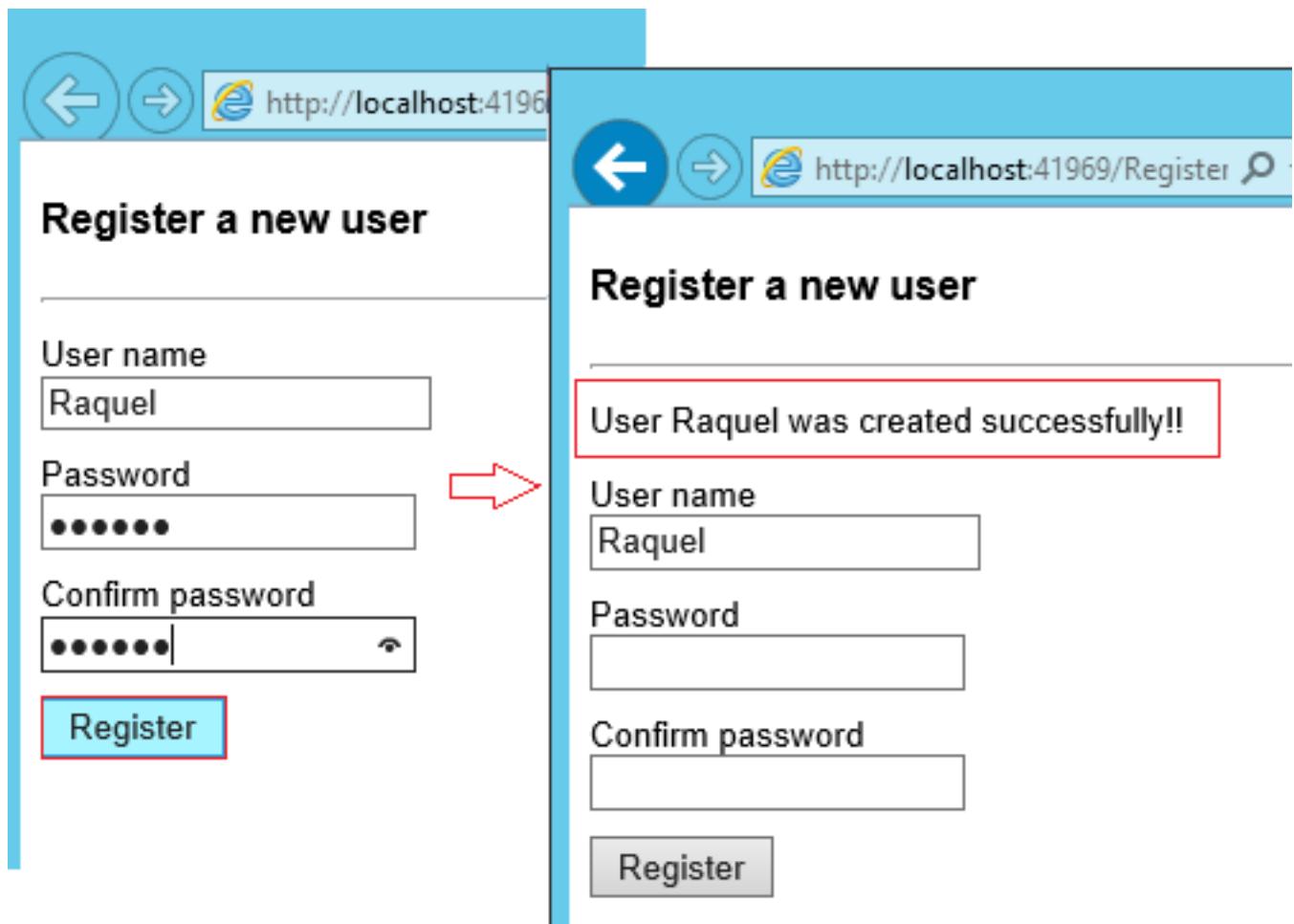
فایل `Web.config` پروژه را باز کنید و رشته اتصال جدیدی برای دیتابیس اطلاعات کاربران اضافه کنید. این دیتابیس در زمان اجرا (runtime) بصورت خودکار توسط EF ساخته می‌شود. این رشته اتصال شبیه به رشته اتصالی است که هنگام ایجاد پروژه بصورت

خودکار برای شما تنظیم می‌شود.

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit
http://go.microsoft.com/fwlink/?LinkID=237468 -->
    <section name="entityFramework"
      type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0,
      Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />
  </configSections>
  <connectionStrings>
    <add name="DefaultConnection" connectionString="Data
Source=(LocalDb)\v11.0;AttachDbFilename=|DataDirectory|\WebFormsIdentity.mdf;Initial
Catalog=WebFormsIdentity;Integrated Security=True"
      providerName="System.Data.SqlClient" />
  </connectionStrings>
  <system.web>
    <compilation debug="true" targetFramework="4.5" />
    <httpRuntime targetFramework="4.5" />
  </system.web>
  <entityFramework>
    <defaultConnectionFactory type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory,
      EntityFramework">
      <parameters>
        <parameter value="v11.0" />
      </parameters>
    </defaultConnectionFactory>
    <providers>
      <provider invariantName="System.Data.SqlClient"
        type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />
    </providers>
  </entityFramework>
</configuration>
```

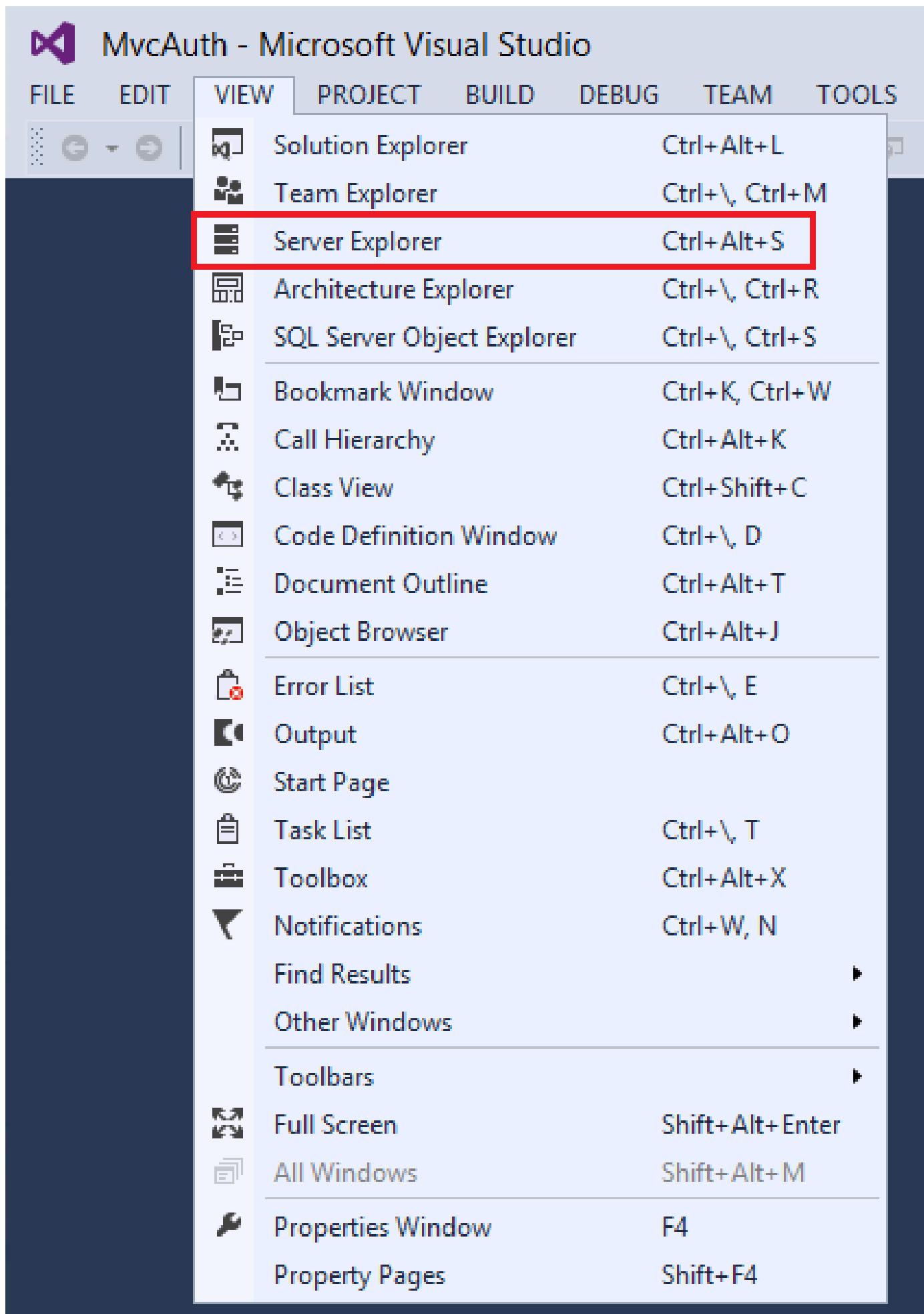
همانطور که مشاهده می‌کنید نام این رشته اتصال *DefaultConnection* است.

روی فایل Register.aspx کلیک راست کنید و گزینه **Set As Start Page** را انتخاب کنید. اپلیکیشن خود را با کلیدهای ترکیبی **Ctrl + F5** اجرا کنید که تمام پروژه را کامپایل نیز خواهد کرد. یک نام کاربری و کلمه عبور وارد کنید و روی **Register** کلیک کنید.

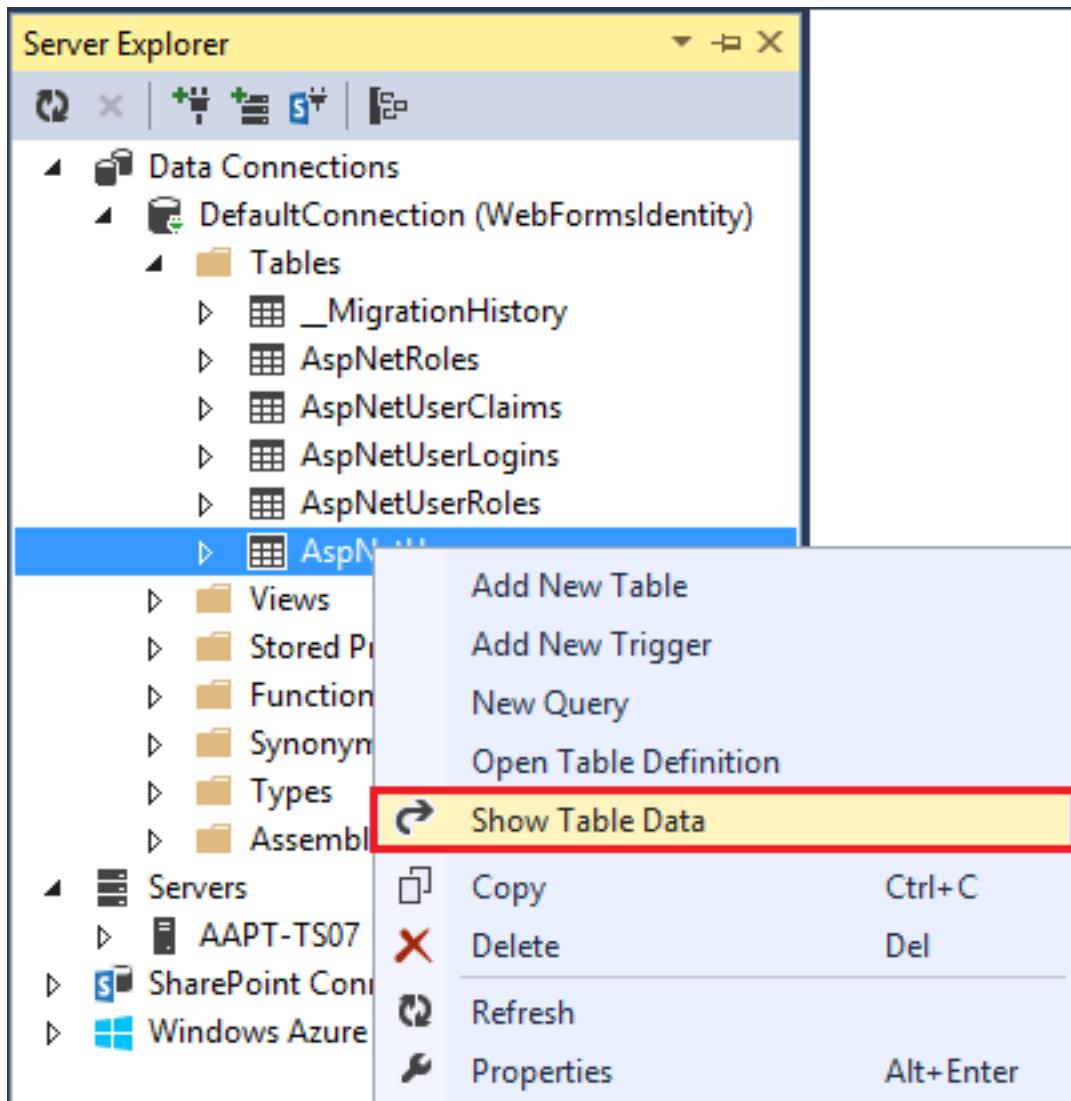


ASP.NET Identity از اعتبارسنجی نیز پشتیبانی می‌کند، مثلا در این مرحله می‌توانید از اعتبارسنج هایی که توسط Identity Core عرضه می‌شوند برای کنترل رفتار فیلد های نام کاربری و کلمه عبور استفاده کنید. اعتبارسنج پیش فرض کاربران (User) که UserValidator نام دارد خاصیتی با نام AllowOnlyAlphanumericUserNames دارد که مقدار پیش فرضش هم true است. اعتبارسنج پیش فرض کلمه عبور (MinimumLengthValidator) اطمینان حاصل می‌کند که کلمه عبور حداقل 6 کاراکتر باشد. این اعتبارسنج ها بصورت property در کلاس UserManager تعریف شده اند و می‌توانید آنها را overwrite کنید و اعتبارسنجی سفارشی خود را پیاده کنید. از آنجا که الگوی دیتابیس سیستم عضویت توسط Entity Framework مدیریت می‌شود، روی الگوی دیتابیس کنترل کامل دارید، پس از Data Annotations نیز می‌توانید استفاده کنید.

تایید دیتابیس LocalDbIdentity که توسط EF ساخته می‌شود از منوی View گزینه Server Explorer را انتخاب کنید.



گردد و **DefaultConnection (WebFormsIdentity)** را انتخاب کنید. روی جدول **AspNetUsers** کلیک راست کرده و **Show Table Data**



	Id	UserName	PasswordHash	SecurityStamp
▶	ca0-b79ad701433d	Raquel	AMmwIk9DNJf...	e12b7b1c-2aac...
*	NULL	NULL	NULL	NULL

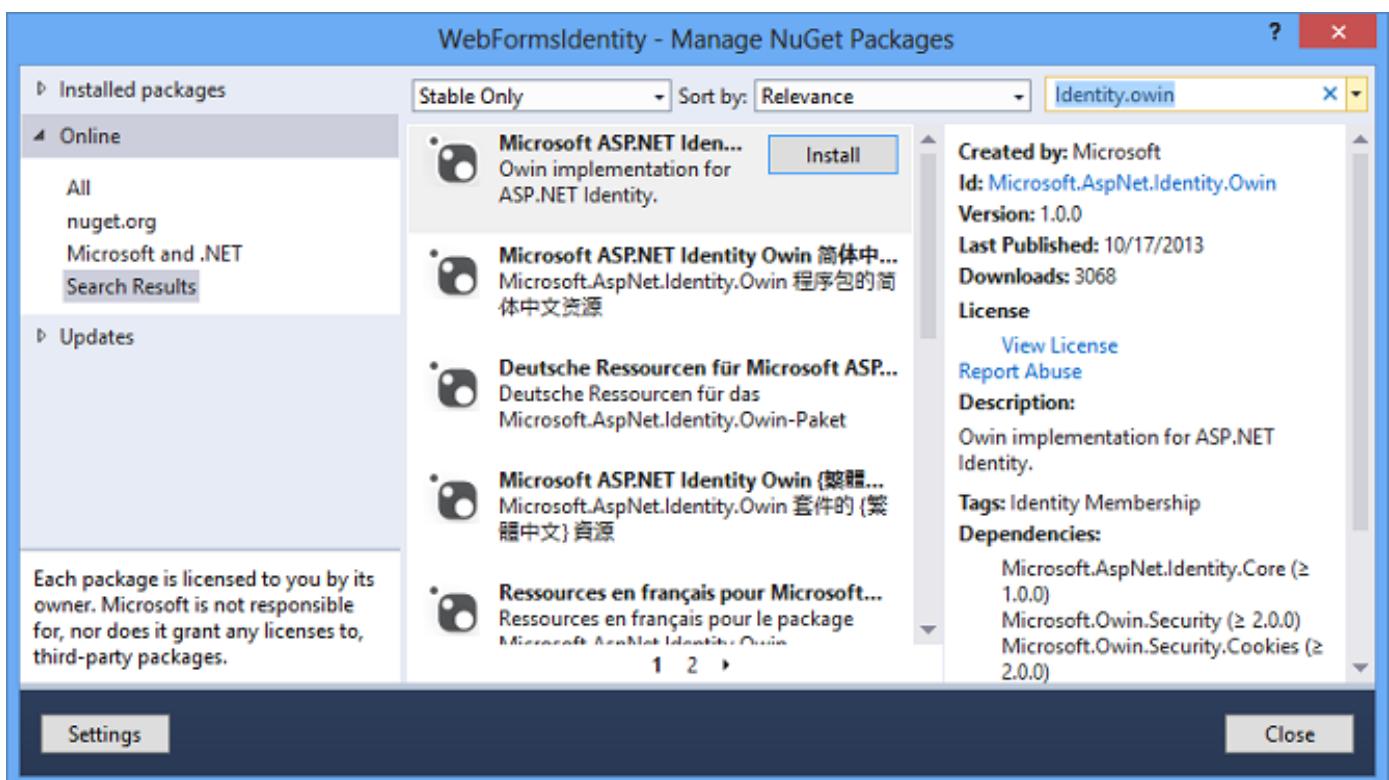
پیکربندی اپلیکیشن برای استفاده از احراز هویت OWIN

تا این مرحله ما تنها امکان ایجاد حساب‌های کاربری را فراهم کرده ایم. حال نیاز داریم امکان احراز هویت کاربران برای ورود آنها به سایت را فراهم کنیم. ASP.NET Identity برای احراز هویت مبتنی بر فرم (forms authentication) از OWIN Authentication استفاده می‌کند. مکانیزمی برای احراز هویت کاربران بر اساس claims و cookie است. این مکانیزم می‌تواند توسط Entity Framework روی OWIN یا IIS استفاده شود.

با چنین مدلی، می‌توانیم از پکیج‌های احراز هویت خود در فریم ورک‌های مختلف استفاده کنیم، مانند ASP.NET MVC و ASP.NET Web Forms. برای اطلاعات بیشتر درباره پروژه Katana و نحوه اجرای آن بصورت Host Agnostic به لینک [Getting Started with the Katana Project](#) مراجعه کنید.

نصب پکیج‌های احراز هویت روی پروژه

روی نام پروژه خود کلیک راست کرده و Manage NuGet Packages را انتخاب کنید. در قسمت جستجوی دیالوگ باز شده عبارت "Identity.Owin" را وارد کنید و این پکیج را نصب کنید.



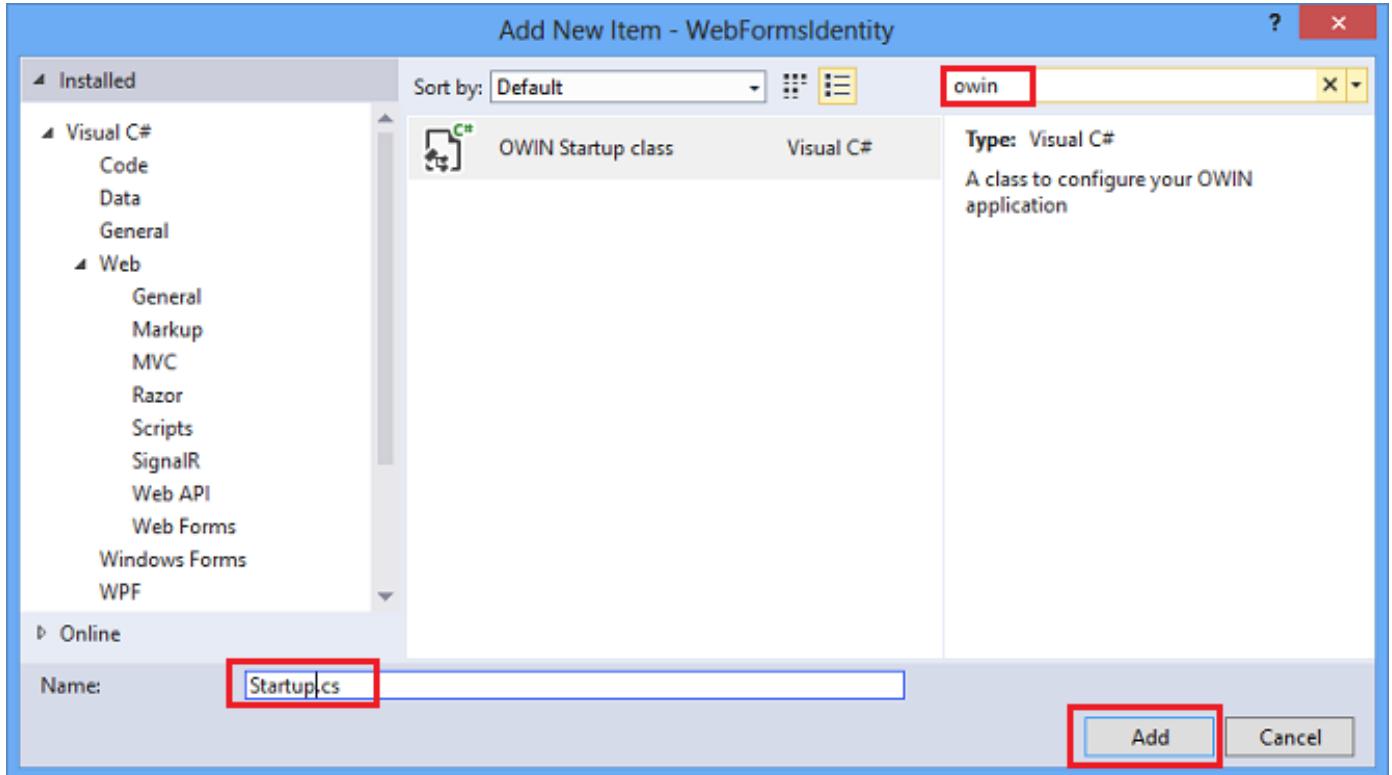
به دنبال پکیجی با نام Microsoft.Owin.Host.SystemWeb بگردید و آن را نیز نصب کنید.

پکیج Microsoft.AspNet.Identity.Owin حاوی یک سری کلاس Owin Extension است و امکان مدیریت و پیکربندی OWIN در پکیج‌های ASP.NET Identity Core Authentication می‌کند.

پکیج Microsoft.Owin.Host.SystemWeb حاوی یک سری اپلیکیشن‌های مبتنی بر OWIN است که اجرای اپلیکیشن‌های OWIN را روی IIS و ASP.NET ممکن می‌سازد. برای اطلاعات بیشتر به [OWIN Middleware in the IIS integrated Pipeline](#) مراجعه کنید.

افزودن کلاس‌های پیکربندی Authentication و Startup

روی پروژه خود کلیک راست کرده و گزینه Add New Item و سپس Add New Item را انتخاب کنید. در قسمت جستجوی دیالوگ باز شده عبارت "Startup" را وارد کنید. نام کلاس را "Startup" تعیین کرده و تایید کنید.



فایل Startup.cs را باز کنید و قطعه کد زیر را با محتویات آن جایگزین کنید تا احراز هویت OWIN Cookie Authentication پیکربندی شود.

```
using Microsoft.AspNet.Identity;
using Microsoft.Owin;
using Microsoft.Owin.Security.Cookies;
using Owin;

[assembly: OwinStartup(typeof(WebFormsIdentity.Startup))]

namespace WebFormsIdentity
{
    public class Startup
    {
        public void Configuration(IAppBuilder app)
        {
            // For more information on how to configure your application, visit
            http://go.microsoft.com/fwlink/?LinkId=316888
            app.UseCookieAuthentication(new CookieAuthenticationOptions
            {
                AuthenticationType = DefaultAuthenticationTypes.ApplicationCookie,
                LoginPath = new PathString("/Login")
            });
        }
    }
}
```

این کلاس حاوی خاصیت `OwinAttribute` است که کلاس راه انداز OWIN را نشانه گذاری می‌کند. هر اپلیکیشن OWIN یک کلاس راه انداز (`startup`) دارد که توسط آن می‌توانید کامپوننت‌های application pipeline را مشخص کنید. برای اطلاعات بیشتر درباره این مدل، به [OWIN Startup Class Detection](#) مراجعه فرمایید.

افزودن فرم‌های وب برای ثبت نام و ورود کاربران
فایل Register.cs را باز کنید و قطعه کد زیر را وارد کنید. این قسمت پس از ثبت نام موفقیت آمیز کاربر را به سایت وارد می‌کند.

```

using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.Owin.Security;
using System;
using System.Linq;
using System.Web;

namespace WebFormsIdentity
{
    public partial class Register : System.Web.UI.Page
    {
        protected void CreateUser_Click(object sender, EventArgs e)
        {
            // Default UserStore constructor uses the default connection string named: DefaultConnection
            var userStore = new UserStore<IdentityUser>();
            var manager = new UserManager<IdentityUser>(userStore);
            var user = new IdentityUser() { UserName = UserName.Text };

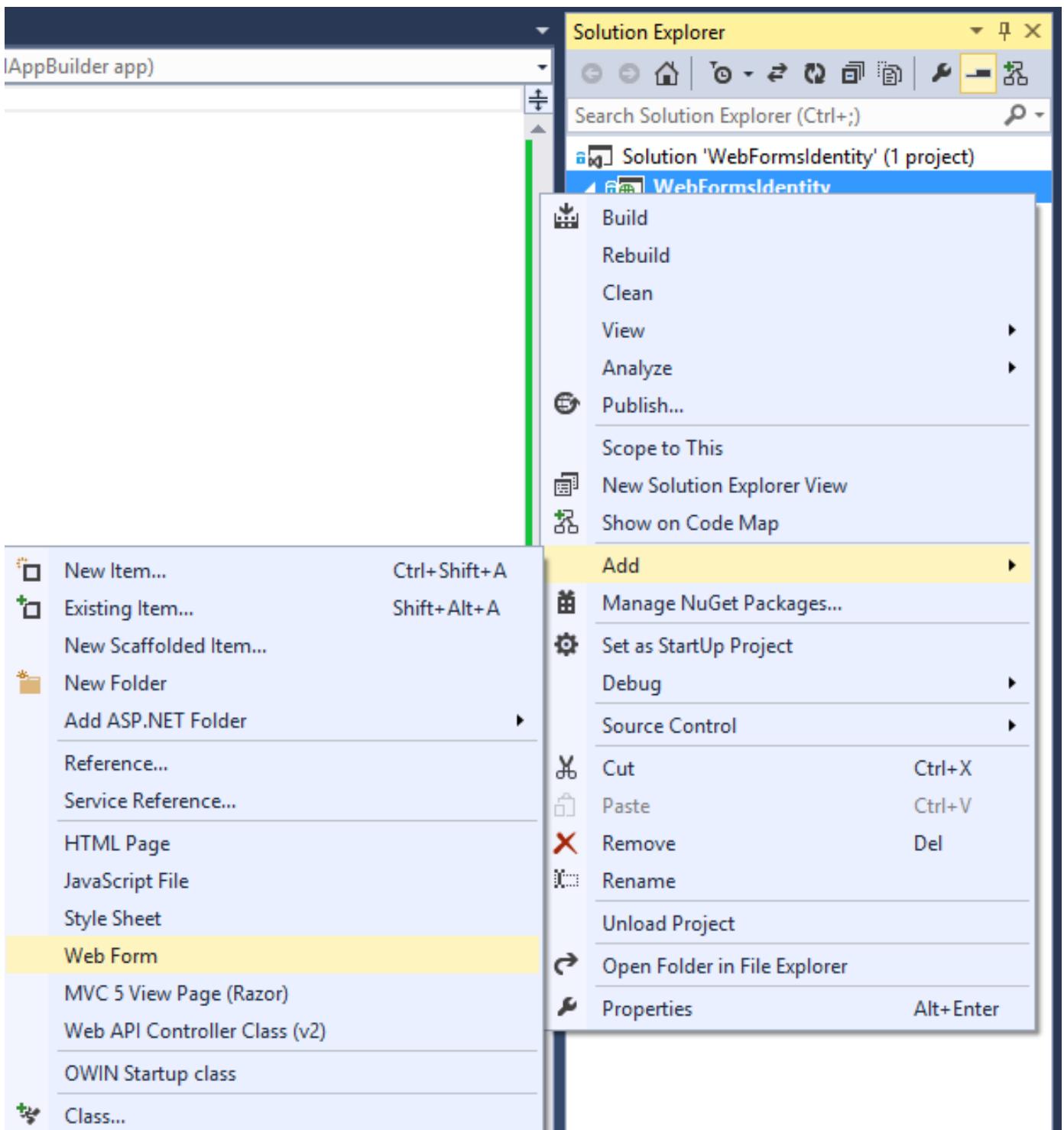
            IdentityResult result = manager.Create(user, Password.Text);

            if (result.Succeeded)
            {
                var authenticationManager = HttpContext.Current.GetOwinContext().Authentication;
                var userIdentity = manager.CreateIdentity(user,
DefaultAuthenticationTypes.ApplicationCookie);
                authenticationManager.SignIn(new AuthenticationProperties() { }, userIdentity);
                Response.Redirect("~/Login.aspx");
            }
            else
            {
                StatusMessage.Text = result.Errors.FirstOrDefault();
            }
        }
    }
}

```

از آنجا که OWIN Cookie Authentication و ASP.NET Identity هستند، فریم ورک از برنامه نویس اپلیکیشن انتظار دارد تا برای کاربر یک آبجکت از نوع [ClaimsIdentity](#) تولید کند. این آبجکت تمام اطلاعات اختیارات کاربر را در بر می‌گیرد، مثلاً اینکه کاربر به چه نقش‌هایی تعلق دارد. همچنین در این مرحله می‌توانید اختیارات (Claims) جدیدی به کاربر اضافه کنید.

شما با استفاده از `AuthenticationManager` که متعلق به OWIN است می‌توانید کاربر را به سایت وارد کنید. برای این کار شما متد `SignIn` را فراخوانی می‌کنید و آبجکتی از نوع `ClaimsIdentity` را به آن پاس می‌دهید. این کد کاربر را به سایت وارد می‌کند و یک کوکی برای او می‌سازد. این فراخوانی معادل همان [FormAuthentication.SetAuthCookie](#) است که توسط ماثول استفاده می‌شود. [FormsAuthentication](#) روی پروژه خود کلیک راست کرده، فرم وب جدیدی با نام `Login` بسازید.



فایل `Login.aspx` را باز کنید و کد زیر تغییر دهید.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Login.aspx.cs"
Inherits="WebFormsIdentity.Login" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body style="font-family: Arial, Helvetica, sans-serif; font-size: small">
    <form id="form1" runat="server">
        <div>
```

```

<h4 style="font-size: medium">Log In</h4>
<hr />
<asp:PlaceHolder runat="server" ID="LoginStatus" Visible="false">
    <p>
        <asp:Literal runat="server" ID="StatusText" />
    </p>
</asp:PlaceHolder>
<asp:PlaceHolder runat="server" ID="LoginForm" Visible="false">
    <div style="margin-bottom: 10px">
        <asp:Label runat="server" AssociatedControlID="UserName">User name</asp:Label>
        <div>
            <asp:TextBox runat="server" ID="UserName" />
        </div>
    </div>
    <div style="margin-bottom: 10px">
        <asp:Label runat="server" AssociatedControlID="Password">Password</asp:Label>
        <div>
            <asp:TextBox runat="server" ID="Password" TextMode="Password" />
        </div>
    </div>
    <div style="margin-bottom: 10px">
        <div>
            <asp:Button runat="server" OnClick="SignIn" Text="Log in" />
        </div>
    </div>
</asp:PlaceHolder>
<asp:PlaceHolder runat="server" ID="LogoutButton" Visible="false">
    <div>
        <div>
            <asp:Button runat="server" OnClick="SignOut" Text="Log out" />
        </div>
    </div>
</asp:PlaceHolder>
</div>
</form>
</body>
</html>

```

محتوای فایل *Login.aspx.cs* را نیز مانند لیست زیر تغییر دهید.

```

using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.Owin.Security;
using System;
using System.Web;
using System.Web.UI.WebControls;

namespace WebFormsIdentity
{
    public partial class Login : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                if (User.Identity.IsAuthenticated)
                {
                    StatusText.Text = string.Format("Hello {0}!", User.Identity.GetUserName());
                    LoginStatus.Visible = true;
                    LogoutButton.Visible = true;
                }
                else
                {
                    LoginForm.Visible = true;
                }
            }
        }

        protected void SignIn(object sender, EventArgs e)
        {
            var userStore = new UserStore<IdentityUser>();
            var userManager = new UserManager<IdentityUser>(userStore);
            var user = userManager.Find(UserName.Text, Password.Text);

            if (user != null)
            {
                var authenticationManager = HttpContext.Current.GetOwinContext().Authentication;
                var userIdentity = userManager.CreateIdentity(user,

```

```
DefaultAuthenticationTypes.ApplicationCookie);

        authenticationManager.SignIn(new AuthenticationProperties() { IsPersistent = false },
userIdentity);
        Response.Redirect("~/Login.aspx");
    }
else
{
    StatusText.Text = "Invalid username or password.";
    LoginStatus.Visible = true;
}
}

protected void SignOut(object sender, EventArgs e)
{
    var authenticationManager = HttpContext.Current.GetOwinContext().Authentication;
    authenticationManager.SignOut();
    Response.Redirect("~/Login.aspx");
}
}
}
```

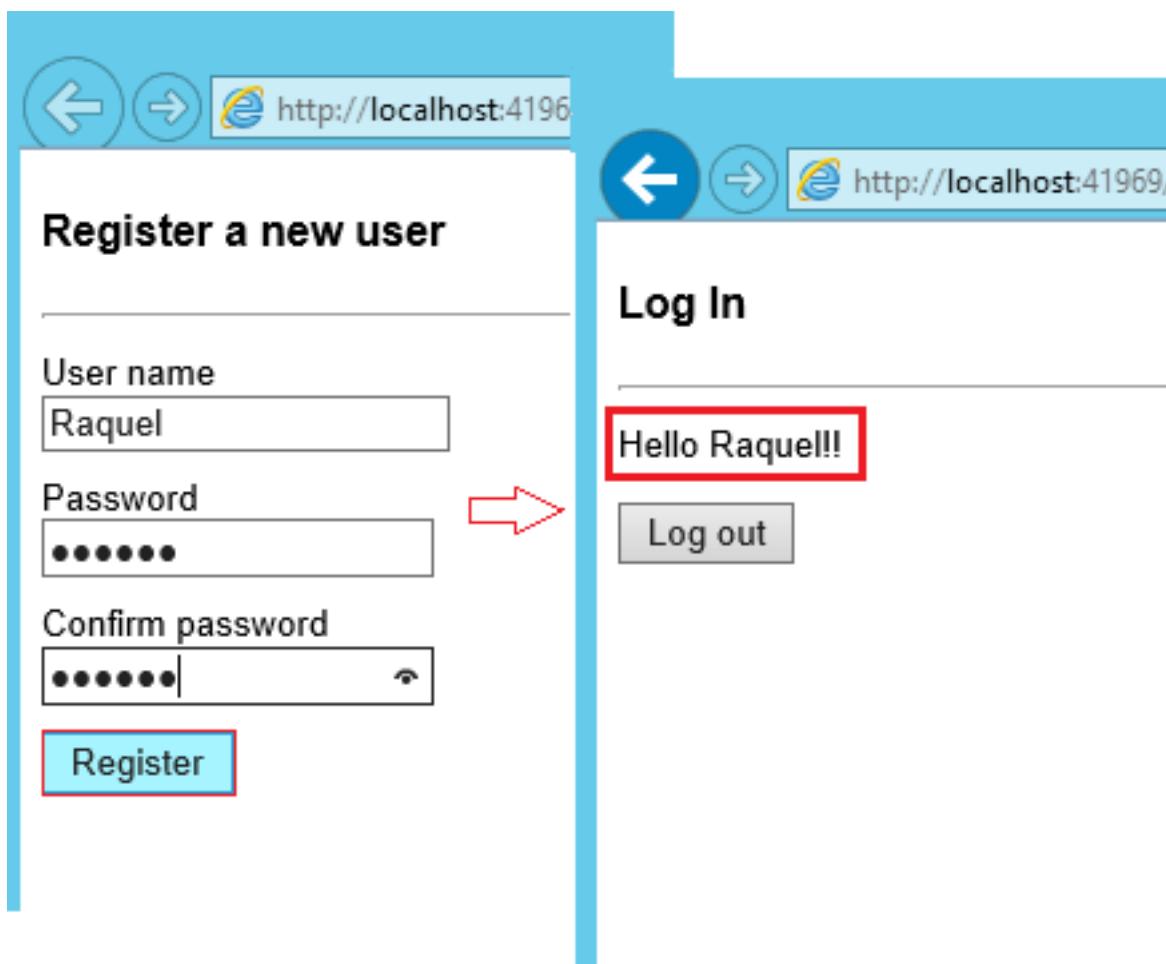
متدهای `Page_Load` حالا وضعیت کاربر جاری را بررسی می‌کند و بر اساس وضعیت `Context.User.Identity.IsAuthenticated` تصمیم گیری می‌کند.

نمایش نام کاربر جاری: فریم ورک `ASP.NET Identity` روی [System.Security.Principal.Identity](#) متدهایی نوشته است که به شما امکان دریافت نام و شناسه کاربر جاری را می‌دهد. این متدها در اسمبلی `Microsoft.AspNet.Identity.Core` وجود دارند. این متدها جایگزین [HttpContext.User.Identity.Name](#) هستند.

متدهای `SignIn`

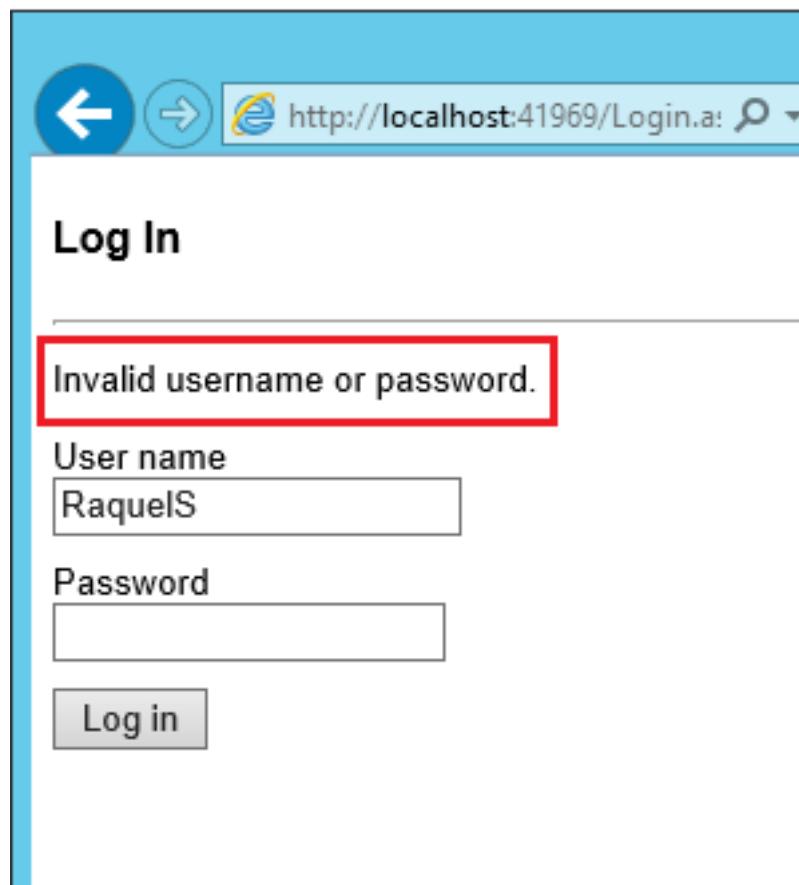
این متدهای `CreateUser_Click` را که پیشتر بصورت خودکار ایجاد شده جایگزین می‌کند و پس از ایجاد موفقیت آمیز حساب کاربری، کاربر جاری را به سایت وارد می‌کند. فریم ورک `OWIN` متدهایی روی `System.Web.HttpContext` افزوده است که به شما این امکان را می‌دهند که یک ارجاع از نوع `IWinContext` بگیرید. این متدها در اسمبلی `Microsoft.Owin.Host.SystemWeb` وجود دارند. کلاس `OwinContext` خاصیتی از نوع `IAuthenticationManager` دارد که امکانات احراز هویت موجود برای درخواست جاری را معرفی می‌کند.

پروژه را با `Ctrl + F5` اجرا کنید و کاربر جدیدی بسازید. پس از وارد کردن نام کاربری و کلمه عبور و کلیک کردن دکمه `Register` باید بصورت خودکار به سایت وارد شوید و نام خود را مشاهده کنید.



همانطور که مشاهده می‌کنید در این مرحله حساب کاربری جدید ایجاد شده و به سایت وارد شده است. روی Log out کلیک کنید تا از سایت خارج شوید. پس از آن باید به صفحه ورود هدایت شوید. حالا یک نام کاربری یا کلمه عبور نامعتبر وارد کنید و روی Log in کلیک کنید.

متد UserManager.Find مقدار null بر می‌گرداند، بنابراین پیام خطای "Invalid username or password" نمایش داده خواهد شد.



نظرات خوانندگان

نویسنده: Programmer | تاریخ: ۱۴:۴۱ ۱۳۹۲/۱۰/۱۹

با عرض سلام و تشکر بابت ترجمه روانتون. خیلی وقت بود که منتظر همچین پستی بودم. اینکه بشه با EF عملیات احراز هویت رو با مکانیسمی قویتر از Membership انجام داد. اگر ممکنه همین مثال رو در قالب پروژه MVC انجام بدید. ممنون

نویسنده: آرمین ضیاء | تاریخ: ۱۸:۹ ۱۳۹۲/۱۰/۱۹

سلام، متشرکم.

ASP.NET Identity بصورت پیش فرض در قالب پروژه های VS 2013 استفاده میشه. در پست های قبلی بیشتر درباره این فریم بحث شده که می تونید مراجعه کنید. برای اطلاعات بیشتر به [ASP.NET Identity](#) سر بزنید.

نویسنده: کامران | تاریخ: ۱۶:۰ ۱۳۹۲/۱۲/۲۶

سلام.

من اگر بخواهم لیست کاربرایی که یک نقش خاص مثل "Admin" رو دارن لیست کنم چطور میتونم دسترسی داشته باشم؟

نویسنده: غلامرضا | تاریخ: ۳:۴۹ ۱۳۹۳/۰۵/۳۰

سلام و میبخشید آیا به غیر از روش ایداعی [این سایت](#) روش دیگه ای برای استفاده از identity با db first وجود دارد؟ اگه هست لطفا راهنمایی کنید.

نویسنده: غلامرضا | تاریخ: ۱۶:۳۲ ۱۳۹۳/۰۵/۳۱

جواب رو پیدا کردم اینم لینکش. [فیلم](#)

در کنار ASP.NET Web API 2 بهمراه یک سری قابلیت جدید جالب منتشر شده است. در این پست ۵ قابلیت برتر از این قابلیت‌های جدید را بررسی می‌کنیم.

Attribute Routing .1

در کنار سیستم routing فعلی، ASP.NET Web API 2 هم پشتیبانی می‌کند. در مورد سیستم Attribute Routing حالا از می‌توانیم قالب‌های متعددی برای routing بنویسیم. هنگامی که یک درخواست به سرور میرسد، کنترلر مناسب انتخاب شده و اکشن متناسب فراخوانی می‌شود.

در لیست زیر قالب پیشفرض routing در Web API را مشاهده می‌کنید.

```
Config.Routes.MapHttpRoute(
    name: "DefaultApi",
    routeTemplate: "api/{Controller}/{id}",
    defaults: new { id = RouteParameter.Optional }
);
```

این رویکرد routing مزایای خود را دارد. از جمله اینکه تمام مسیرها در یک مکان واحد تعریف می‌شوند، اما تنها برای الگوهای مشخص. مثلاً پشتیبانی از nested routing روی یک کنترلر مشکل می‌شود.

در ASP.NET Web API 2 به سادگی می‌توانیم الگوی URI ذکرد شده را پشتیبانی کنیم. لیست زیر نمونه‌ای از یک الگوی URI با AttributeRouting را نشان می‌دهد.

URI Pattern --> books/1/authors

```
[Route("books/{bookId}/authors")]
public IEnumerable<Author> GetAuthorByBook(int bookId) { ..... }
```

CORS - Cross Origin Resource Sharing .2

بصورت نرمال، مرورگرها اجازه درخواست‌های cross-domain را نمی‌دهند، که بخارط same-origin policy است. خوب، CORS (Cross Origin Resource Sharing) چیست؟

CORS یک مکانیزم است که به صفحات وب این را اجازه می‌دهد تا یک درخواست آزادکسی (Ajax Request) به دامنه‌ای دیگر ارسال کنند. دامنه‌ای به غیر از دامنه‌ای که صفحه وب را رندر کرده است. CORS با استانداردهای W3C سازگار است و حالا در نسخه 2 خود از آن پشتیبانی می‌کند.

OWIN (Open Web Interface for .NET) self-hosting .3

ASP.NET Web API 2 بهمراه یک پکیج عرضه می‌شود، که Microsoft.AspNet.WebApi.OwinSelfHost نام دارد. طبق گفته وب سایت :

یک اینترفیس استاندارد بین سرورهای دات نت و اپلیکیشن‌های وب تعریف می‌کند. هدف این اینترفیس جداسازی (decoupling) سرور و اپلیکیشن است. تشویق به توسعه ماژول‌های ساده برای توسعه اپلیکیشن‌های وب دات نت. و بعنوان یک استاندارد باز (open standard) اکوسیستم نرم افزارهای متن باز را تحریک کند تا ابزار توسعه اپلیکیشن‌های وب دات نت توسعه یابند.

بنابراین طبق گفته‌های بالا، OWIN گزینه‌ای ایده آل برای میزبانی اپلیکیشن‌های وب روی پروسس‌هایی به غیر از پروسس IIS است. پیاده سازی‌های دیگری از OWIN نیز وجود دارند، مانند Katana، Giacomo، Kayak، Firefly و غیره. اما Katana گزینه توصیه شده برای سرورهای مایکروسافت و فریم ورک‌های Web API است.

IHttpActionResult .4

در کنار دو روش موجود فعلی برای ساختن response اکشن متد‌ها در کنترلرها، ASP.NET Web API 2 حالا از مدل جدیدی هم

پشتیبانی می‌کند. *IHttpResponseMessage* یک اینترفیس است که بعنوان یک فاکتوری (factory) برای *HttpServletResponse* کار می‌کند. این روش بسیار قدرتمند است بدلیل اینکه web api را گسترش می‌دهد. با استفاده از این رویکرد، می‌توانیم هایی با هر نوع دلخواه بسازیم. برای اطلاعات بیشتر به [how to serve HTML with IHTTPActionResult](#) مراجعه کنید.

Web API OData .5

پروتکل OData (Open Data Protocol) در واقع یک پروتکل وب برای کوئری گرفتن و بروز رسانی داده‌ها است. 2 پشتیبانی از \$value و \$expand، \$select را اضافه کرده است. با استفاده از این امکانات، می‌توانیم نحوه معرفی پاسخ سرور را کنترل کنیم، یعنی representation دریافتی از سرور را می‌توانید سفارشی کنید.
\$expand: بصورت نرم‌ال، هنگام کوئری گرفتن از یک کالکشن OData، پاسخ سرور موجودیت‌های مرتبط (related entities) را شامل نمی‌شود. با استفاده از \$expand می‌توانیم موجودیت‌های مرتبط را بصورت inline در پاسخ سرور دریافت کنیم.
\$select: از این متدهای انتخاب چند خاصیت بخصوص از پاسخ سرور استفاده می‌شود، بجای آنکه تمام خاصیت‌ها بارگذاری شوند.
\$value: با این متدهای انتخاب مقدار خام (raw) فیلدها را بدست می‌آورید، بجای دریافت آنها در فرمت OData.

چند مقاله خوب دیگر

[Top 10 ASP.NET MVC Interview Questions](#)

[jQuery code snippets every web developer must have](#) 7

[WCF Vs ASMX Web Services](#)

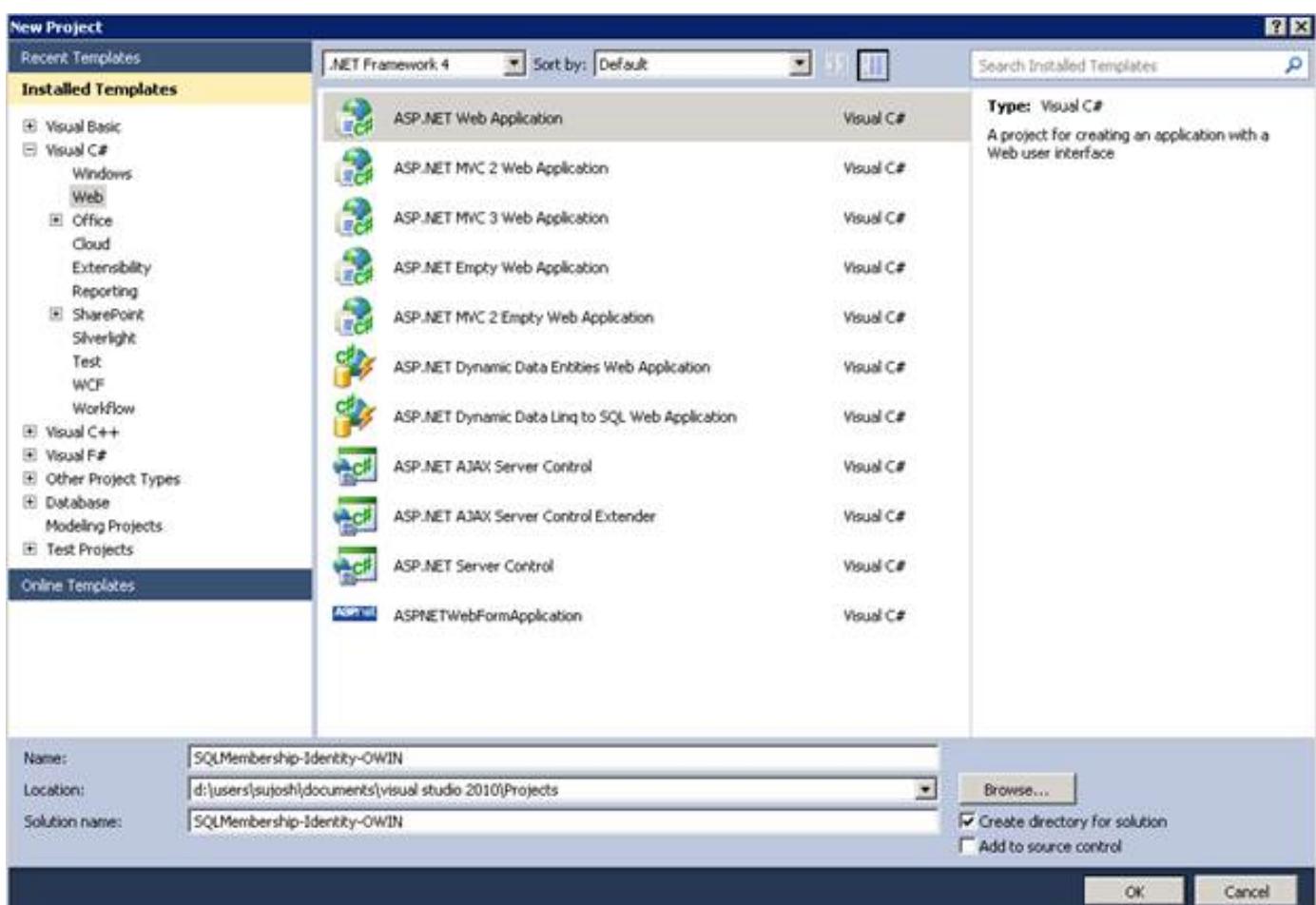
[Top 10 HTML5 Interview Questions](#)

[JavaScript : Validating Letters and Numbers](#)

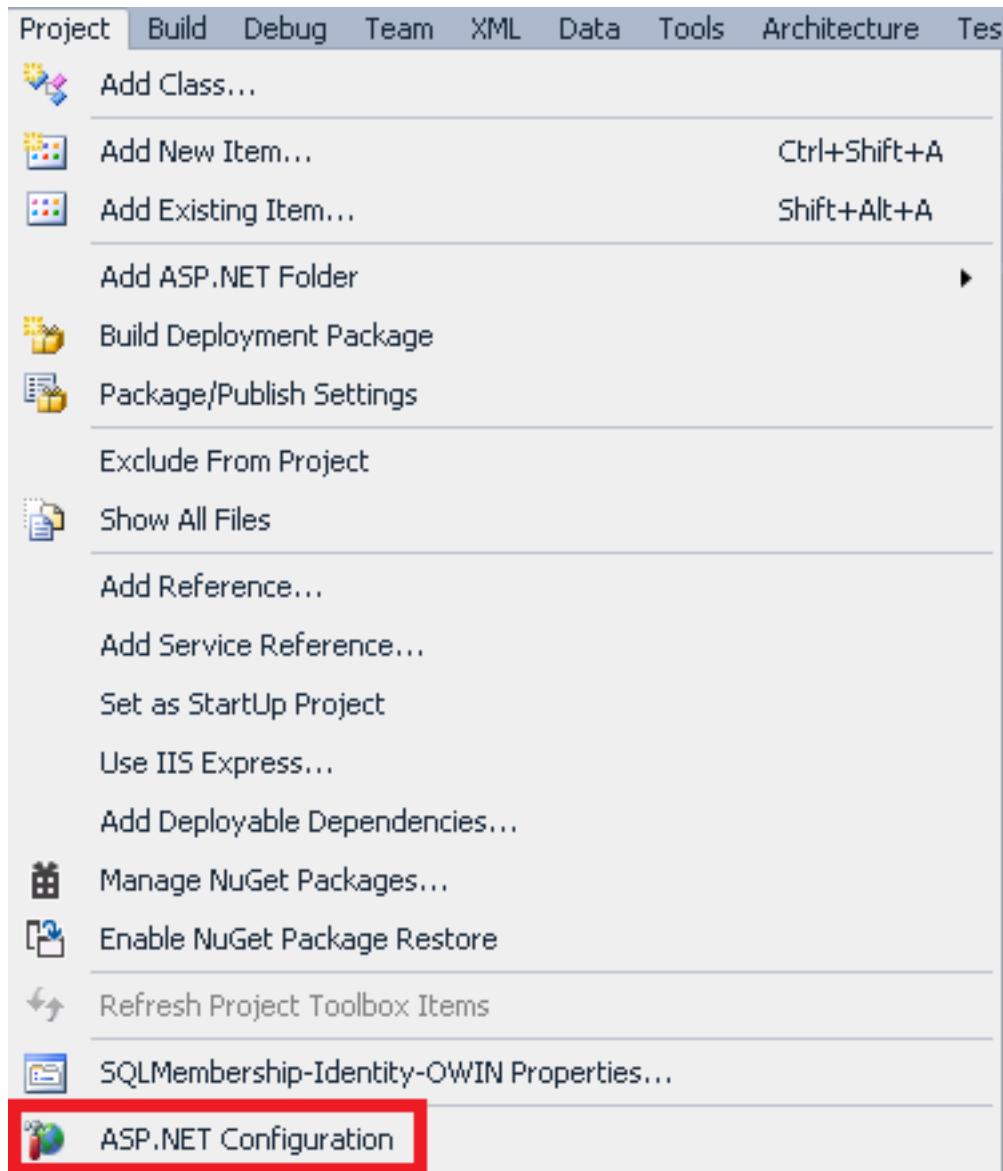
در این مقاله مهاجرت یک اپلیکیشن وب که توسط SQL Membership ساخته شده است را به سیستم جدید بررسی می‌کنیم. برای این مقاله از یک قالب اپلیکیشن وب (Web Forms) که توسط Visual Studio 2010 ساخته شده است برای ساختن کاربران و نقش‌ها استفاده می‌کنیم. سپس با استفاده از یک SQL Script دیتابیس موجود را به دیتابیسی که ASP.NET نیاز دارد تبدیل می‌کنیم. در قدم بعدی پکیج‌های مورد نیاز را به پروژه اضافه می‌کنیم و صفحات جدیدی برای مدیریت حساب‌های کاربری خواهیم ساخت. بنویان یک تست، کاربران قدیمی که توسط SQL Membership ساخته شده بودند باید قادر باشند به سایت وارد شوند. همچنین کاربران جدید باید بتوانند بدون هیچ مشکلی در سیستم ثبت نام کنند. سورس کد کامل این مقاله را می‌توانید از [این لینک](#) دریافت کنید.

یک اپلیکیشن با SQL Membership بسازید

برای شروع به اپلیکیشنی نیاز داریم که از SQL Membership استفاده می‌کند و دارای داده‌هایی از کاربران و نقش‌ها است. برای این مقاله، بگذارید پروژه جدیدی توسط VS 2010 بسازیم.



حال با استفاده از ابزار ASP.NET Configuration .oldUser و .oldAdminUser دو کاربر جدید بسازید:



Home

Security

Application

Provider

You can use the Web Site Administration Tool to manage all the security (rules for controlling access to parts of your application).

By default, user information is stored in a Microsoft SQL Server Express different provider.

[Use the security Setup Wizard to configure security step by step.](#)

Click the links in the table to manage the settings for your application.

Users

Existing users: 2

[Create user](#)

[Manage users](#)

[Select authentication type](#)

نقش جدیدی با نام Admin بسازید و کاربر oldAdminUser را به آن اضافه کنید.

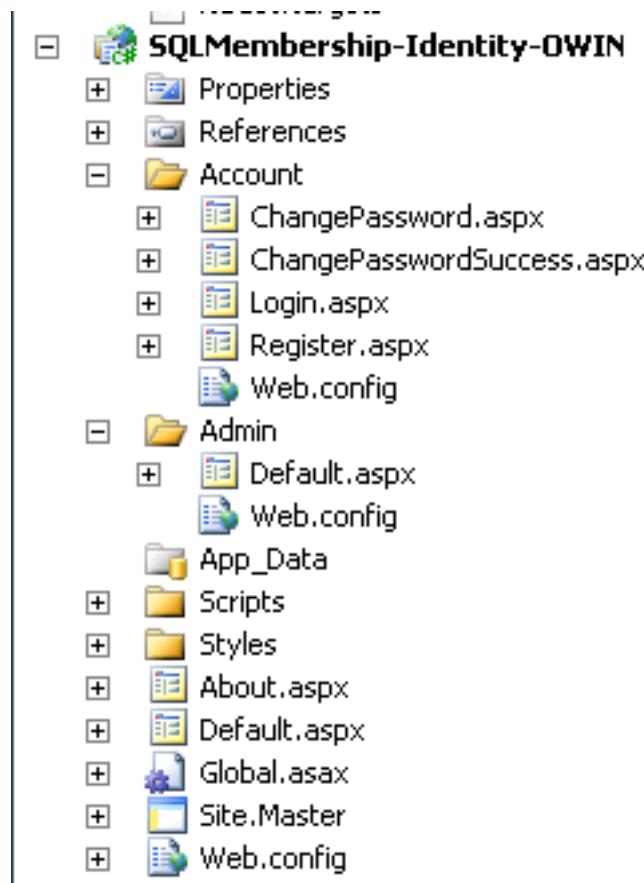
Roles

Existing roles: 1

[Disable Roles](#)

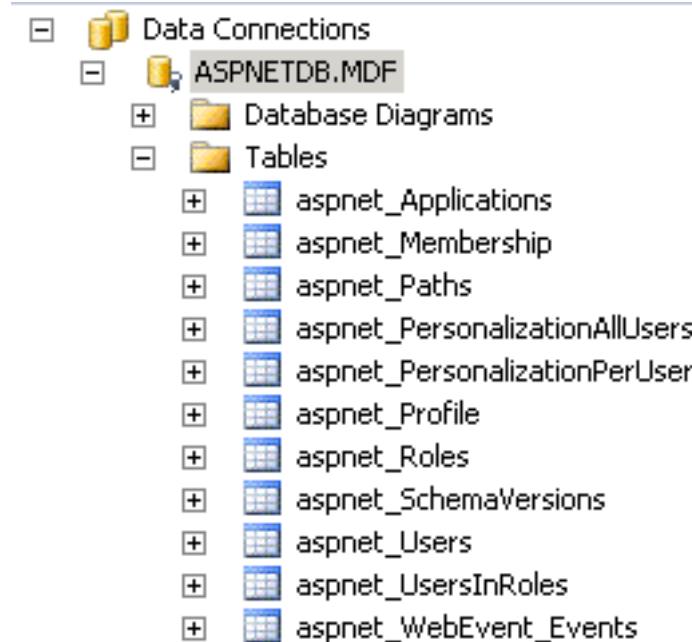
[Create or Manage roles](#)

بخش جدیدی با نام Admin در سایت خود بسازید و فرمی بنام Default.aspx به آن اضافه کنید. همچنین فایل web.config این قسمت را طوری پیکربندی کنید تا تنها کاربرانی که در نقش Admin هستند به آن دسترسی داشته باشند. برای اطلاعات بیشتر به [این لینک](#) مراجعه کنید.



پنجره Server Explorer را باز کنید و جداول ساخته شده توسط SQL Membership را بررسی کنید. اطلاعات اصلی کاربران که برای ورود به سایت استفاده می‌شوند، در جداول **aspnet_Membership** و **aspnet_Users** ذخیره می‌شوند. داده‌های مربوط به نقش‌ها نیز در جدول **aspnet_Roles** ذخیره خواهند شد. رابطه بین کاربران و نقش‌ها نیز در جدول **aspnet_UsersInRoles** ذخیره می‌شود، یعنی اینکه هر کاربری به چه نقش‌هایی تعلق دارد.

برای مدیریت اساسی سیستم عضویت، مهاجرت جداول ذکر شده به سیستم جدید ASP.NET Identity کفایت می‌کند.



مهاجرت به Visual Studio 2013

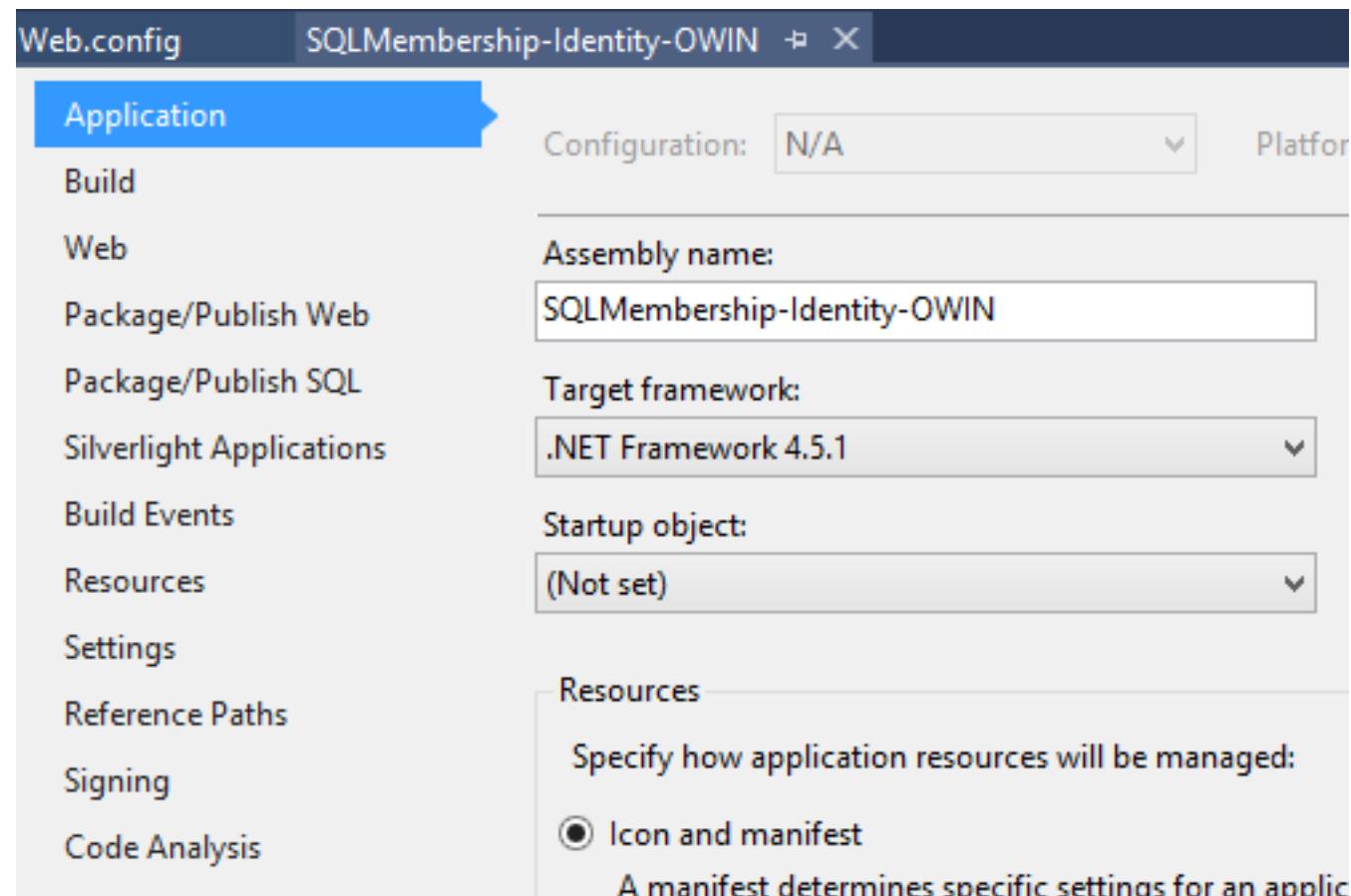
برای شروع ابتدا Visual Studio 2013 for Web یا Visual Studio Express 2013 را نصب کنید.

حال پروژه ایجاد شده را در نسخه جدید ویژوال استودیو باز کنید. اگر نسخه ای از SQL Server Express را روی سیستم خود نصب نکرده باشید، هنگام باز کردن پروژه پیغامی به شما نشان داده می‌شود. دلیل آن وجود رشته اتصالی است که از SQL Server Express استفاده می‌کند. برای رفع این مساله می‌توانید SQL Express را نصب کنید، و یا رشته اتصال را طوری تغییر دهید که از LocalDB استفاده کند.

فایل web.config را باز کرده و رشته اتصال را مانند تصویر زیر ویرایش کنید.

```
<configuration>
  <connectionStrings>
    <add name="ApplicationServices"
      connectionString="data source=(LocalDb)\v11.0;Integrated Security=SSPI;AttachDBFilename=|DataDirectory|\aspnetdb.mdf"
      providerName="System.Data.SqlClient" />
  </connectionStrings>
```

پنجره Server Explorer را باز کنید و مطمئن شوید که الگوی جداول و داده‌ها قابل رویت هستند. سیستم ASP.NET Identity با نسخه 4.5 دات نت فریم ورک و بالاتر سازگار است. پس نسخه فریم ورک پروژه را به آخرین نسخه (4.5.1) تغییر دهید.



پروژه را Build کنید تا مطمئن شوید هیچ خطای وجود ندارد.

NuGet پکیج‌های

در پنجره Solution Explorer روی نام پروژه خود کلیک راست کرده، و گزینه Manage NuGet Packages را انتخاب کنید. در قسمت جستجوی دیالوگ باز شده، عبارت "Microsoft.AspNet.Identity.EntityFramework" را وارد کنید. این پکیج را در لیست نتایج انتخاب کرده و آن را نصب کنید. نصب این بسته، نیازمندی‌های موجود را بصورت خودکار دانلود و نصب می‌کند: ASP.NET Identity Core و EntityFramework . حال پکیج‌های زیر را هم نصب کنید (اگر نمی‌خواهید OAuth را فعال کنید، 4 پکیج آخر را نادیده بگیرید).

Microsoft.AspNet.Identity.Owin

Microsoft.Owin.Host.SystemWeb

Microsoft.Owin.Security.Facebook

Microsoft.Owin.Security.Google

Microsoft.Owin.Security.MicrosoftAccount

Microsoft.Owin.Security.Twitter

SQLMembership-Identity-OWIN - Manage NuGet Packages

Include Prerelease Sort by: Relevance

asp.net identity

DotNetOpenAuth extensions for ASP.NET (WebPages)
Allow your web visitors to log into your web site using accounts they already have with popular services.

Thinktecture.IdentityModel
Helper library for identity & access control in .NET 4.0/WIF and .NET 4.5 (includes MVC4 and Web API support).

Microsoft ASP.NET Identity EntityFrame...
ASP.NET Identity providers that use Entity Framework. **Install**

Microsoft ASP.NET Identity Core
Core interfaces for ASP.NET Identity.

Microsoft ASP.NET Razor
This package contains the runtime assemblies for ASP.NET Web Pages.

Microsoft ASP.NET Web Pages
This package contains core runtime assemblies shared between ASP.NET MVC and ASP.NET Web Pages.

DotNetOpenAuth OpenID InfoCard Relying Party ASP.NET...
Allow your web site's visitors to log in using InfoCards.

Created by: Microsoft
Id: Microsoft.AspNet.Identity.EntityFramework
Version: 1.0.0
Last Published: 10/17/2013
Downloads: 6966
License
View License
Report Abuse
Description:
ASP.NET Identity providers that use Entity Framework.
Tags: Identity Membership
Dependencies:
Microsoft.AspNet.Identity.Core (> 1.0.0)
EntityFramework (> 6.0.0)
Each item above may have sub-dependencies subject to additional license agreements.

1 2 3 4 5 >

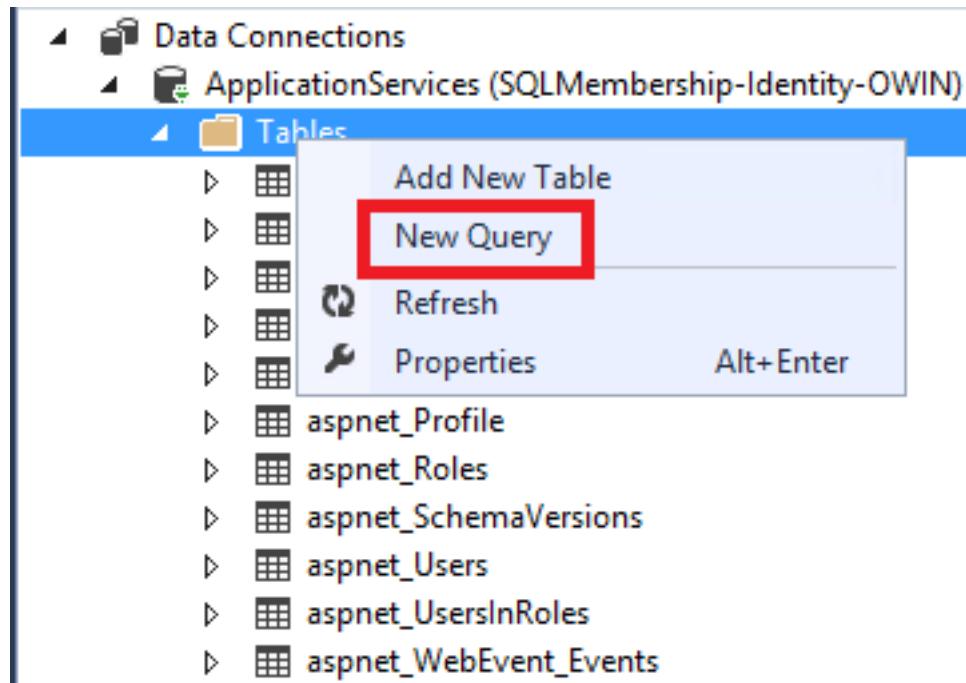
Close

مهاجرت دیتابیس فعلی به سیستم ASP.NET Identity

قدم بعدی مهاجرت دیتابیس فعلی به الگویی است، که سیستم ASP.NET Identity به آن نیاز دارد. بدین منظور ما یک اسکریپت SQL را اجرا می‌کنیم تا جداول جدیدی بسازد و اطلاعات کاربران را به آنها انتقال دهد. فایل این اسکریپت را می‌توانید از لینک [دریافت کنید](https://github.com/suhasj/SQLMembership-Identity-OWIN).

این اسکریپت مختص این مقاله است. اگر الگوی استفاده شده برای جداول سیستم عضویت شما ویرایش/سفارشی-سازی شده باشد این اسکریپت را هم بر اساس این تغییرات بروز رسانی کنید.

پنجره Server Explorer را باز کنید. گره اتصال ApplicationServices را باز کنید تا جداول را مشاهده کنید. روی گره Tables کلیک راست کرده و گزینه New Query را انتخاب کنید.



در پنجره کوئری باز شده، تمام محتويات فایل *Migrations.sql* را کپی کنید. سپس اسکریپت را با کلیک کردن دکمه **Execute** اجرا کنید.

```

SQLQuery1.sql * X
B1615B31750BE7C9C207609833D8K

DROP TABLE AspNetUserRoles;

DROP TABLE AspNetUserClaims;

DROP TABLE AspNetUserLogins;

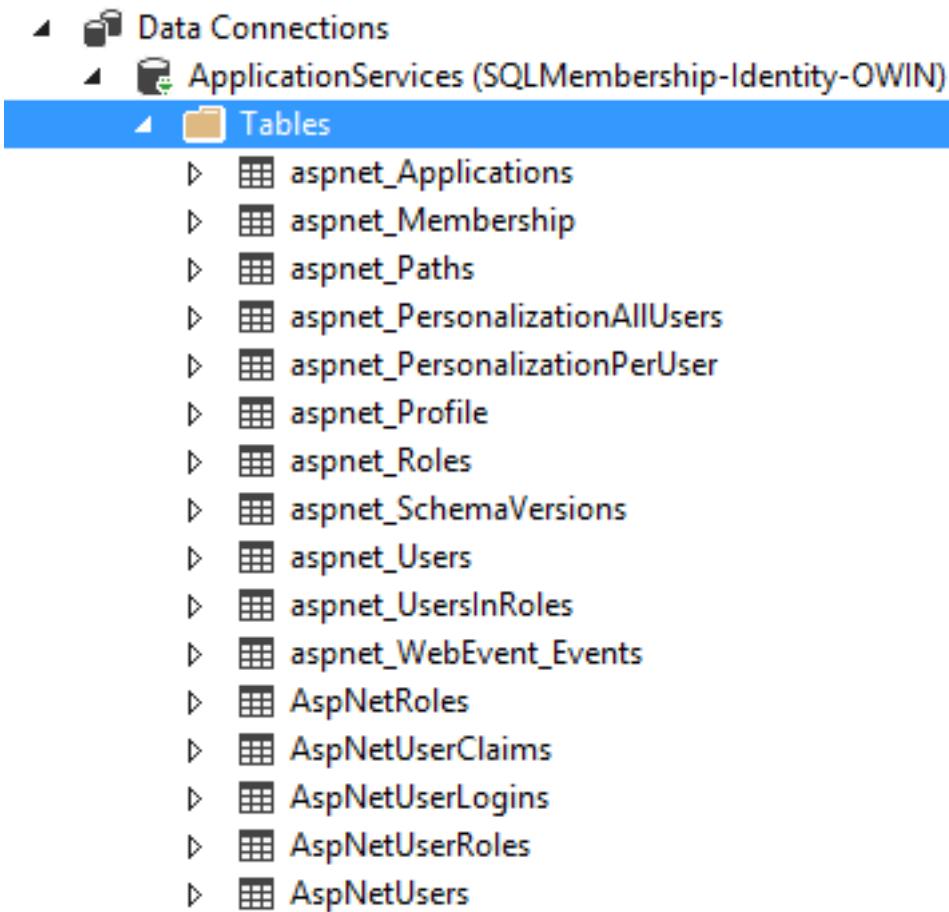
DROP TABLE AspNetRoles;

DROP TABLE AspNetUsers;

CREATE TABLE [dbo].[AspNetUsers] (
    [Id] NVARCHAR (128) NOT NULL,
    [UserName] NVARCHAR (MAX) NULL,
    [PasswordHash] NVARCHAR (MAX) NULL,
    [SecurityStamp] NVARCHAR (MAX) NULL,
    [Discriminator] NVARCHAR (128) NOT NULL,
    [ApplicationId] UNIQUEIDENTIFIER NOT NULL,
    [PasswordFormat] INT DEFAULT ((0)) NULL,
    [PasswordSalt] NVARCHAR (128) NULL,
    [LegacyPasswordHash] NVARCHAR (MAX) NULL,
    [LoweredUserName] NVARCHAR (256) NOT NULL,
    [MobileAlias] NVARCHAR (16) DEFAULT (NULL) NULL,
    [IsAnonymous] BIT DEFAULT ((0)) NOT NULL,
    [LastActivityDate] DATETIME NOT NULL,
    [MobilePIN] NVARCHAR (16) NULL,
    [Email] NVARCHAR (256) NULL,
    [LoweredEmail] NVARCHAR (256) NULL,
    [PasswordQuestion] NVARCHAR (256) NULL,
    [PasswordAnswer] NVARCHAR (128) NULL,
    [IsApproved] BIT NOT NULL,
    [IsLockedOut] BIT NOT NULL,
    [CreateDate] DATETIME NOT NULL,
    [LastLoginDate] DATETIME NOT NULL,
    [LastPasswordChangedDate] DATETIME NOT NULL,
    [LastLockoutDate] DATETIME NOT NULL,
    [FailedPasswordAttemptCount] INT NOT NULL,
    [FailedPasswordAttemptWindowStart] DATETIME NOT NULL,
    [FailedPasswordAnswerAttemptCount] INT NOT NULL,
)

```

ممکن است با اخطاری موافق شوید مبنی بر آنکه امکان حذف (drop) بعضی از جداول وجود نداشت. دلیلش آن است که چهار عبارت اولیه در این اسکریپت، تمام جداول مربوط به Identity را در صورت وجود حذف می‌کنند. از آنجا که با اجرای اولیه این اسکریپت چنین جداولی وجود ندارند، می‌توانیم این خطاهای را نادیده بگیریم. حال پنجه Server Explorer را تازه (refresh) کنید و خواهید دید که پنج جدول جدید ساخته شده اند.



لیست زیر نحوه Map کردن اطلاعات از جداول Identity به سیستم SQL Membership را نشان می‌دهد.

aspnet_Roles --> **AspNetRoles**

aspnet_Users, aspnet_Membership --> **AspNetUsers**

aspnet_UsersInRoles --> **AspNetUserRoles**

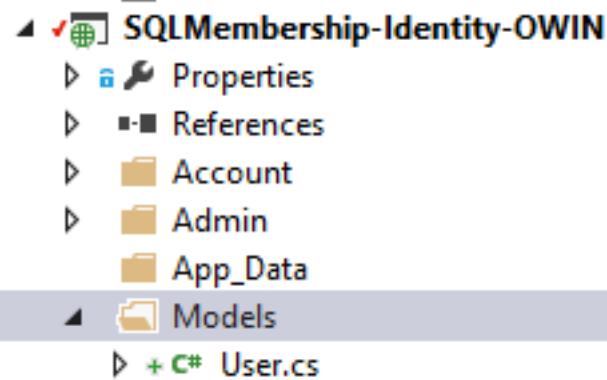
جدول AspNetUsers و AspNetUserClaims خالی هستند. فیلد تفکیک کننده (Discriminator) در جدول AspNetUserLogins باید مطابق نام کلاس مدل باشد، که در مرحله بعدی تعریف خواهد شد. همچنین ستون 'password' در فرم 'Format' می‌باشد. این شما را قادر می‌سازد تا از رمزنگاری برای ذخیره و بازیابی کلمه‌های عبور استفاده کنید. این مورد نیز در ادامه مقاله بررسی شده است.

ساختن مدل‌ها و صفحات عضویت

تصورت پیش فرض سیستم ASP.NET Identity برای دریافت و ذخیره اطلاعات در دیتابیس عضویت از Entity Framework استفاده می‌کند. برای آنکه بتوانیم با جداول موجود کار کنیم، می‌بایست ابتدا مدل‌هایی که الگوی دیتابیس را نمایندگی می‌کنند ایجاد کنیم. برای این کار مدل‌های ما یا باید اینترفیس‌های موجود در Identity.Core را پیاده سازی کنند، یا می‌توانند پیاده سازی‌های پیش فرض را توسعه دهند. پیاده سازی‌های پیش فرض در Microsoft.AspNet.Identity.EntityFramework وجود دارند.

در نمونه ما، جداول AspNetRole, AspNetClaims, AspNetLogins و AspNetUserRole ستون‌هایی دارند که شباهت زیادی به پیاده سازی‌های پیش فرض سیستم Identity دارند. در نتیجه می‌توانیم از کلاس‌های موجود، برای Map کردن الگوی جدید استفاده کنیم. جدول AspNetUsers ستون‌های جدیدی نیز دارد. می‌توانیم کلاس جدیدی بسازیم که از **IdentityUser** ارث بری کند و آن را گسترش دهیم تا این فیلهای جدید را پوشش دهد.

یوشه‌ای با نام **Models** بسازید (در صورتی که وجود ندارد) و کلاسی با نام **User** به آن اضافه کنید.



کلاس User باید کلاس Microsoft.AspNet.Identity.EntityFramework.IdentityUser را که در اسمبلی وجود دارد گسترش دهد. خصیت هایی را تعریف کنید که نماینده الگوی جدول AspNetUser خواص ID, Username, PasswordHash و SecurityStamp در کلاس IdentityUser تعریف شده اند، بنابراین این خواص را در لیست زیر نمی بینید.

```
public class User : IdentityUser
{
    public User()
    {
        CreateDate = DateTime.Now;
        IsApproved = false;
        LastLoginDate = DateTime.Now;
        LastActivityDate = DateTime.Now;
        LastPasswordChangedDate = DateTime.Now;
        LastLockoutDate = DateTime.Parse("1/1/1754");
        FailedPasswordAnswerAttemptWindowStart = DateTime.Parse("1/1/1754");
        FailedPasswordAttemptWindowStart = DateTime.Parse("1/1/1754");
    }

    public System.Guid ApplicationId { get; set; }
    public string MobileAlias { get; set; }
    public bool IsAnonymous { get; set; }
    public System.DateTime LastActivityDate { get; set; }
    public string MobilePIN { get; set; }
    public string Email { get; set; }
    public string LoweredEmail { get; set; }
    public string LoweredUserName { get; set; }
    public string PasswordQuestion { get; set; }
    public string PasswordAnswer { get; set; }
    public bool IsApproved { get; set; }
    public bool IsLockedOut { get; set; }
    public System.DateTime CreateDate { get; set; }
    public System.DateTime LastLoginDate { get; set; }
    public System.DateTime LastPasswordChangedDate { get; set; }
    public System.DateTime LastLockoutDate { get; set; }
    public int FailedPasswordAttemptCount { get; set; }
    public System.DateTime FailedPasswordAttemptWindowStart { get; set; }
    public int FailedPasswordAnswerAttemptCount { get; set; }
    public System.DateTime FailedPasswordAnswerAttemptWindowStart { get; set; }
    public string Comment { get; set; }
}
```

حال برای دسترسی به دیتابیس مورد نظر، نیاز به یک DbContext داریم. اسمبلی Microsoft.AspNet.Identity.EntityFramework کلاسی با نام IdentityDbContext دارد که پیاده سازی پیش فرض برای دسترسی به دیتابیس ASP.NET Identity است. نکته قابل توجه این است که IdentityDbContext آبجکتی از نوع TUser را می تواند هر کلاسی باشد که از IdentityUser ارث بری کرده و آن را گسترش می دهد.

در پوشه Models کلاس جدیدی با نام ApplicationDbContext بسازید که از IdentityDbContext ارث بری کرده و از کلاس

User استفاده می‌کند.

```
public class ApplicationDbContext : IdentityDbContext<User>
{
}
```

مدیریت کاربران در ASP.NET Identity توسط کلاسی با نام `UserManager` انجام می‌شود که در اسمبلی `Microsoft.AspNet.Identity.EntityFramework` قرار دارد. چیزی که ما در این مرحله نیاز داریم، کلاسی است که از ارث بری می‌کند و آن را طوری توسعه می‌دهد که از کلاس `User` استفاده کند.

در پوشه `Models` کلاس جدیدی با نام `UserManager` بسازید.

```
public class UserManager : UserManager<User>
{
}
```

کلمه عبور کاربران بصورت رمز نگاری شده در دیتابیس ذخیره می‌شوند. الگوریتم رمز نگاری SQL Membership با سیستم `Identity` تفاوت دارد. هنگامی که کاربران قدیمی به سایت وارد می‌شوند، کلمه عبورشان را توسط الگوریتم‌های قدیمی SQL Membership رمزگشایی می‌کنیم، اما کاربران جدید از الگوریتم‌های ASP.NET Identity استفاده خواهند کرد.

کلاس `UserManager` خاصیتی با نام `PasswordHasher` دارد. این خاصیت نمونه ای از یک کلاس را ذخیره می‌کند، که اینترفیس `IPasswordHasher` را پیاده سازی کرده است. این کلاس هنگام تراکنش‌های احراز هویت کاربران استفاده می‌شود تا کلمه‌های عبور را رمز نگاری/رمزگشایی شوند. در کلاس `UserManager` کلاس جدیدی بنام `SQLPasswordHasher` بسازید. کد کامل را در لیست زیر مشاهده می‌کنید.

```
public class SQLPasswordHasher : PasswordHasher
{
    public override string HashPassword(string password)
    {
        return base.HashPassword(password);
    }

    public override PasswordVerificationResult VerifyHashedPassword(string hashedPassword, string providedPassword)
    {
        string[] passwordProperties = hashedPassword.Split('|');
        if (passwordProperties.Length != 3)
        {
            return base.VerifyHashedPassword(hashedPassword, providedPassword);
        }
        else
        {
            string passwordHash = passwordProperties[0];
            int passwordformat = 1;
            string salt = passwordProperties[2];
            if (String.Equals(EncryptPassword(providedPassword, passwordformat, salt),
passwordHash, StringComparison.CurrentCultureIgnoreCase))
            {
                return PasswordVerificationResult.SuccessRehashNeeded;
            }
            else
            {
                return PasswordVerificationResult.Failed;
            }
        }
    }

    //This is copied from the existing SQL providers and is provided only for back-compat.
    private string EncryptPassword(string pass, int passwordFormat, string salt)
    {
        if (passwordFormat == 0) // MembershipPasswordFormat.Clear
        return pass;

        byte[] bIn = Encoding.Unicode.GetBytes(pass);
```

```

byte[] bSalt = Convert.FromBase64String(salt);
byte[] bRet = null;

if (passwordFormat == 1)
{ // MembershipPasswordFormat.Hashed
    HashAlgorithm hm = HashAlgorithm.Create("SHA1");
    if (hm is KeyedHashAlgorithm)
    {
        KeyedHashAlgorithm kha = (KeyedHashAlgorithm)hm;
        if (kha.Key.Length == bSalt.Length)
        {
            kha.Key = bSalt;
        }
        else if (kha.Key.Length < bSalt.Length)
        {
            byte[] bKey = new byte[kha.Key.Length];
            Buffer.BlockCopy(bSalt, 0, bKey, 0, bKey.Length);
            kha.Key = bKey;
        }
        else
        {
            byte[] bKey = new byte[kha.Key.Length];
            for (int iter = 0; iter < bKey.Length; )
            {
                int len = Math.Min(bSalt.Length, bKey.Length - iter);
                Buffer.BlockCopy(bSalt, 0, bKey, iter, len);
                iter += len;
            }
            kha.Key = bKey;
        }
        bRet = kha.ComputeHash(bIn);
    }
    else
    {
        byte[] bAll = new byte[bSalt.Length + bIn.Length];
        Buffer.BlockCopy(bSalt, 0, bAll, 0, bSalt.Length);
        Buffer.BlockCopy(bIn, 0, bAll, bSalt.Length, bIn.Length);
        bRet = hm.ComputeHash(bAll);
    }
}
}

return Convert.ToBase64String(bRet);
}
}

```

دقت کنید تا فضاهای نام System.Text و System.Security.Cryptography را وارد کرده باشید.

متد EncodePassword کلمه عبور را بر اساس پیاده سازی پیش فرض SQL Membership رمزنگاری می کند. این الگوریتم از System.Web گرفته می شود. اگر اپلیکیشن قدیمی شما از الگوریتم خاصی استفاده می کرده است، همینجا باید آن را منعکس کنید. دو متد دیگر نیز بنامهای VerifyHashedPassword و HashPassword نیاز داریم. این متدها از EncodePassword برای رمزنگاری کلمه های عبور و تایید آنها در دیتابیس استفاده می کنند.

سیستم SQL Membership برای رمزنگاری (Hash) کلمه های عبور هنگام ثبت نام و تغییر آنها توسط کاربران، از PasswordHash و PasswordFormat استفاده می کرد. در روند مهاجرت، این سه فیلد در ستون PasswordHash و PasswordSalt جدول SQL ذخیره شده و با کاراکتر ']' جدا شده اند. هنگام ورود کاربری به سایت، اگر کله عبور شامل این فیلدها باشد از الگوریتم Membership برای بررسی آن استفاده می کیم. در غیر اینصورت از پیاده سازی پیش فرض ASP.NET Identity استفاده خواهد شد. با این روش، کاربران قدیمی لازم نیست کلمه های عبور خود را صرفا بدلیل مهاجرت اپلیکیشن ما تغییر دهند.

کلاس UserManager را مانند قطعه کد زیر بروز رسانی کنید.

```

public UserManager()
    : base(new UserStore<User>(new ApplicationDbContext()))
{
    this.PasswordHasher = new SQLPasswordHasher();
}

```

ایجاد صفحات جدید مدیریت کاربران

قدم بعدی ایجاد صفحاتی است که به کاربران اجازه ثبت نام و ورود را می‌دهند. صفحات قدیمی SQL Membership از کنترل‌های استفاده می‌کنند که با ASP.NET Identity سازگار نیستند. برای ساختن این صفحات جدید به [این مقاله](#) مراجعه کنید. از آنجا که در این مقاله پروژه جدید را ساخته ایم و پکیج‌های لازم را هم نصب کرده ایم، می‌توانید مستقیماً به قسمت Adding Web Forms for registering users to your application بروید.

چند تغییر که باید اعمال شوند:

فایل‌های User.cs و Register.aspx.cs از کلاس UserManager استفاده می‌کنند. این ارجاعات را با کلاس Models جدیدی که در پوشش ساختید جایگزین کنید.

همچنین ارجاعات استفاده از کلاس IdentityUser را به کلاس User که در پوشش Models ساختید تغییر دهید. لازم است توسعه دهنده مقدار ApplicationId را برای کاربران جدید طوری تنظیم کند که با شناسه اپلیکیشن حاری تطابق داشته باشد. برای این کار می‌توانید پیش از ساختن حساب‌های کاربری جدید در فایل Register.aspx.cs ابتدا شناسه اپلیکیشن را بدست آورید و اطلاعات کاربر را بدرستی تنظیم کنید.

مثال: در فایل Register.cs متدهای تعریف کنید که جدول aspnet_Applications را بررسی می‌کند و شناسه اپلیکیشن را بر اساس نام اپلیکیشن بدست می‌آورد.

```
private Guid GetApplicationID()
{
    using (SqlConnection connection = new
SqlConnection(ConfigurationManager.ConnectionStrings["ApplicationServices"].ConnectionString))
    {
        string queryString = "SELECT ApplicationId from aspnet_Applications WHERE
ApplicationName = '/'; //Set application name as in database

        SqlCommand command = new SqlCommand(queryString, connection);
        command.Connection.Open();

        var reader = command.ExecuteReader();
        while (reader.Read())
        {
            return reader.GetGuid(0);
        }
    }

    return Guid.NewGuid();
}
```

حال می‌توانید این مقدار را برای آبجکت کاربر تنظیم کنید.

```
var currentApplicationId = GetApplicationID();

User user = new User() { UserName = Username.Text,
ApplicationId=currentApplicationId, ...};
```

در این مرحله می‌توانید با استفاده از اطلاعات پیشین وارد سایت شوید، یا کاربران جدیدی ثبت کنید. همچنین اطمینان حاصل کنید که کاربران پیشین در نقشهای مورد نظر وجود دارند.

مهاجرت به ASP.NET Identity مزايا و قابلیت‌های جدیدی را به شما ارائه می‌کند. مثل کاربران می‌توانند با استفاده از تامین کنندگان ثالثی مثل Facebook, Google, Microsoft, Twitter وغیره به سایت وارد شوند. اگر به [سورس کد این مقاله](#) مراجعه کنید خواهید دید که امکانات OAuth نیز فعال شده‌اند.

در این مقاله انتقال داده‌های پروفایل کاربران بررسی نشد. اما با استفاده از نکات ذکر شده می‌توانید پروفایل کاربران را هم بسادگی منتقل کنید. کافی است مدل‌های لازم را در پروژه خود تعریف کرده و با استفاده از اسکریپت‌های SQL داده‌ها را انتقال دهید.

نظرات خوانندگان

نویسنده: سمیرا قادری
تاریخ: ۱۳۹۲/۱۰/۲۳ ۱۳:۳۳

سلام ممنون از مطالب خوبتون .

زمانیکه در اپلکیشن خودم asp.net Configuration Security را انتخاب می نمایم با پیغام زیر مواجه می شوم
Unable to connect to SQL Server database

دستور زیر را هم اجرا کردم

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet_regsql.exe -S . -A all -d tt -U sa_tt -P asd123
در صورتیکه از login و password آن مطمئن هستم ولی مشکل حل نشد
در صورتیکه مشکل از permission چگونه باید آن را حل کنم
برنامه به شرح زیر است
Data Source=.;Initial Catalog=tt;User ID=sa_tt;Password=asd123
با تشکر
```

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۲۳ ۱۴:۴۵

تنظیمات وب کانفیگ خودتون رو با [نمونه MSDN](#) برای تعریف membership مطابقت بدید. [جزئیات قسمت های مختلف آن](#)

نویسنده: حمید
تاریخ: ۱۳۹۲/۱۰/۲۳ ۲۱:۱۰

خیلی ممنون آموزش های مفیدتون !

فرض کنید ما یک سیستم Authentication مثل Membership داریم با ۲ تا نقش. حالا می خواهیم نقش A به فرم F1 و بعضی از متدهاش دسترسی داشته و در فرم F2، نقش B به آن دسترسی داشته باشه. خوب تا اینجا فکر کنم پیاده سازیش راحت باشه. ولی مساله از جایی پیچیده میشه که ما بخواهیم در زمان اجرا این تنظیمات رو عوض کنیم مثلا: در فرم F1، نقش A دسترسیش به متدها و حتی به کل فرم تغییر کنه و یا اجازه دسترسی به فرم B بهش داده بشه و این شرایط رو برای چندین نقش و یا در سطح کاربر داشت توصیه چیه؟ از چه تکنولوژی و یا راهکاری میشه به این مقصود رسید.

خیلی ممنون

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۲۳ ۲۱:۳۷

پایه اش همین مسایل هست. فقط قسمتی که کار Authorization انجام میشه رو می توان سفارشی کرد. مثلا:

[Dynamic Controller/Action Authorization in ASP.NET MVC](#)
[MVC Dynamic Authorization](#)
[ASP.NET MVC Custom Authorize Attribute with Roles Parser](#)

نویسنده: کامران سادیں
تاریخ: ۱۳۹۲/۱۱/۰۸ ۲:۱۳

ممنون از مطلب خوبتون.

من جداول خودم رو برای احراز هویت دارم آیا میشه همین جداول رو با ASP.NET Identity کار کنم؟
اگر نمیشه، این بانک ASP.NET Identity که توی SQL Server نمیسازه بانک Database.mdf رو میسازه، میشه توی SQL بسازیم از

همون ابتدای پروژه؟

نویسنده: محسن خان
تاریخ: ۹:۲۰ ۱۳۹۲/۱۱/۰۸

سیستم کارش EF Code first هست. این سیستم کدهاش گره خورده به بانک اطلاعاتی خاصی نیست. الان در این مثال رشته اتصالی به یک localdb اشاره می‌کند. شما می‌توانید کلا این رشته و نحوه تعریف اون رو برای کار با SQL CE یا SQL Server یا هر بانک اطلاعاتی دیگری که پروایدر code first دارد، تغییر بدید و استفاده کنید. (و اگر با ef code first آشنایی ندارید، کم کم در آینده نمی‌توانید با کتابخانه‌های کمکی و جانبی دات نت کار کنید)

نویسنده: کامران سادین
تاریخ: ۱۰:۲۷ ۱۳۹۲/۱۱/۰۸

مقاله‌ای نیز راجع به تغییر فیلد هاش هم بنویسید بد نیست دوستان علاقه دارند. با تشکر.

نویسنده: محسن خان
تاریخ: ۱۰:۵۸ ۱۳۹۲/۱۱/۰۸

» [سفارشی کردن ASP.NET Identity در MVC 5](#) « بیشتر در اینجا

نویسنده: مهدی
تاریخ: ۱۲:۳۴ ۱۳۹۳/۰۱/۱۹

دوست عزیز برای حل این مشکل شما باید پوشه سورس برنامه رو تغییر بدین. این مشکل به دلیل وجود خط فاصله تو آدرس (دایرکتوری) محل قرار گیری پروژه هست، اگه محل پروژه رو در جایی قرار دهید که در آدرس آن از فاصله استفاده نشده باشد به خوبی کار خواهد کرد.

مايكروسافت در تاریخ 20 دسامبر 2013 پیش نمایش نسخه جدید ASP.NET Identity را معرفی کرد. تمرکز اصلی در این انتشار، رفع مشکلات نسخه 1.0 بود. امکانات جدیدی هم مانند Password Reset و Account Confirmation اضافه شده اند.

دانلود این انتشار

ASP.NET Identity را می‌توانید در قالب یک پکیج NuGet دریافت کنید. در پنجره Manage NuGet Packages می‌توانید پکیچ‌های Preview را لیست کرده و گزینه مورد نظر را نصب کنید. برای نصب پکیچ‌های pre-release Package Manager Console از فرامین زیر استفاده کنید.

```
Install-Package Microsoft.AspNet.Identity.EntityFramework -Version 2.0.0-alpha1 -Pre
```

```
Install-Package Microsoft.AspNet.Identity.Core -Version 2.0.0-alpha1 -Pre
```

```
Install-Package Microsoft.AspNet.Identity.OWIN -Version 2.0.0-alpha1 -Pre
```

دقت کنید که حتما از گزینه "Include Prerelease" استفاده می‌کنید. برای اطلاعات بیشتر درباره نصب پکیچ‌های Pre-release لطفا به [این لینک](#) و یا [این لینک](#) مراجعه کنید.

در ادامه لیست امکانات جدید و مشکلات رفع شده را می‌خوانید.

Account Confirmation

سیستم ASP.NET Identity حالا از Account Confirmation پشتیبانی می‌کند. این یک سفاری‌یوی بسیار رایج است. در اکثر وب سایت‌های امروزی پس از ثبت نام، حتما باید ایمیل خود را تایید کنید. پیش از تایید ثبت نام قادر به انجام هیچ کاری در وب سایت نخواهد بود، یعنی نمی‌توانید Login است. این روش مفید است، چرا که از ایجاد حساب‌های کاربری نامعتبر (bogus) جلوگیری می‌کند. همچنین این روش برای برقراری ارتباط با کاربران هم بسیار کارآمد است. از آدرس‌های ایمیل کاربران می‌توانید در وب سایت‌های فروم، شبکه‌های اجتماعی، تجارت آنلاین و بانکداری برای اطلاع رسانی و دیگر موارد استفاده کنید.

نکته: برای ارسال ایمیل باید تنظیمات SMTP را پیکربندی کنید. مثلا می‌توانید از سرویس‌های ایمیل محبوبی مانند [SendGrid](#) استفاده کنید، که با Windows Azure، که با

در مثال زیر نیاز دارید تا یک سرویس ایمیل برای ارسال ایمیل‌ها پیکربندی کنید. همچنین کاربران پیش از تایید ایمیل شان قادر به بازنشانی کلمه عبور نیستند.

Password Reset

این هم یک سفاری‌یوی رایج و استاندارد است. کاربران در صورتی که کلمه عبورشان را فراموش کنند، می‌توانند از این قابلیت برای بازنشانی آن استفاده کنند. کلمه عبور جدیدی بصورت خودکار تولید شده و برای آنها ارسال می‌شود. کاربران با استفاده از این رمز عبور جدید می‌توانند وارد سایت شوند و سپس آن را تغییر دهند.

Security Token Provider

هنگامی که کاربران کلمه عبورشان را تغییر می‌دهند، یا اطلاعات امنیتی خود را بروز رسانی می‌کنند (مثلا حذف کردن لگین‌های خارجی مثل فیسبوک، گوگل و غیره) باید شناسه امنیتی (security token) کاربر را بازتولید کنیم و مقدار قبلی را Invalidate یا بی اعتبار سازیم. این کار بمنظور حصول اطمینان از بی اعتبار بودن تمام شناسه‌های قبلی است که توسط کلمه عبور پیشین تولید شده بودند. این قابلیت، یک لایه امنیتی بیشتر برای اپلیکیشن شما فراهم می‌کند. چرا که وقتی کاربری کلمه عبورش را تغییر بدهد از همه جا logged-out می‌شود. یعنی از تمام مرورگرهایی که برای استفاده از اپلیکیشن استفاده کرده خارج خواهد شد.

برای پیکربندی تنظیمات این قابلیت می‌توانید از فایل Startup.Auth.cs استفاده کنید. می‌توانید مشخص کنید که میان افزار OWIN cookie هر چند وقت یکبار باید شناسه امنیتی کاربران را بررسی کند. به لیست زیر دقت کنید.

```
// Enable the application to use a cookie to store information for the signed in user
// and to use a cookie to temporarily store information about a user logging in with a third party
login provider
// Configure the sign in cookie
app.UseCookieAuthentication(newCookieAuthenticationOptions {
    AuthenticationType = DefaultAuthenticationTypes.ApplicationCookie,
    LoginPath = newPathString("/Account/Login"),
    Provider = newCookieAuthenticationProvider {
        OnValidateIdentity = SecurityStampValidator.OnValidateIdentity<ApplicationUserManager,
        ApplicationUser>()
```

```

        validateInterval: TimeSpan.FromSeconds(5),
        regenerateIdentity: (manager, user) => user.GenerateUserIdentityAsync(manager))
    }
});
```

امکان سفارشی کردن کلیدهای اصلی `Users` و `Roles`

در نسخه 1.0 نوع فیلد های کلید اصلی در جداول `Users` و `Roles` از نوع رشته (`string`) بود. این بدین معنا است که وقتی از `Sql Server` برای ذخیره داده های `ASP.NET Identity` استفاده می کنیم داده های این فیلد ها بعنوان `nvarchar` ذخیره می شوند. درباره این پیاده سازی پیش فرض در فروم هایی مانند سایت StackOverflow بسیار بحث شده است. و در آخر با در نظر گرفتن تمام بازخوردها، تصمیم گرفته شد یک نقطه توسعه پذیری (`extensibility`) اضافه شود که توسط آن بتوان نوع فیلد های اصلی را مشخص کرد. مثلاً شاید بخواهید کلیدهای اصلی جداول `Users` و `Roles` از نوع `int` باشند. این نقطه توسعه پذیری مخصوصاً هنگام مهاجرت داده های قبلی بسیار مفید است، مثلاً ممکن است دیتابیس قبلی فیلد های `UserId` را با فرمات `GUID` ذخیره کرده باشد.

اگر نوع فیلد های کلید اصلی را تغییر دهید، باید کلاس های مورد نیاز برای `Claims` و `Logins` را هم اضافه کنید تا کلید اصلی معتبری دریافت کنند. قطعه کد زیر نمونه ای از نحوه استفاده این قابلیت برای تعریف کلیدهای `int` را نشان می دهد.

```

de Snippet
public class ApplicationUser : IdentityUser<int, CustomUserLogin, CustomUserRole, CustomUserClaim>
{
}
public class CustomRole : IdentityRole<int, CustomUserRole>
{
    public CustomRole() { }
    public CustomRole(string name) { Name = name; }
}
public class CustomUserRole : IdentityUserRole<int> { }
public class CustomUserClaim : IdentityUserClaim<int> { }
public class CustomUserLogin : IdentityUserLogin<int> { }

public class ApplicationDbContext : IdentityDbContext<ApplicationUser, CustomRole, int, CustomUserLogin,
CustomUserRole, CustomUserClaim>
{ }
```

پشتیبانی از `Queryable` روی `Users` و `Roles`

کلاس های `UserStore` و `RoleStore` حالا از `Queryable` پشتیبانی می کنند، بنابراین می توانید براحتی لیست کاربران و نقش ها را کوئری کنید.

بعنوان مثال قطعه کد زیر دریافت لیست کاربران را نشان می دهد. از همین روش برای دریافت لیست نقش ها از `RoleManager` می توانید استفاده کنید.

```

// GET: /Users/
public async Task<ActionResult> Index()
{
    return View(await UserManager.Users.ToListAsync()); }
```

پشتیبانی از عملیات `Delete` از طریق `UserManager`

در نسخه 1.0 اگر قصد حذف یک کاربر را داشتید، نمی توانستید این کار را از طریق `UserManager` انجام دهید. اما حالا می توانید مانند قطعه کد زیر عمل کنید.

```

var user = await UserManager.FindByIdAsync(id);
if (user == null)
{
    return HttpNotFound(); }
```

```

    }

var result = await UserManager.DeleteAsync(user);

```

میان افزار **UserManagerFactory**

شما می‌توانید با استفاده از یک پیاده سازی Factory، و هله ای از OWIN context UserManager را از دریافت کنید. این الگو مشابه چیزی است که برای گرفتن AuthenticationManager در OWIN context استفاده می‌کنیم. این الگو همچنین روش توصیه شده برای گرفتن یک نمونه از UserManager به ازای هر درخواست در اپلیکیشن است. قطعه کد زیر نحوه پیکربندی این میان افزار در فایل StartupAuth.cs را نشان می‌دهد.

```

// Configure the UserManager
app.UseUserManagerFactory<newUserManagerOptions<ApplicationUserManager>()
{
    DataProtectionProvider = app.GetDataProtectionProvider(),
    Provider = newUserManagerProvider<ApplicationUserManager>()
    {
        OnCreate = ApplicationUserManager.Create
    }
});

```

و برای گرفتن یک و هله از `:UserManager`
`HttpContext.GetOwinContext().GetUserManager<ApplicationUserManager>();`

میان افزار **DbContextFactory**

سیستم Entity Framework از ASP.NET Identity برای ذخیره داده هایی در Sql Server استفاده می‌کند. بدین منظور، ASP.NET Identity کلاس ApplicationDbContext را رفرنس می‌کند. میان افزار DbContextFactory به ازای هر درخواست در اپلیکیشن یک و هله از ApplicationDbContext را به شما تحویل می‌دهد. می‌توانید پیکربندی لازم را در StartupAuth.cs انجام دهید.

```
app.UseDbContextFactory<ApplicationDbContext>(Create);
```

Samples

امکانات جدید را می‌توانید در پروژه <https://aspnet.codeplex.com> پیدا کنید. لطفاً به پوشه Identity در سورس کد مراجعه کنید. برای اطلاعاتی درباره نحوه اجرای پروژه هم فایل `readme` را بخوانید.
 برای مستندات ASP.NET Identity 1.0 هم <http://www.asp.net/identity> سر بزنید. هنوز مستنداتی برای نسخه 2.0 منتشر نشده، اما بازودی با انتشار نسخه نهایی مستندات و مثال‌های جدیدی به سایت اضافه خواهد شد.

Known Issues

در کنار قابلیت‌های جدیدی مانند Password Reset و Account Confirmation، دو خاصیت جدید به کلاس `IdentityUser` اضافه شده اند: 'IsConfirmed' و 'Email'. این تغییرات الگوی دیتابیسی که توسط ASP.NET Identity 1.0 ساخته شده است را تغییر می‌دهد. بروز رسانی پکیج‌ها از نسخه 1.0 به 2.0 باعث می‌شود که اپلیکیشن شما دیگر قادر به دسترسی به دیتابیس عضویت نباشد، چرا که مدل دیتابیس تغییر کرده. برای بروز رسانی الگوی دیتابیس می‌توانید از Code First Migrations استفاده کنید. نکته: نسخه جدید به EntityFramework 6.1.0-alpha1 وابستگی دارد، که در همین تاریخ (20 دسامبر 2013) پیش نمایش شد.

<http://blogs.msdn.com/b/adonet/archive/2013/12/20/ef-6-1-alpha-1-available.aspx>

بروز رسانی هایی دارد که سناریوی مهاجرت در ASP.NET Identity را تسهیل می‌کند، به همین دلیل از نسخه جدید EF استفاده شده. تیم ASP.NET هنوز باگ‌های زیادی را باید رفع کند و قابلیت‌های جدیدی را هم باید پیاده سازی کند. بنابراین پیش از نسخه نهایی RTM شاهد پیش نمایش‌های دیگری هم خواهیم بود که در ماه‌های آتی منتشر می‌شوند. برای اطلاعات بیشتر درباره آینده ASP.NET Identity به لینک زیر سری بزنید.

<https://aspnetidentity.codeplex.com/wikipage?title=Roadmap&version=1>

نظرات خوانندگان

نویسنده: ابوالفضل رجب پور
تاریخ: ۲۰:۴۴ ۱۳۹۲/۱۰/۳۰

سلام

پیاده سازی Single Sign on در این سیستم کجا کار قرار دارد؟ در واقع چطور میشه پیاده سازی ش کرد؟ در سیستم membership قبلی، اگر کلید اپلیکیشن رو در وب کانفیگ برنامه هاتون که دامین هاشون مشترک بود (در واقع ساب دامین ها)، یکسان وارد میکردی، برنامه ها بصورت SSO کار می کرد و احتیاجی به هیچ کاری نداشت. حالا در سیستم جدید همون روش جواب میده؟ برای برنامه های با دامین های متفاوت چطور؟

نویسنده: محسن خان
تاریخ: ۲۳:۷ ۱۳۹۲/۱۰/۳۰

رو باید تنظیم کنید. [CookieDomain](#)

عنوان:	تغییر فضای نام کلاس poco استفاده شده در wcf و از کار افتادن برنامه‌ی مشتری بدون دریافت پیام خطا
نویسنده:	ابوالفضل رجب پور
تاریخ:	۱۱:۳۵ ۱۳۹۲/۱۰/۲۴
آدرس:	www.dotnettips.info
گروه‌ها:	ASP.NET, WCF, namespace

چند وقت پیش در پروژه‌ای یک سرویس WCF داشتم که اطلاعاتی را در قالب یک کلاس poco برگشت می‌داد. اخیراً بعد از اصلاحاتی در پروژه متوجه شدم که سرویس کار نمی‌کند. هیچ خطای هم وجود نداشت. شروع به دیباگ کردم و متوجه شدم که سرویس برنامه اطلاعات را برگشت می‌دهد، اما برنامه‌ی مشتری تعداد اطلاعات دریافتی را صفر اعلام می‌کند و هیچ خطای هم گزارش نمی‌شود.

چون اطلاعات در قالب باینری در قسمتی از کلاس poco برگشت می‌شد، ابتدا حدسام حجم فایل بود.
اطلاعات کلاس poco:

```
public class OutgoingJob
{
    public int Id;
    public string JobId;
    public string Subject;
    public string Reciver;
    public byte[][] Attachments;
}
```

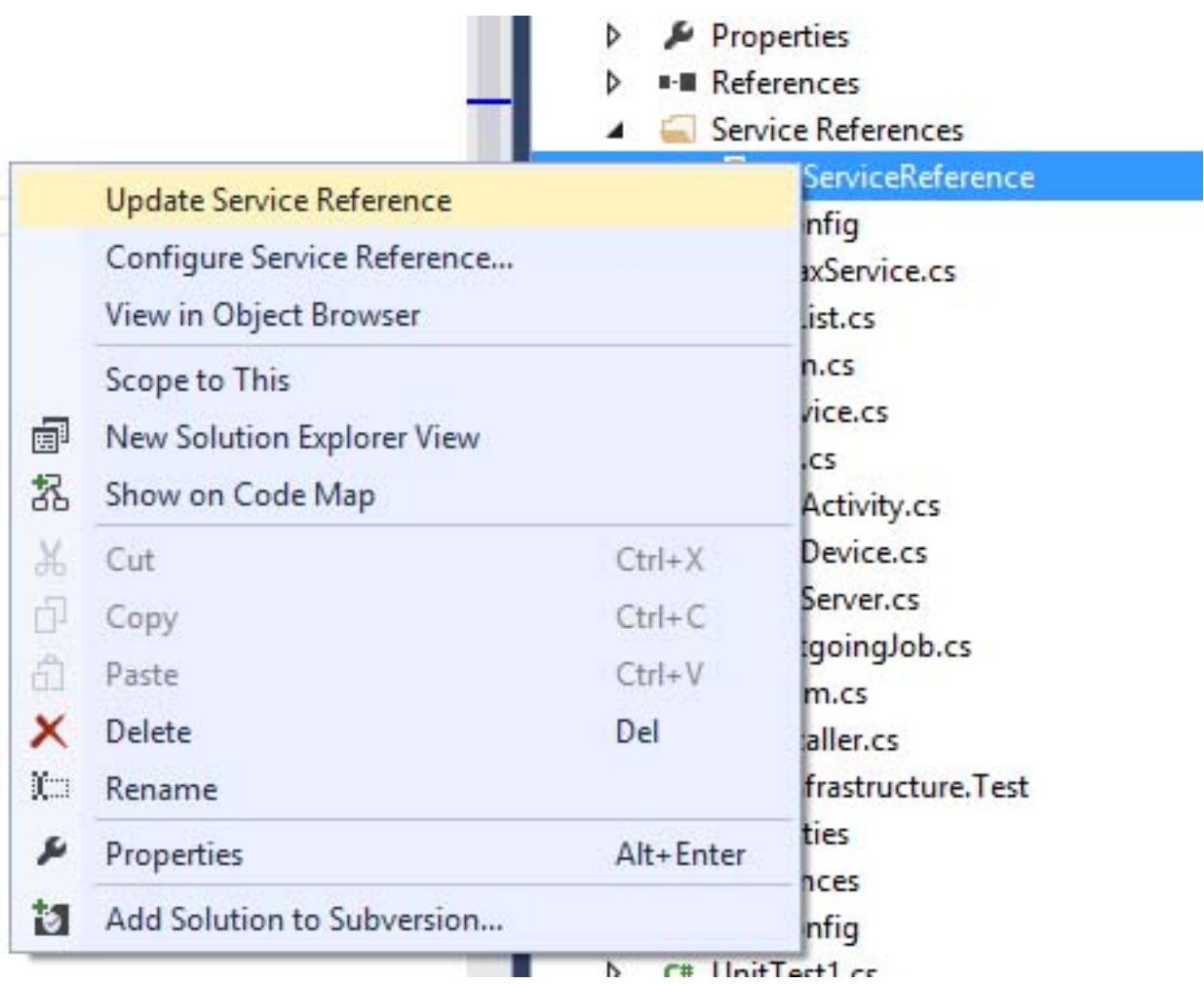
تنظیمات سایز ارسال و دریافت رو به حداکثر رسوندم. هیچ فایده‌ای نداشت. برنامه‌ی مشتری به راحتی به سرویس وصل می‌شد و با سایر متدهایی که خروجی‌های تایپ‌های اصلی مثل bool و string را برmi گرداند کار می‌کرد. فقط با متدهایی که لیست poco داشت، تعداد لیست اطلاعات دریافتی ۰ اعلام می‌شد.
متدهای WCF برای برگشت اطلاعات و لاغ کردن وقایع:

```
public List<OutgoingJob> GetJobsList(int Count)
{
    LogEvent("GetFaxsList Start...");
    List<OutgoingJob> OutgoingJobs = new List<OutgoingJob>();
    // business for fill list
    LogEvent("return job Count = " + OutgoingJobs.Count);
    return OutgoingJobs;
}
```

```
[TestMethod]
public void TestMethod1()
{
    jobService ser = new jobService();
    var listjob = ser.GetJobsList(5);
    Assert.AreNotEqual(0, listjob.Count);
}
```

لاغ‌های متدهای WCF، تعداد را ۱ اعلام می‌کند، اما تست، نتیجه را صفر برmi گرداند.
بعد از کلی کلنچار با تنظیمات binding و serviceBehaviors متوجه شدم که اشکال کار یه نکته‌ی کوچک خیلی ساده است. من هیچ تغییری در کلاس‌ها نداده بودم، اما برای مدیریت بهتر پروژه، فضای نام کلاس‌ها را تغییر داده بودم و مسبب همه‌ی مشکلات و وقت کشی‌ها همین بود.

راه حل هم که ساده است. هنگامی که فضای نام کلاس‌های برگشتی را تغییر می‌دهید، حتماً باید update service reference را در برنامه‌ی مشتری اجرا کنید تا اطلاعات سرویس بروز شود.



مشکل و راه حل خیلی ساده بود، ولی از من که خیلی وقت گرفت. امیدوارم وقت دوستان مثل من هدر نره موفق باشید

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۲۴ ۱۱:۴۱

استفاده از روش دیگر اتصال به سرور هم می‌توانه کمک کنه: [مقایسه بین Proxy و ChannelFactory در WCF](#)

نویسنده: افشار محبی
تاریخ: ۱۳۹۲/۱۰/۲۴ ۱۲:۵

بله کاملاً درسته. روش Channel Factory می‌توانه کمک خیلی خوبی به کاهش این گونه مشکلات بکند.

نویسنده: ابوالفضل رجب پور
تاریخ: ۱۳۹۲/۱۰/۲۴ ۱۲:۲۸

با تشکر از ارجاع خوبتون.

مطلوب اینجا راجع به یک نکته و روش رفعش در روش اول بود. مزايا و معایب هر روش سرچای خود محفوظ

در ASP.NET، ما user-control را جهت استفاده مجدد و مستقل در صفحات ASPX ایجاد می‌کنیم. هر user-control properties عمومی، متد‌ها و یا delegates خاص خود است و زمانی که در یک صفحه وب جاسازی (embedded) یا فراخوانی (load) می‌شود بوسیله صفحه وب قابل استفاده است.
 بعد از درج user-control در صفحه وب و فراخوانی آن، ممکن است نیاز باشد مثلاً باکلیک بر روی دکمه‌ای از user-control متدی از صفحه اجرا شود. اما یک مشکل، زمانی که در حال ایجاد user-control هستید هیچ اطلاعی از صفحه ای که قرار است user-control در آن قرار بگیرد ندارید پس چگونه می‌توانیم به متد‌های آن دسترسی داشته باشیم؟!
 در کلاس Delegate، متدی بنام [DynamicInvoke](#) وجود دارد که برای فراخوانی (Invoke) متد اشاره شده در delegate استفاده می‌شود. ما از این متد برای صدا زدن یک متد صفحه وبی که در آن قرار دارد استفاده می‌کنیم.
 مثال:

```

public partial class CustomUserCtrl : System.Web.UI.UserControl
{
  private System.Delegate _delWithParam;
  private System.Delegate _delNoParam;

  // برای فراخوانی متد‌هایی از صفحه که دارای پارامتر هستند
  public Delegate PageMethodWithParamRef
  {
    set { _delWithParam = value; }
  }

  // برای فراخوانی متد‌هایی از صفحه که بدون پارامتر هستند
  public Delegate PageMethodWithNoParamRef
  {
    set { _delNoParam = value; }
  }

  protected void Page_Load(object sender, EventArgs e)
  {}

  protected void BtnMethodWithParam_Click(object sender, System.EventArgs e)
  {
    //Parameter to a method is being made ready
    object[] obj = new object[1];
    obj[0] = "Parameter Value" as object;
    _delWithParam.DynamicInvoke(obj);
  }

  protected void BtnMethodWithoutParam_Click(object sender, System.EventArgs e)
  {
    //Invoke a method with no parameter
    _delNoParam.DynamicInvoke();
  }
}
  
```

فرض کنید در user-control بالا، دو دکمه وجود دارد که متد `BtnMethodWithParam_Click` یک دکمه، و متد `BtnMethodWithoutParam_Click` به رویداد کلیک دکمه دیگر منتبه می‌کنیم، سپس دو عامل خصوصی (`Private`) را تعریف می‌کنیم و متد `DynamicInvoke` این عامل‌های خصوصی را در متد‌های `BtnMethodWithParam_Click` و `BtnMethodWithoutParam_Click` فراخوانی می‌کنیم حال کافیست عامل‌هایی در صفحه تعریف کنیم که این عامل‌ها به متد‌های مورد نظر صفحه اشاره کنند و این عامل‌های صفحه را در عامل‌های عمومی user-control قرار دهیم. در ادامه به پیاده سازی صفحه می‌پردازیم:
 ابتدا دو عامل تعریف می‌کنیم:

```

public partial class _Default : System.Web.UI.Page
{
  delegate void DelMethodWithParam(string strParam);
  delegate void DelMethodWithoutParam();
}
  
```

در رویداد Page_Load، یک وله از هر کدام از عامل‌های بالا که به متد (توجه: امضاء متدها با امضاء عامل‌ها یکسان است) مورد نظر ما در صفحه اشاره می‌کند ایجاد می‌کنیم:

```
protected void Page_Load(object sender, EventArgs e)
{
    DelMethodWithParam delParam = new DelMethodWithParam(MethodWithParam);
    عامل صفحه را به عامل عمومی تعریف شده در یوزر کنترل تخصیص می‌دهیم //
    this.UserCtrl.PageMethodWithParamRef = delParam;
    DelMethodWithoutParam delNoParam = new DelMethodWithoutParam(MethodWithNoParam);

    عامل صفحه را به عامل عمومی تعریف شده در یوزر کنترل تخصیص می‌دهیم //
    this.UserCtrl.PageMethodWithNoParamRef = delNoParam;
}
```

در زیر متدهایی خصوصی که در صفحه وجود دارند و قرار است با کلیک بر روی دکمه‌های user-control فراخوانی شوند را مشاهده می‌کنید:

```
// متد دارای پارامتری که قرار است در کنترل فراخوانی شود
private void MethodWithParam(string strParam)
{
    Response.Write("It has parameter: " + strParam);
}

// متد بدون پارامتری که قرار است در کنترل فراخوانی شود
private void MethodWithNoParam()
{
    Response.Write("It has no parameter.");
}
```

و در نهایت کد پیاده سازی نهایی صفحه ما بشكل زیر خواهد شد:

```
public partial class _Default : System.Web.UI.Page
{
    delegate void DelMethodWithParam(string strParam);
    delegate void DelMethodWithoutParam();

    protected void Page_Load(object sender, EventArgs e)
    {
        DelMethodWithParam delParam = new DelMethodWithParam(MethodWithParam);
        عامل صفحه را به عامل عمومی تعریف شده در یوزر کنترل تخصیص می‌دهیم //
        this.UserCtrl.PageMethodWithParamRef = delParam;
        DelMethodWithoutParam delNoParam = new DelMethodWithoutParam(MethodWithNoParam);

        عامل صفحه را به عامل عمومی تعریف شده در یوزر کنترل تخصیص می‌دهیم //
        this.UserCtrl.PageMethodWithNoParamRef = delNoParam;
    }

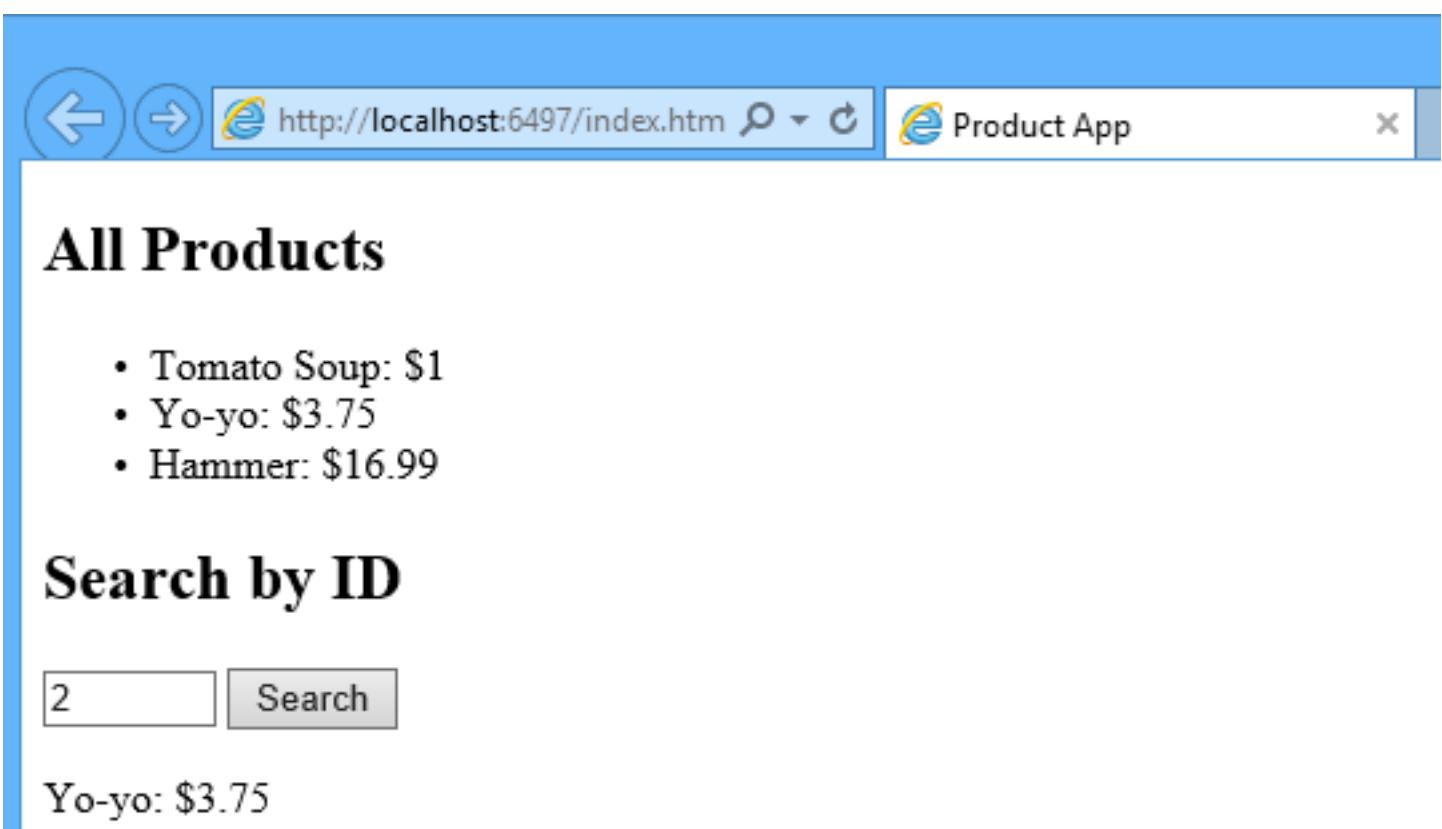
    // متد دارای پارامتری که قرار است در کنترل فراخوانی شود
    private void MethodWithParam(string strParam)
    {
        Response.Write("It has parameter: " + strParam);
    }

    // متد بدون پارامتری که قرار است در کنترل فراخوانی شود
    private void MethodWithNoParam()
    {
        Response.Write("It has no parameter.");
    }
}
```

برداشتی آزاد از [این مقاله](#).

HTTP تنها برای به خدمت گرفتن صفحات وب نیست. این پروتکل همچنین پلتفرمی قدرتمند برای ساختن API هایی است که سرویس‌ها و داده را در دسترس قرار می‌دهند. این پروتکل ساده، انعطاف پذیر و در همه جا حاضر است. هر پلتفرمی که فکرش را بتوانید بکنید کتابخانه‌ای برای HTTP دارد، بنابراین سرویس‌های HTTP می‌توانند بازه بسیار گسترده‌ای از کلاینت‌ها را پوشش دهند، مانند مرورگرها، دستگاه‌های موبایل و اپلیکیشن‌های مرسوم دسکتاپ.

فريم وركی برای ساختن API‌های وب بر روی فريم ورك دات نت است. در اين مقاله با استفاده از اين فريم ورك، API وبي خواهيم ساخت که ليستي از محصولات را براي گرداند. صفحه وب کلاینت، با استفاده از jQuery نتایج را نمایش خواهد داد.



All Products

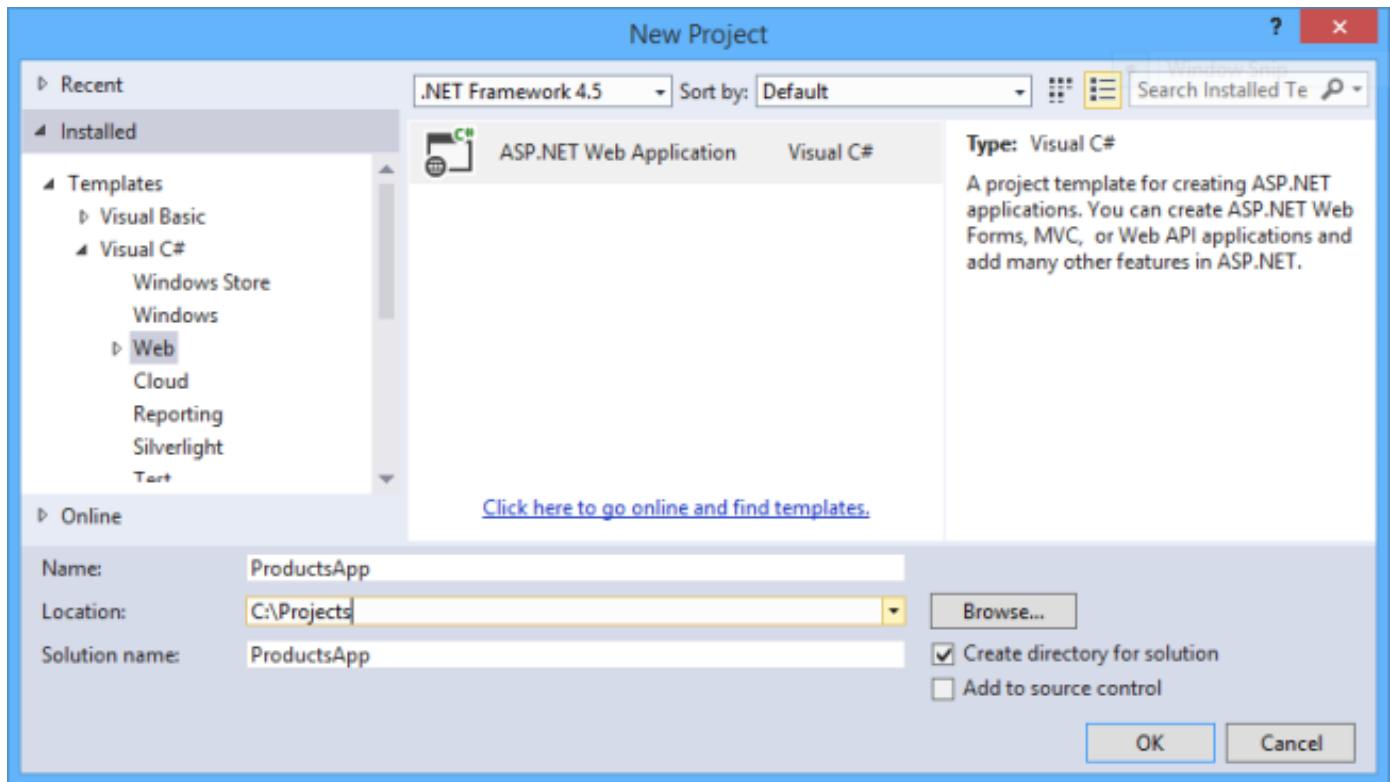
- Tomato Soup: \$1
- Yo-yo: \$3.75
- Hammer: \$16.99

Search by ID

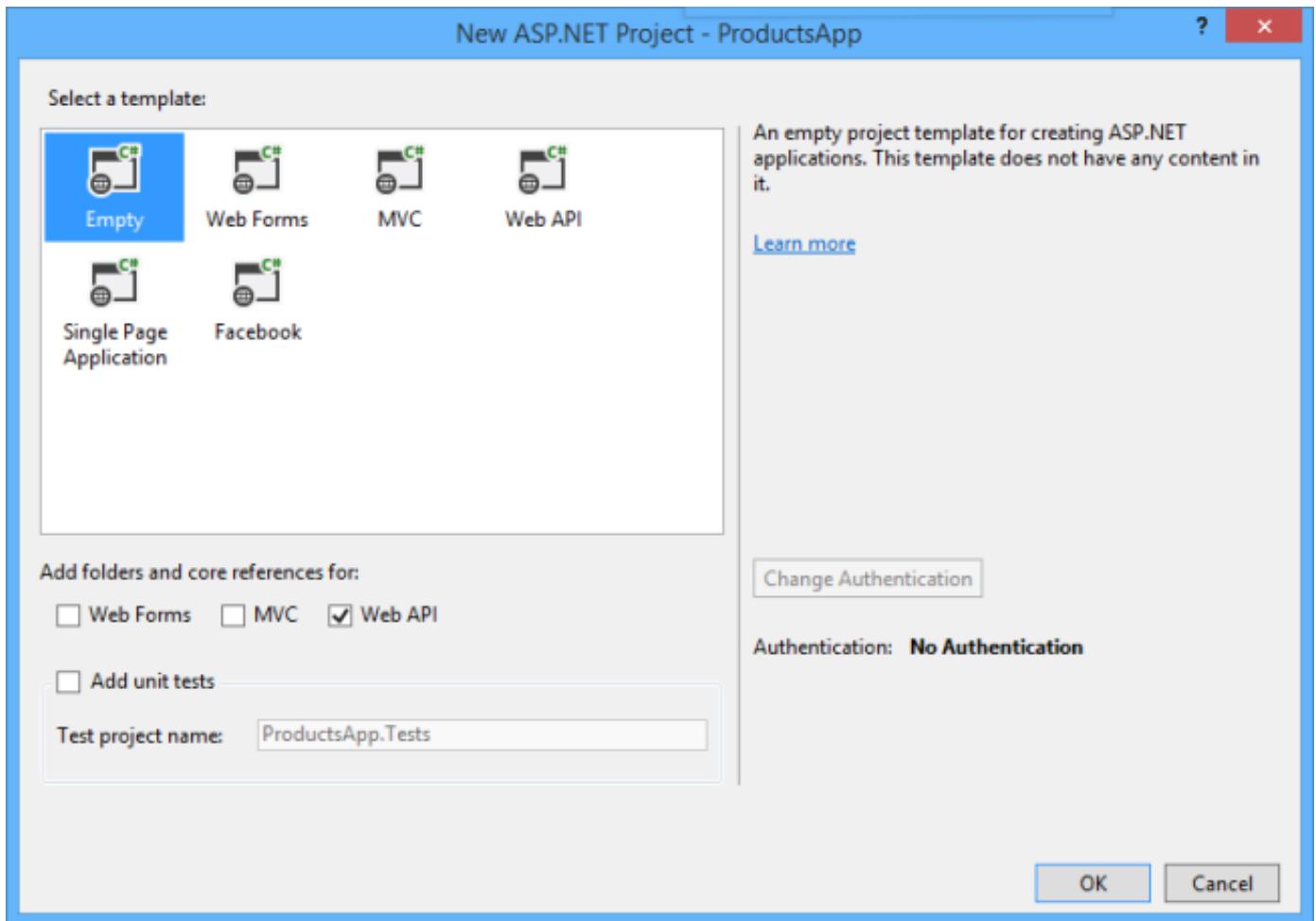
Yo-yo: \$3.75

يك پروژه Web API بسازيد

در ويزوال استوديو 2013 پروژه جديدی از نوع ASP.NET Web Application بسازيد و نام آن را "ProductsApp" انتخاب کنيد.



در دیالوگ "Add folders and core references for" قالب Empty New ASP.NET Project گزینه Web API را انتخاب نمایید.



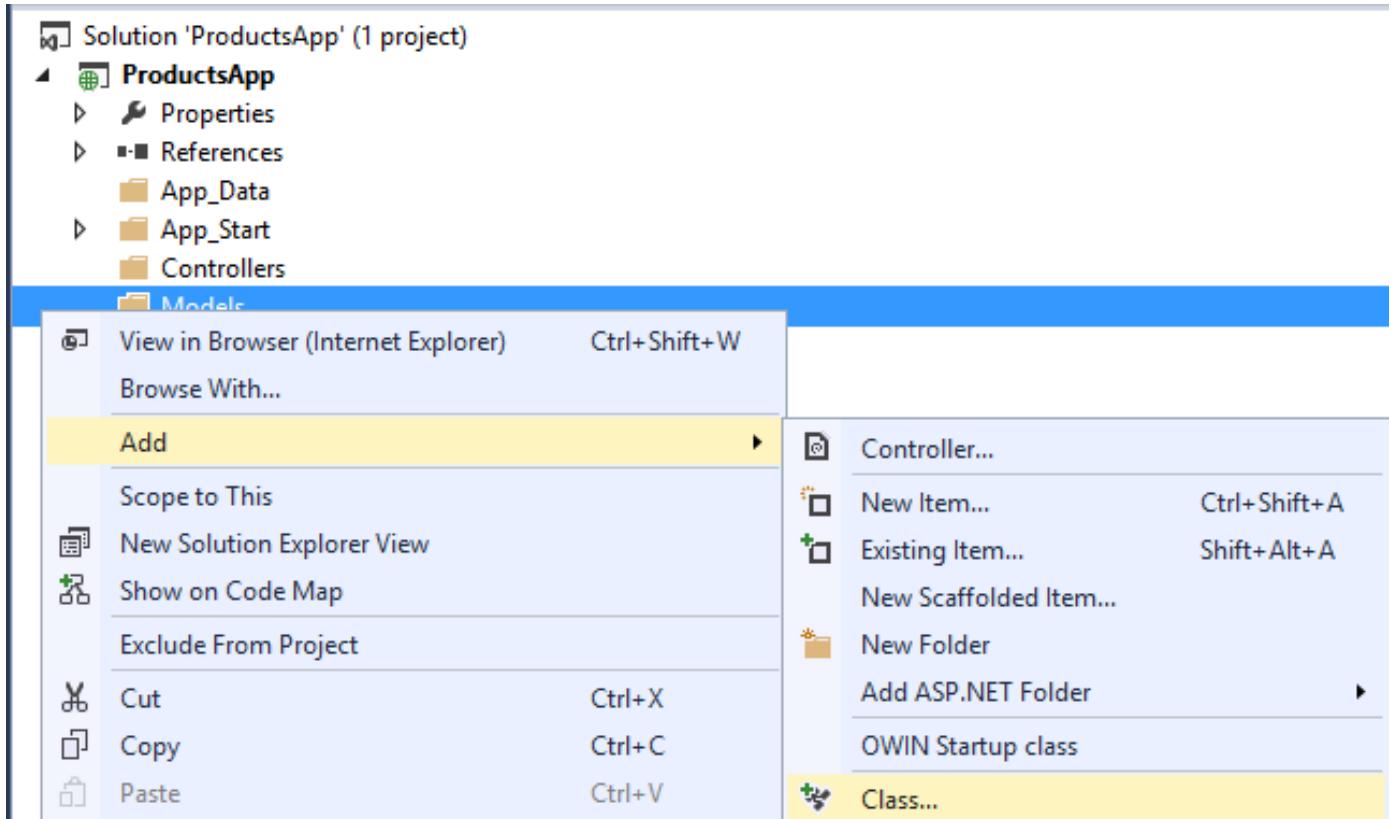
می توانید از قالب Web API هم استفاده کنید. این قالب با استفاده از ASP.NET MVC صفحات راهنمای API را خواهد ساخت. در این مقاله از قالب Empty استفاده میکنیم تا تمرکز اصلی، روی خود فریم ورک Web API باشد. بطور کلی برای استفاده از این فریم ورک لازم نیست با ASP.NET MVC آشنایی داشته باشید.

افزودن یک مدل

یک مدل (model) آبجکتی است که داده اپلیکیشن شما را معرفی میکند. ASP.NET Web API میتواند بصورت خودکار مدل شما را به XML, JSON و برخی فرمتهای دیگر مرتب (serialize) کند، و سپس داده مرتب شده را در پیام HTTP Response بنویسد. تا وقتی که یک کلاینت بتواند فرمت مرتب سازی دادهها را بخواند، میتواند آبجکت شما را deserialize کند. اکثر کلاینتها میتوانند XML یا JSON را تفسیر کنند. بعلاوه کلاینتها میتوانند فرمت مورد نظرشان را با تنظیم Accept header در پیام HTTP مشخص کنند. Request

بگذارید تا با ساختن مدلی ساده که یک محصول (product) را معرفی میکند شروع کنیم.

کلاس جدیدی در پوشه Models ایجاد کنید.



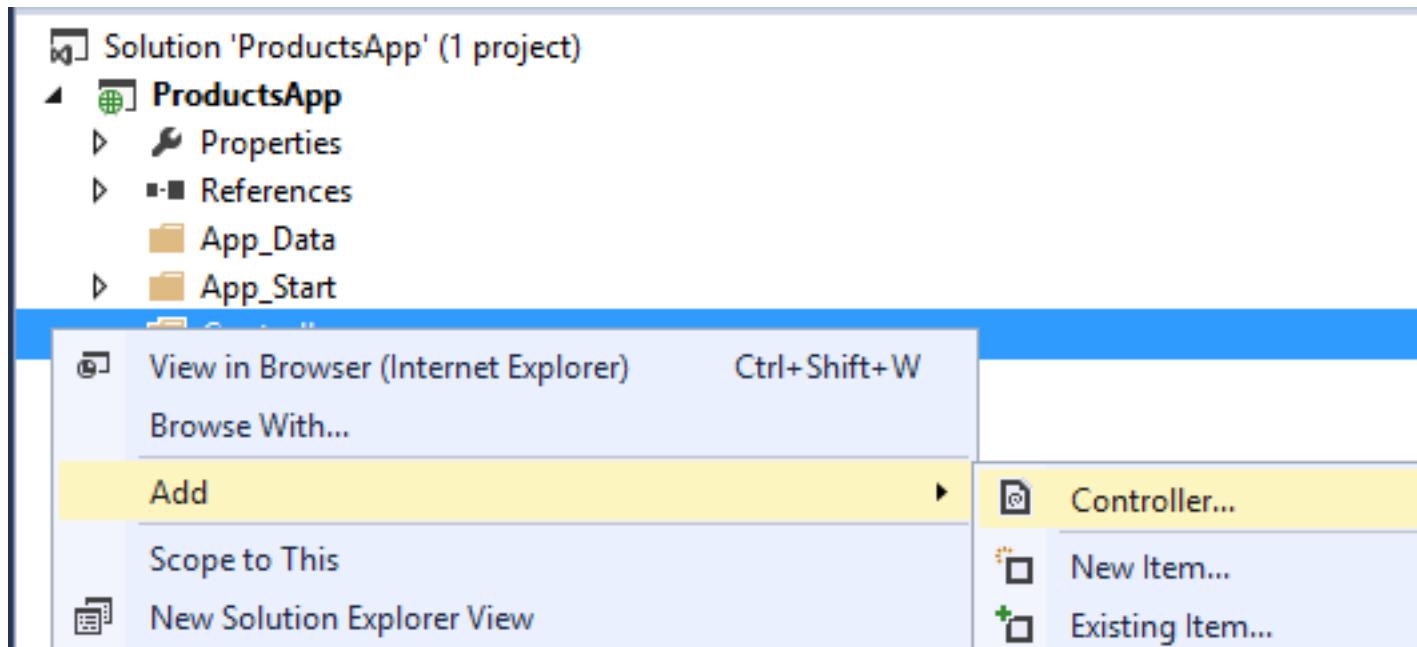
نام کلاس را به "Product" تغییر دهید، و خواص زیر را به آن اضافه کنید.

```
namespace ProductsApp.Models
{
    public class Product
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Category { get; set; }
        public decimal Price { get; set; }
    }
}
```

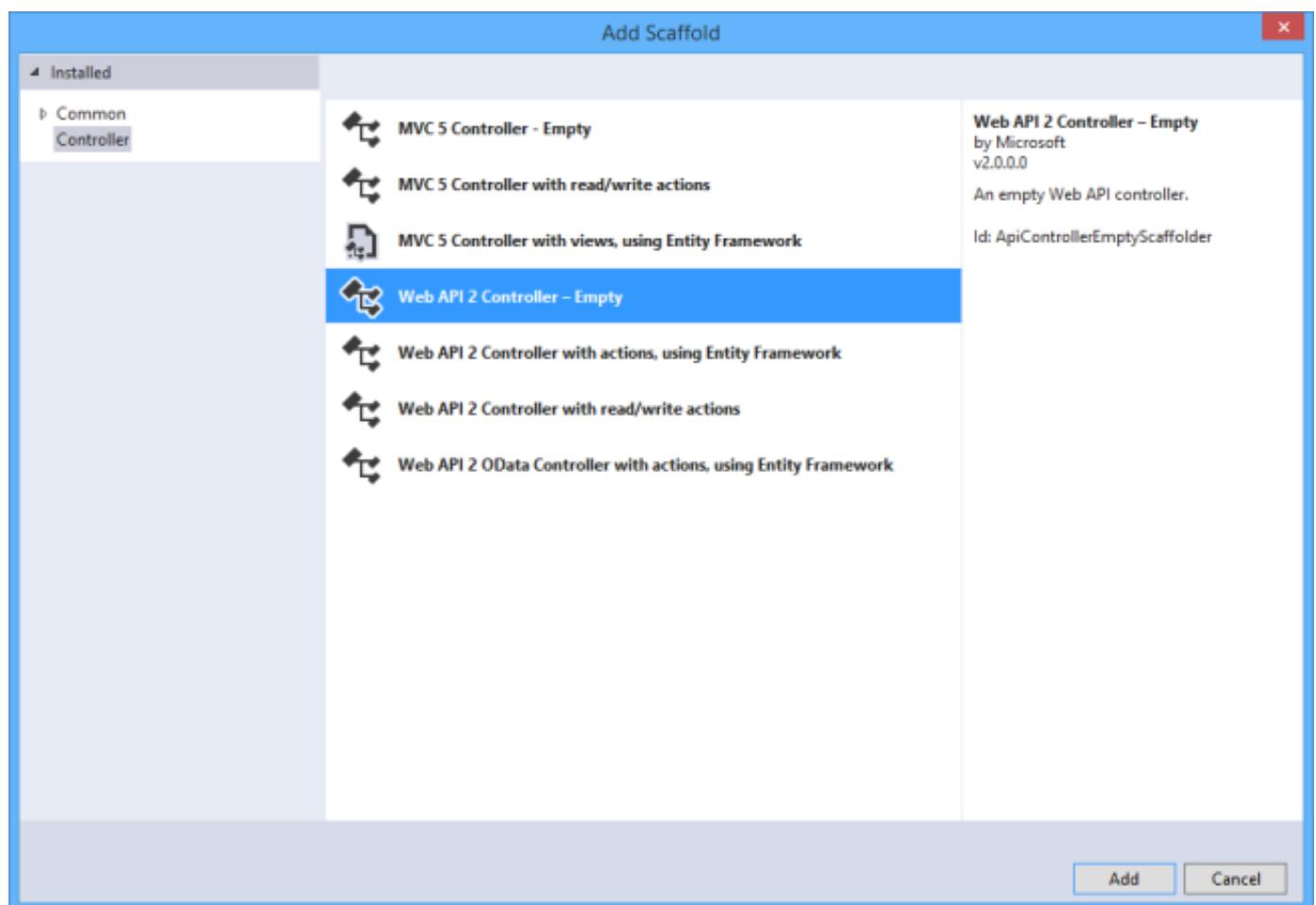
افزودن یک کنترلر

در API Web کنترلرها آبجکت هایی هستند که درخواست‌های HTTP را مدیریت کرده و آنها را به اکشن متدها نگاشت می‌کنند. ما کنترلری خواهیم ساخت که می‌تواند لیستی از محصولات، یا محصولی بخصوص را بر اساس شناسه برگرداند. اگر از ASP.NET MVC استفاده کرده اید، با کنترلرها آشنا هستید. کنترلرهای Web API مشابه کنترلرهای MVC هستند، با این تفاوت که بجای ارث بری از کلاس Controller از کلاس ApiController مشتق می‌شوند.

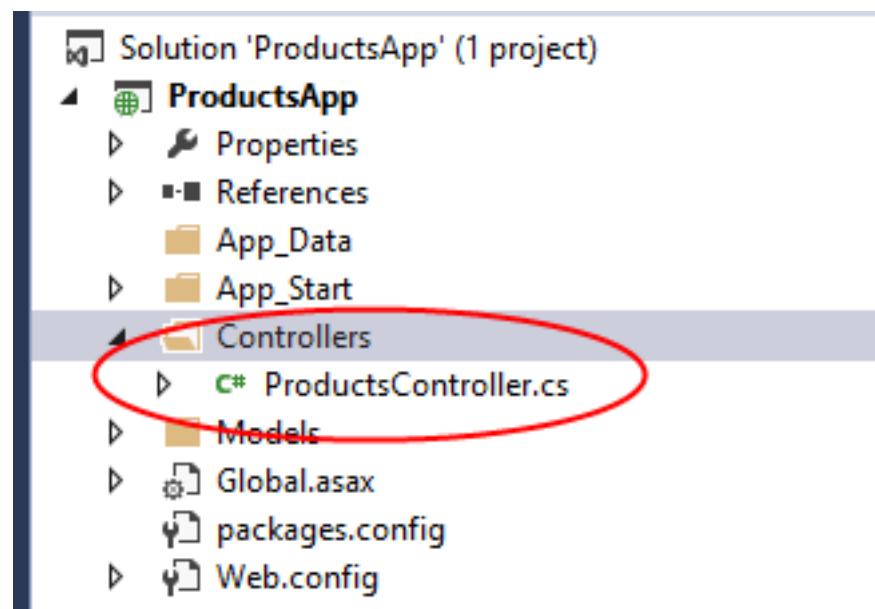
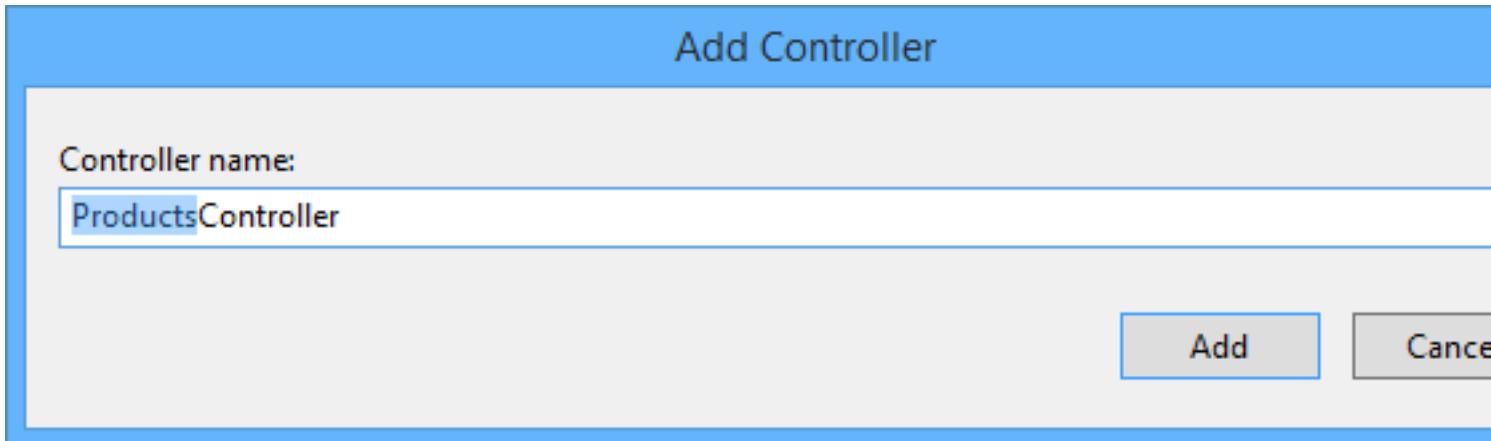
کنترلر جدیدی در پوشه Controllers ایجاد کنید.



در دیالوگ Add Scaffold گزینه Add Controller - Empty را انتخاب کرده و روی Add کلیک کنید.



در دیالوگ Add Controller نام کنترلر را به "ProductsController" تغییر دهید و روی Add کلیک کنید.



توجه کنید که ملزم به ساختن کنترلرهای خود در پوشه Controllers نیستید، و این روش صرفا قراردادی برای مرتب نگاه داشتن ساختار پروژه‌ها است. کنترلر ساخته شده را باز کنید و کد زیر را به آن اضافه نمایید.

```
using ProductsApp.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web.Http;

namespace ProductsApp.Controllers
{
    public class ProductsController : ApiController
    {
        Product[] products = new Product[]
        {
            new Product { Id = 1, Name = "Tomato Soup", Category = "Groceries", Price = 1 },
            new Product { Id = 2, Name = "Yo-yo", Category = "Toys", Price = 3.75M },
            new Product { Id = 3, Name = "Hammer", Category = "Hardware", Price = 16.99M }
        }
    }
}
```

```

};

public IEnumerable<Product> GetAllProducts()
{
    return products;
}

public IHttpActionResult GetProduct(int id)
{
    var product = products.FirstOrDefault((p) => p.Id == id);
    if (product == null)
    {
        return NotFound();
    }
    return Ok(product);
}
}

```

برای اینکه مثال جاری را ساده نگاه داریم، محصولات مورد نظر در یک آرایه استاتیک ذخیره شده اند. مسلمان در یک اپلیکیشن واقعی برای گرفتن لیست محصولات از دیتابیس یا منبع داده ای دیگر کوئری می‌گیرید.

کنترلر ما دو متد برای دریافت محصولات تعریف می‌کند:

متد GetAllProducts لیست تمام محصولات را در قالب یک `IEnumerable<Product>` بر می‌گرداند.
متد GetProductById سعی می‌کند محصولی را بر اساس شناسه تعیین شده پیدا کند.

همین! حالا یک Web API ساده دارید. هر یک از متدهای این کنترلر، به یک یا چند URI پاسخ می‌دهند:

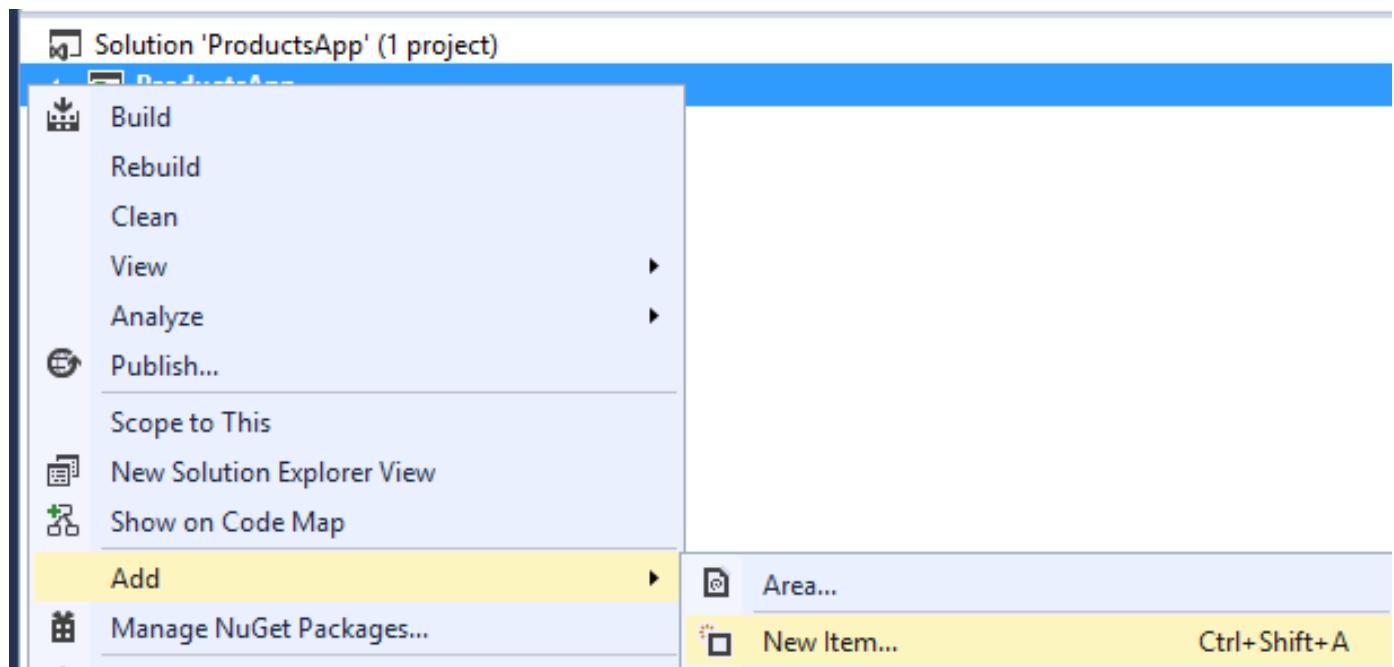
Controller Method	URI
GetAllProducts	api/products/
GetProductById	api/products/ <i>id</i> /

برای اطلاعات بیشتر درباره نحوه نگاشت درخواست‌های HTTP به اکشن متدها توسط Web API به [این لینک](#) مراجعه کنید.

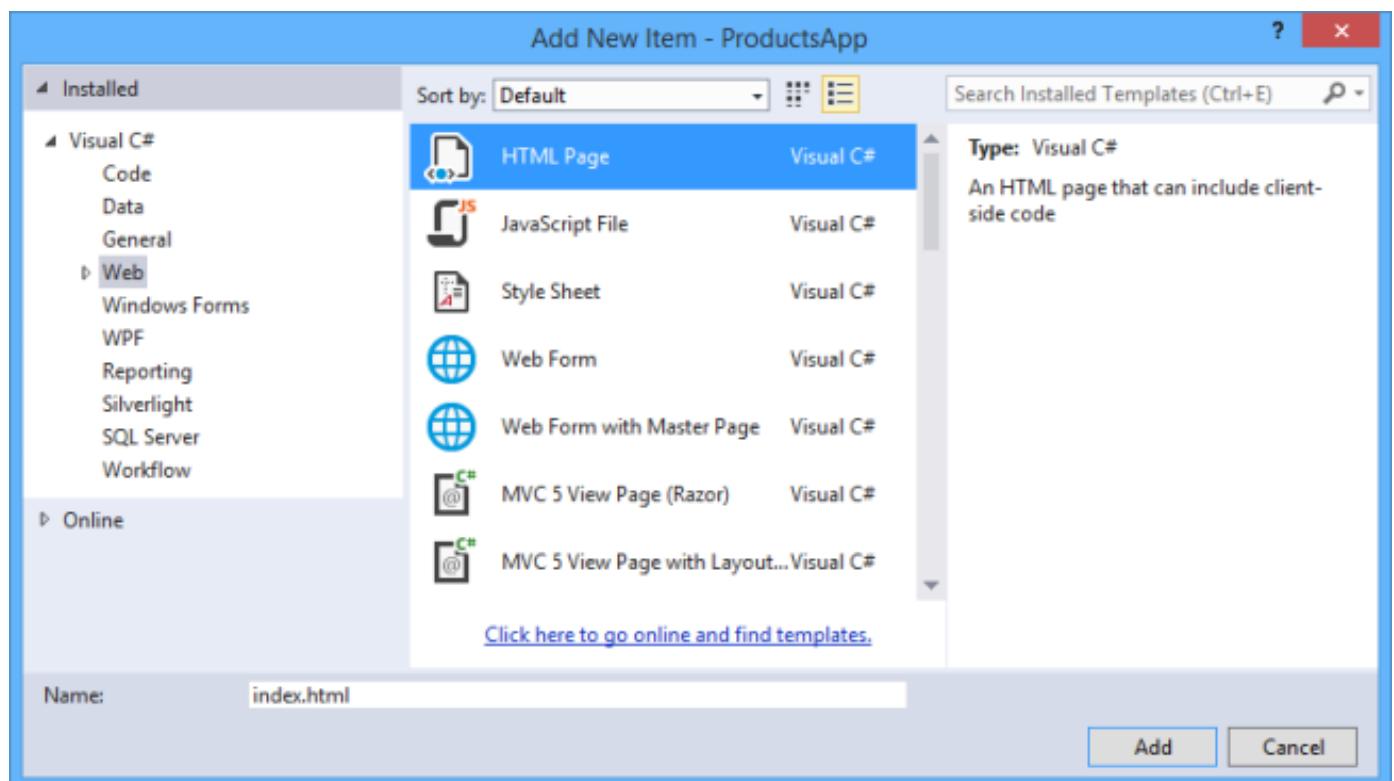
فرآخوانی Web API با جاوا اسکریپت و jQuery

در این قسمت یک صفحه HTML خواهیم ساخت که با استفاده از AJAX متدهای Web API را فراخوانی می‌کند. برای ارسال درخواست‌های آژاکسی و بروز رسانی صفحه بمنظور نمایش نتایج دریافتی از jQuery استفاده می‌کنیم.

در پنجره Solution Explorer روی نام پروژه کلیک راست کرده و گزینه Add, New Item را انتخاب کنید.



در دیالوگ Add New Item قالب HTML Page را انتخاب کنید و نام فایل را به "index.html" تغییر دهید.



حال محتوای این فایل را با لیست زیر جایگزین کنید.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```

```

<title>Product App</title>
</head>
<body>

<div>
  <h2>All Products</h2>
  <ul id="products" />
</div>
<div>
  <h2>Search by ID</h2>
  <input type="text" id="prodId" size="5" />
  <input type="button" value="Search" onclick="find();"/>
  <p id="product" />
</div>

<script src="http://ajax.aspnetcdn.com/ajax/jquery/jquery-2.0.3.min.js"></script>
<script>
  var uri = 'api/products';

  $(document).ready(function () {
    // Send an AJAX request
    $.getJSON(uri)
      .done(function (data) {
        // On success, 'data' contains a list of products.
        $.each(data, function (key, item) {
          // Add a list item for the product.
          $('- ', { text: formatItem(item) }).appendTo($('#products'));
        });
      });
  });

  function formatItem(item) {
    return item.Name + ': $' + item.Price;
  }

  function find() {
    var id = $('#prodId').val();
    $.getJSON(uri + '/' + id)
      .done(function (data) {
        $('#product').text(formatItem(data));
      })
      .fail(function (jqXHR, textStatus, err) {
        $('#product').text('Error: ' + err);
      });
  }
</script>
</body>
</html>

```

راههای مختلفی برای گرفتن jQuery وجود دارد، در این مثال از [Microsoft Ajax CDN](#) استفاده شده. می‌توانید این کتابخانه را از دانلود کنید و بصورت محلی استفاده کنید. همچنین قالب پروژه‌های Web API این کتابخانه را به پروژه نیز اضافه می‌کند.

گرفتن لیستی از محصولات

برای گرفتن لیستی از محصولات، یک درخواست HTTP GET به آدرس "/api/products" ارسال کنید.

تابع `getJSON` یک درخواست آزاسی ارسال می‌کند. پاسخ دریافتی هم آرایه‌ای از آبجکت‌های JSON خواهد بود. تابع `done` در صورت موفقیت آمیز بودن درخواست، اجرا می‌شود. که در این صورت ما DOM را با اطلاعات محصولات بروز رسانی می‌کنیم.

```

$(document).ready(function () {
  // Send an AJAX request
  $.getJSON(apiUrl)
    .done(function (data) {
      // On success, 'data' contains a list of products.
      $.each(data, function (key, item) {
        // Add a list item for the product.
        $('- ', { text: formatItem(item) }).appendTo($('#products'));
      });
    });
});

```

گرفتن محصولی مشخص

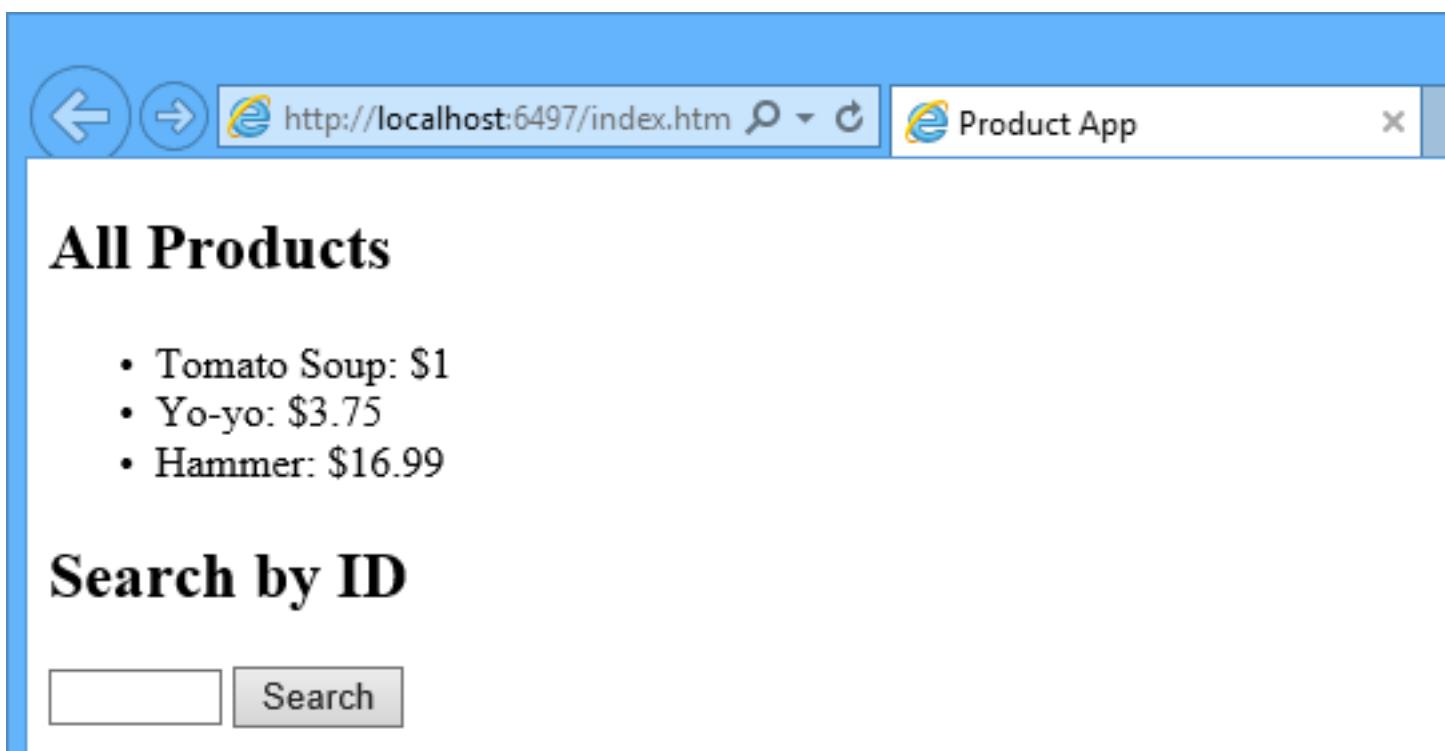
برای گرفتن یک محصول توسط شناسه (ID) آن کافی است یک درخواست GET HTTP به آدرس "/api/products/id" ارسال کنید.

```
function find() {
    var id = $('#prodId').val();
    $.getJSON(apiUrl + '/' + id)
        .done(function (data) {
            $('#product').text(formatItem(data));
        })
        .fail(function (jqXHR, textStatus, err) {
            $('#product').text('Error: ' + err);
        });
}
```

برای این کار هنوز از `getJSON` برای ارسال درخواست آژاکسی استفاده می‌کنیم، اما اینبار شناسه محصول را هم به آدرس درخواستی اضافه کرده ایم. پاسخ دریافتی از این درخواست، اطلاعات یک محصول با فرمت JSON است.

اجرای اپلیکیشن

اپلیکیشن را با F5 اجرا کنید. صفحه وеб باز شده باید چیزی مشابه تصویر زیر باشد.



برای گرفتن محصولی مشخص، شناسه آن را وارد کنید و روی Search کلیک کنید.

All Products

- Tomato Soup: \$1
- Yo-yo: \$3.75
- Hammer: \$16.99

Search by ID

2

Yo-yo: \$3.75

اگر شناسه نامعتبری وارد کنید، سرور یک خطای HTTP بر می‌گرداند.

استفاده از F12 برای مشاهده درخواست‌ها و پاسخ‌ها

هنگام کار با سرویس‌های HTTP، مشاهده‌ی درخواست‌های ارسال شده و پاسخ‌های دریافتی بسیار مفید است. برای اینکار می‌توانید از ابزار توسعه دهندگان وب استفاده کنید، که اکثر مرورگرهای مدرن، پیاده سازی خودشان را دارند. در اینترنت اکسپلورر می‌توانید با F12 به این ابزار دسترسی پیدا کنید. به برگه Network بروید و روی Start Capturing کلیک کنید. حالا صفحه وب را مجدداً بارگذاری (reload) کنید. در این مرحله اینترنت اکسپلورر ترافیک HTTP بین مرورگر و سرور را تسوییر می‌کند. می‌توانید تمام ترافیک HTTP روی صفحه جاری را مشاهده کنید.

The screenshot shows the F12 developer tools Network tab. The browser mode is set to IE9. The table lists network requests:

URL	Method	Result	Type	Received	Taken
http://localhost:26481/	GET	200	text/html	1.66 KB	78 ms
/Content/css?v=ji3nO1pdg6VLv3CV...	GET	200	text/css	0.89 KB	47 ms
/Content/themes/base/css?v=UM62...	GET	200	text/css	24.73 KB	63 ms
/Scripts/js?v=4h5lPNUsLiFoa0vqrItj...	GET	200	text/javascript	325.88...	62 ms
/Scripts/jquery-1.6.2.min.js	GET	304	application/x-ja...	176 B	47 ms
/api/products/	GET	200	application/json	398 B	281 ms

به دنبال آدرس نسبی "/api/products" بگردید و آن را انتخاب کنید. سپس روی Go to detailed view کلیک کنید تا جزئیات ترافیک را مشاهده کنید. در نمای جزئیات، می‌توانید headerها و بدنه درخواست‌ها و پاسخ‌ها را ببینید. مثلاً اگر روی برگه Request headers کلیک کنید، خواهید دید که اپلیکیشن ما در Accept header داده‌ها را با فرمت "application/json" درخواست کرده است.

The screenshot shows the Fiddler interface with the Network tab selected. The URL bar contains "URL: http://localhost:26481/api/products/". Below it, a table lists Request headers:

Key	Value
Request	GET /api/products/ HTTP/1.1
X-Requested-With	XMLHttpRequest
Accept	application/json, text/javascript, */*; q=0.0
Referer	http://localhost:26481/
Accept-Language	en-us
Accept-Encoding	gzip, deflate
User-Agent	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0; .NET4.0ED; .NET4.0E)
Host	localhost:26481
Connection	Keep-Alive

اگر روی برگه Response body کلیک کنید، می‌توانید بینید چگونه لیست محصولات با فرمت JSON سریال شده است. همانطور که گفته شده مرورگرهای دیگر هم قابلیت‌های مشابهی دارند. یک ابزار مفید دیگر [Fiddler](#) است. با استفاده از این ابزار می‌توانید تمام ترافیک HTTP خود را مانیتور کرده، و همچنین درخواست‌های جدیدی بسازید که این امر کنترل کاملی روی HTTP headers به شما می‌دهد.

قدم‌های بعدی

برای یک مثال کامل از سرویس‌های HTTP که از عملیات POST, PUT و DELETE پشتیبانی می‌کند به [این لینک](#) مراجعه کنید. برای اطلاعات بیشتر درباره طراحی و اکنش گرا در کنار سرویس‌های HTTP به [این لینک](#) مراجعه کنید، که اپلیکیشن‌های تک صفحه ای (SPA) را بررسی می‌کند.

نظرات خوانندگان

نویسنده: پوریا منفرد
تاریخ: ۱۳۹۳/۰۳/۰۵ ۱:۲۶

من یه سوالی برام پیش اومده اینه که میشه از Web API برای پروژه‌های بزرگ مبتنی بر روی HTTP استفاده کرد؟ منظورم از پروژه‌های بزرگ یعنی Request هایی که شاید اطلاعات برگشتی مثلا بیش از 1000 رکورد باشه آیا شدنیه؟

یعنی منبع منبع داده بتونه بوسیله Web API عملیات‌های Crud رو بر روی بستر اینترنت برای پروژه‌های این چنینی که امکان واکشی اطلاعات بیشمار و ورود اطلاعات همزمان بوسیله کاربرهای مختلف با دیوایس‌های مختلف وجود داره رو ارائه بد؟

نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۳/۰۳/۰۷ ۰:۷

بله. به طور کلی، هر پلتفرمی که دارای کتابخانه ای جهت کار با سرویس‌های Http است می‌تواند از سرویس‌های Asp.Net WebApi استفاده نماید.

اما در هنگام پیاده سازی پروژه‌های مقیاس بزرگ حتماً به طراحی زیر ساخت توجه ویژه ای داشته باشد. اگر کتابهای

[Designing Evolvable Web Api With Asp.Net](#)

آنها بیندازید.
در همین رابطه:

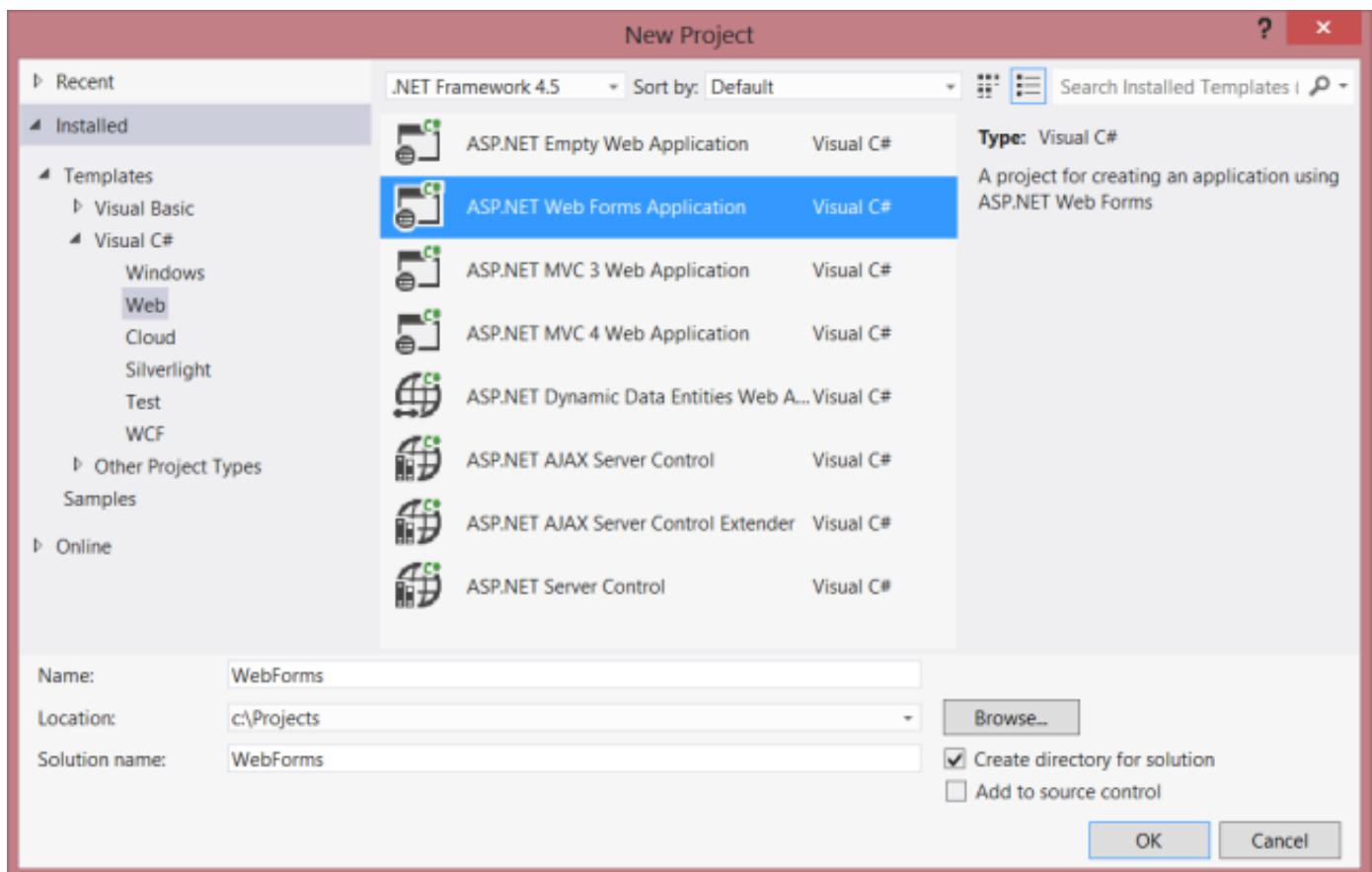
[«مقایسه بین امکانات WCF و Web Api](#)

گرچه ASP.NET MVC بهمراه ASP.NET Web API بسته بندی شده و استفاده می‌شود، اما اضافه کردن آن به اپلیکیشن‌های Forms کار ساده‌ای است. در این مقاله مراحل لازم را بررسی می‌کنیم.

برای استفاده از Web API در یک اپلیکیشن ASP.NET Web Forms دو قدم اصلی باید برداشته شود:

اضافه کردن یک کنترلر Web API که از کلاس **ApiController** مشتق می‌شود.
 اضافه کردن مسیرهای جدید به متد **Application_Start**.

یک پروژه Web Forms بسازید
 ویژوال استودیو را اجرا کنید و پروژه جدیدی از نوع ASP.NET Web Forms Application ایجاد کنید.

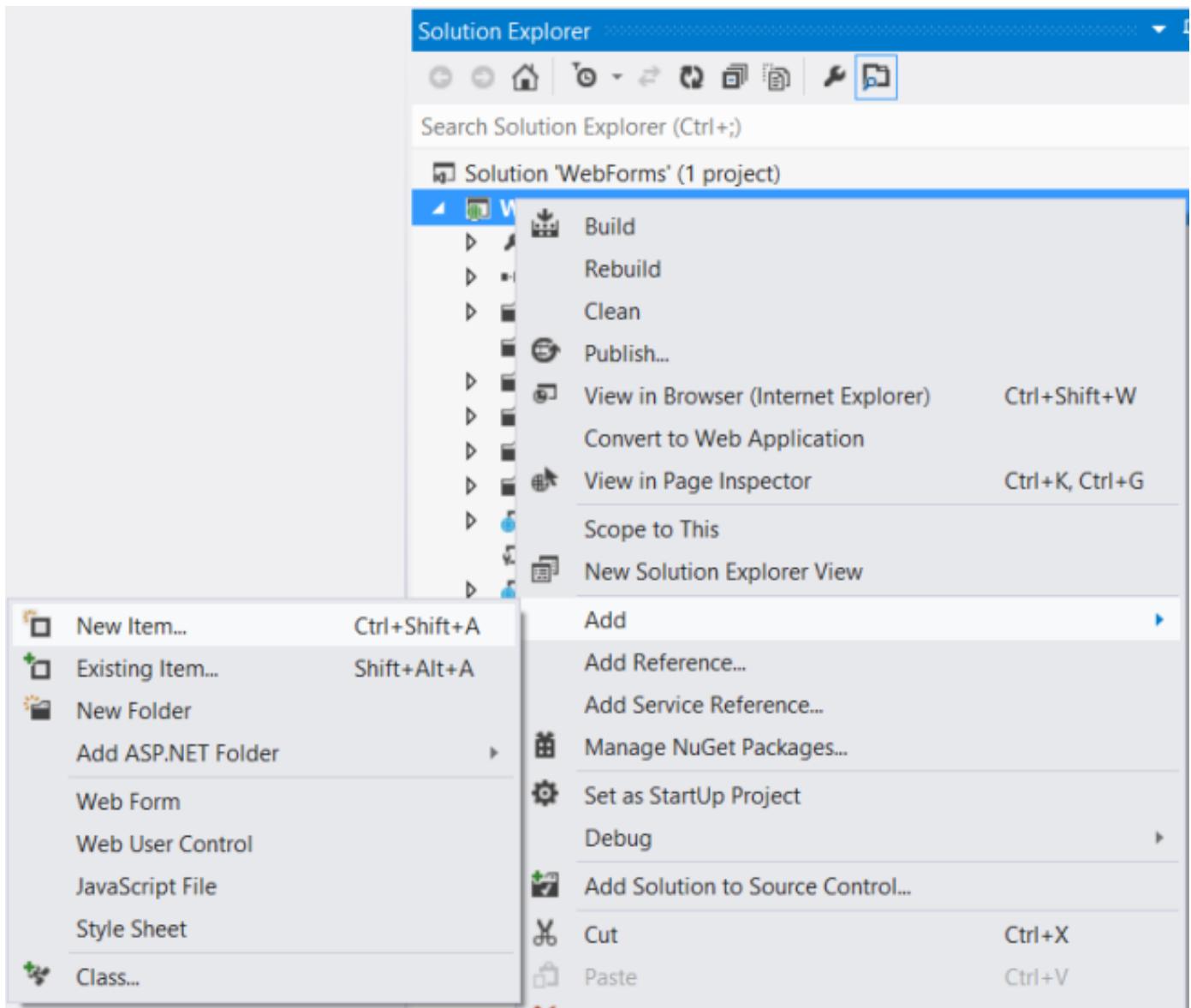


کنترلر و مدل اپلیکیشن را ایجاد کنید
 کلاس جدیدی با نام **Product** بسازید و خواص زیر را به آن اضافه کنید.

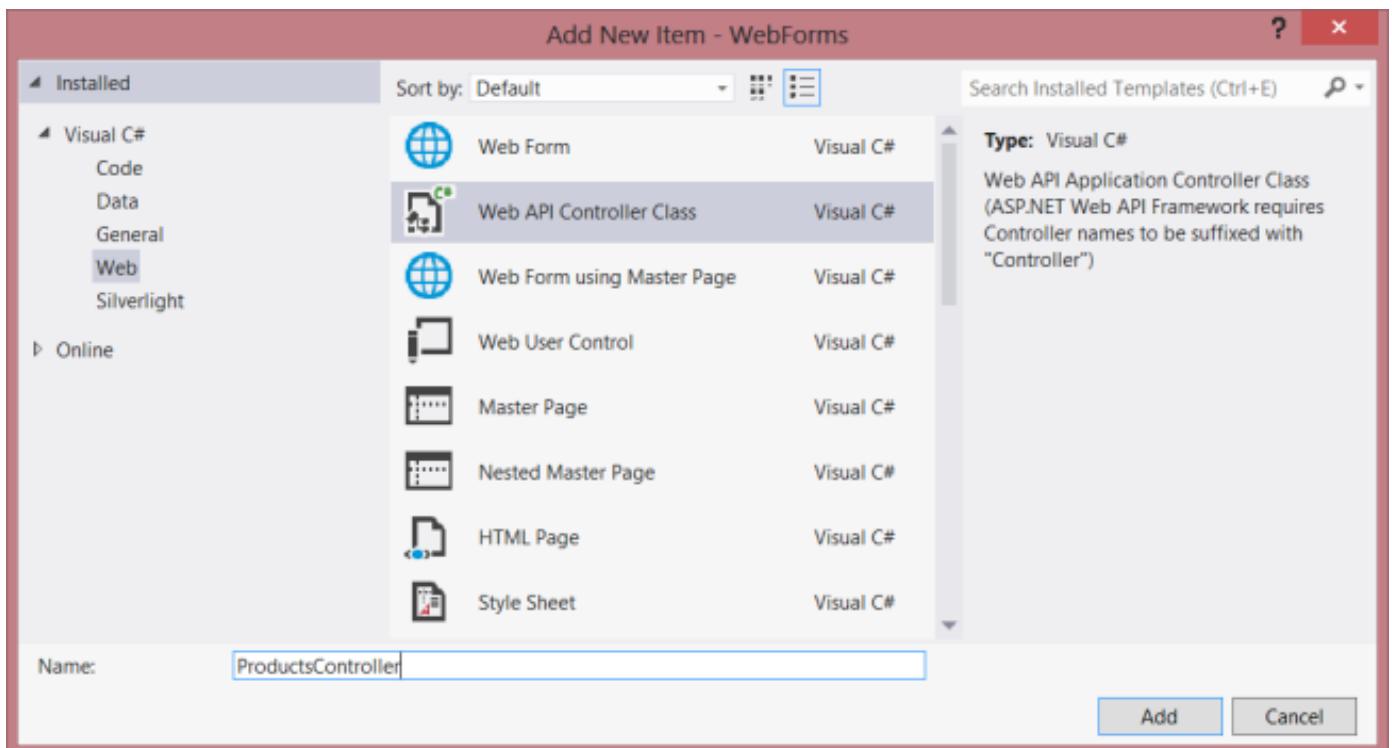
```
public class Product
{
    public int Id { get; set; }
    public string Name { get; set; }
    public decimal Price { get; set; }
    public string Category { get; set; }
```

}

همانطور که مشاهده می‌کنید مدل مثل نمایانگر یک محصول است. حال یک کنترلر Web API به پروژه اضافه کنید. کنترلرهای Web API در خواست‌های HTTP را به اکشن‌های آنها نگاشت می‌کنند. در پنجره Solution Explorer روی نام پروژه کلیک راست کنید و گزینه Add, New Item را انتخاب کنید.



در دیالوگ باز شده گزینه Web API Controller Class را از پانل سمت چپ کلیک کنید و نوع آیتم جدید را ProductsController را به "ProductsController" تغییر دهید و OK کنید.



کنترلر ایجاد شده شامل یک سری متد است که بصورت خودکار برای شما اضافه شده اند، آنها را حذف کنید و کد زیر را به کنترلر خود اضافه کنید.

```
namespace WebForms
{
    using System;
    using System.Collections.Generic;
    using System.Linq;
    using System.Net;
    using System.Net.Http;
    using System.Web.Http;

    public class ProductsController : ApiController
    {
        Product[] products = new Product[]
        {
            new Product { Id = 1, Name = "Tomato Soup", Category = "Groceries", Price = 1 },
            new Product { Id = 2, Name = "Yo-yo", Category = "Toys", Price = 3.75M },
            new Product { Id = 3, Name = "Hammer", Category = "Hardware", Price = 16.99M }
        };

        public IEnumerable<Product> GetAllProducts()
        {
            return products;
        }

        public Product GetProductById(int id)
        {
            var product = products.FirstOrDefault((p) => p.Id == id);
            if (product == null)
            {
                throw new HttpResponseException(HttpStatusCode.NotFound);
            }
            return product;
        }

        public IEnumerable<Product> GetProductsByCategory(string category)
        {
            return products.Where(
                (p) => string.Equals(p.Category, category,
                    StringComparison.OrdinalIgnoreCase));
        }
    }
}
```

```
}
```

کنترلر جاری لیستی از محصولات را بصورت استاتیک در حافظه محلی نگهداری می‌کند. متدهایی هم برای دریافت لیست محصولات تعریف شده اند.

اطلاعات مسیریابی را اضافه کنید

مرحله بعدی اضافه کردن اطلاعات مسیریابی (routing) است. در مثال جاری می‌خواهیم آدرس هایی مانند "api/products" به کنترلر Web API نگاشت شوند. فایل **Global.asax** را باز کنید و عبارت زیر را به بالای آن اضافه نمایید.

```
using System.Web.Http;
```

حال کد زیر را به متد Application_Start اضافه کنید.

```
RouteTable.Routes.MapHttpRoute(
    name: "DefaultApi",
    routeTemplate: "api/{controller}/{id}",
    defaults: new { id = System.Web.Http.RouteParameter.Optional } );
```

برای اطلاعات بیشتر درباره مسیریابی در Web API به [این لینک](#) مراجعه کنید.

دربیافت اطلاعات بصورت آژاکسی در کلاینت

تا اینجا شما یک API دارید که کلاینت‌ها می‌توانند به آن دسترسی داشته باشند. حال یک صفحه HTML خواهیم ساخت که با استفاده از jQuery سرویس را فراخوانی می‌کند. صفحه Default.aspx را باز کنید و کدی که بصورت خودکار در قسمت Content تولید شده است را حذف کرده و کد زیر را به این قسمت اضافه کنید:

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.Master"
   AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="WebForms._Default" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
</asp:Content>

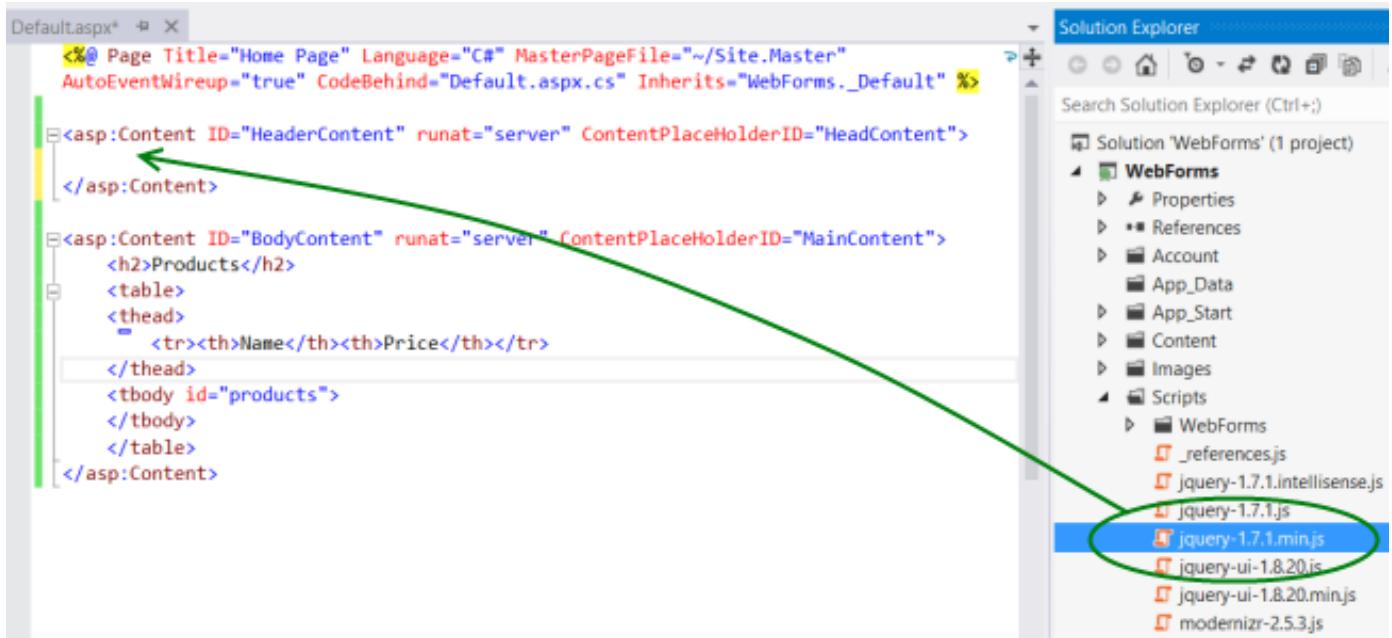
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
  <h2>Products</h2>
  <table>
    <thead>
      <tr><th>Name</th><th>Price</th></tr>
    </thead>
    <tbody id="products">
    </tbody>
  </table>
</asp:Content>
```

حال در قسمت HeaderContent jQuery را ارجاع دهید.

```
<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
  <script src="Scripts/jquery-1.7.1.min.js" type="text/javascript"></script>
</asp:Content>
```

همانطور که می‌بینید در مثال جاری از فایل محلی استفاده شده است اما در اپلیکیشن‌های واقعی بهتر است از CDN‌ها استفاده کنید.

نکته: برای ارجاع دادن اسکریپت‌ها می‌توانید بسادگی فایل مورد نظر را با drag & drop به کد خود اضافه کنید.



زیر تگ `<script>` اسکریپت زیر را اضافه کنید.

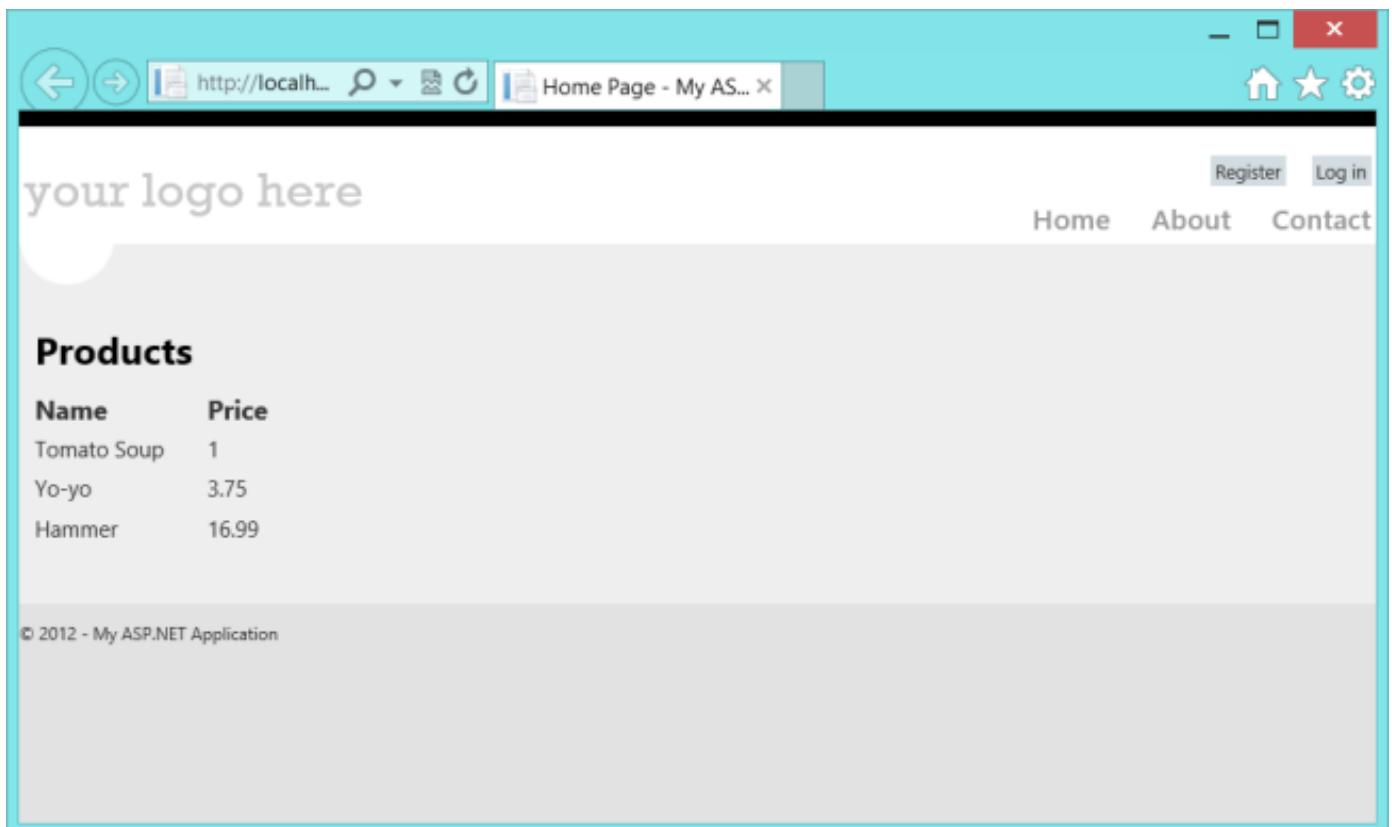
```
<script type="text/javascript">
    function getProducts() {
        $.getJSON("api/products",
            function (data) {
                $('#products').empty(); // Clear the table body.

                // Loop through the list of products.
                $.each(data, function (key, val) {
                    // Add a table row for the product.
                    var row = '<td>' + val.Name + '</td><td>' + val.Price + '</td>';
                    $('<tr/>', { text: row }) // Append the name.
                        .appendTo($('#products'));
                });
            });
    }

    $(document).ready(getProducts);
</script>
```

هنگامی که سند جاری (document) بارگذاری شد این اسکریپت یک درخواست آژاکسی به آدرس "api/products" ارسال می‌کند. سرویس ما لیستی از محصولات را با فرمات JSON بر می‌گرداند، سپس این اسکریپت لیست دریافت شده را به جدول HTML اضافه می‌کند.

اگر اپلیکیشن را اجرا کنید باید با نمایی مانند تصویر زیر مواجه شوید:



نظرات خوانندگان

نویسنده: ارشیا

تاریخ: ۱۳۹۲/۱۱/۰۵ ۱۳:۲۷

اگر بخواهیم زمانی که برای فیلد price مقداری که وارد میکنند حتما نوع عددی یا اعشاری که شما در نظر گرفتید باشه ، باید چه کدی را اضافه کنیم . تا زمانی که مقدار عددی و یا اعشاری وارد نکند اجازه اضافه کردن سطر دیگر را ندهد . امکان چنین کاری وجود دارد ؟

نویسنده: آرمین ضیاء

تاریخ: ۱۳۹۲/۱۱/۰۵ ۱۹:۵۰

می تونید از جاوا اسکریپت و [Remote Validation](#) استفاده کنید.

نویسنده: ارشیا

تاریخ: ۱۳۹۲/۱۱/۰۶ ۱۱:۲۲

امکانش هست لینک مثالی در این باره بفرمایید یا نمونه برنامه ای ؟

نویسنده: محسن خان

تاریخ: ۱۳۹۲/۱۱/۰۶ ۱۲:۵

مفاهیم [اعتبارسنجی در MVC](#) با [Web Api](#) تقریبا یکی است.

نویسنده: مهرداد

تاریخ: ۱۳۹۳/۰۲/۱۵ ۱۶:۴۵

- آیا برای عملیات CRUD می توان از آن استفاده کرد؟ اضافه ، حذف ، آپدیت؟ (مثال؟)
 - آیا استفاده از web api جهت عملیات CRUD بجای استفاده از MS AJAX بهتر است ؟
 - برای اینکه فقط یوزرهای سایت به این web api دسترسی داشته باشد ، کد خاصی باید اضافه شود ؟
 - در نهایت سوال آخر : اگر بخواهیم تمام عملیات CRUD سایت(ASP.NET Web forms) را با api انجام دهیم کار درستی است ؟
- بسیار مشکرم

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۲/۱۵ ۱۷:۲۵

- بله. [گروه Web API](#) و EF را در سایت پیگیری کنید.
- Web API یک بحث سمت سرور است. به آن به زبان ساده به چشم یک وب سرویس مدرن نگاه کنید. برای نمونه بجای وب متد های استاتیک صفحات aspx یا فایل های ashx یا asmx و حتی سرویس های WCF از نوع REST و امثال آن، بهتر است از API استفاده کنید.
- برای نمونه پایه مباحثی مانند Forms Authentication در اینجا هم کاربرد دارد (البته این یک نمونه است).
- برای کار با Web API الزاما نیازی به ASP.NET ندارید (نه وب فرم ها و نه MVC)؛ به هیچکدام از نگارش های آن. سمت کاربر آن AngularJS و سمت سرور آن Web API باشد. کار می کند. (اهمیت این مساله در اینجا است که الان می شود یک فریم ورک جدید توسعه دی بر نامه های وب را کاملا مستقل از وب فرم ها و MVC طراحی کرد)

وقتی یک Web API می‌سازید بهتر است صفحات راهنمایی هم برای آن در نظر بگیرید، تا توسعه دهندگان بدانند چگونه باید سرویس شما را فراخوانی و استفاده کنند. گرچه می‌توانید مستندات را بصورت دستی ایجاد کنید، اما بهتر است تا جایی که ممکن است آنها را بصورت خودکار تولید نمایید.

بدین منظور فریم ورک ASP.NET Web API کتابخانه‌ای برای تولید خودکار صفحات راهنمایی در زمان اجرا (run-time) فراهم کرده است.

ASP.NET Web API

Home

ASP.NET Web API Help Page

Introduction

This API enables CRUD operations on a set of products.

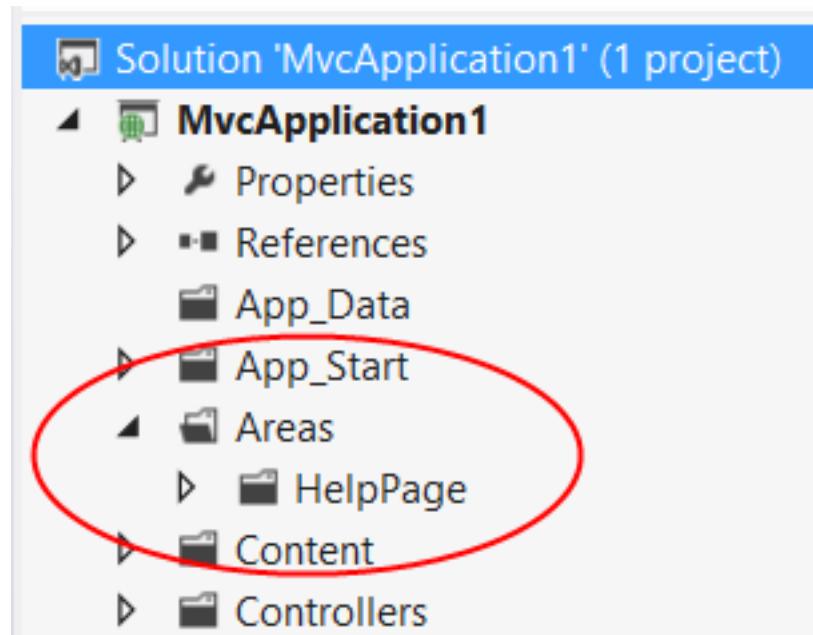
Products

API	Description
GET api/Products	Returns a list of products.
GET api/Products/{id}	Finds a product by ID.
POST api/Products	Creates a new product entity.

API ایجاد صفحات راهنمایی

برای شروع ابتدا ابزار [ASP.NET and Web Tools 2012.2 Update](#) را نصب کنید. اگر از ویژوال استودیو 2013 استفاده می‌کنید این ابزار بصورت خودکار نصب شده است. این ابزار صفحات راهنمایی را به قالب پروژه‌های ASP.NET Web API اضافه می‌کند.

یک پروژه جدید از نوع ASP.NET MVC Application بسازید و قالب Web API را برای آن انتخاب کنید. این قالب پروژه کنترلری بنام ValuesController را بصورت خودکار برای شما ایجاد می‌کند. همچنین صفحات راهنمایی API هم برای شما ساخته می‌شوند. تمام کد مربوط به صفحات راهنمایی در قسمت Areas قرار دارند.



اگر اپلیکیشن را اجرا کنید خواهید دید که صفحه اصلی لینکی به صفحه راهنمای API دارد. از صفحه اصلی، مسیر تقریبی /Help خواهد بود.



این لینک شما را به یک صفحه خلاصه (summary) هدایت می‌کند.

API	Description
GET api/Values	Documentation for 'Get'.
GET api/Values/{id}	Documentation for 'Get'.
POST api/Values	Documentation for 'Post'.
PUT api/Values/{id}	Documentation for 'Put'.
DELETE api/Values/{id}	Documentation for 'Delete'.

نمای این صفحه در مسیر Areas/HelpPage/Views/Help/Index.cshtml قرار دارد. می‌توانید این نما را ویرایش کنید و مثلاً قالب، عنوان، استایل‌ها و دیگر موارد را تغییر دهید.

بخش اصلی این صفحه متشکل از جدولی است که API‌ها را بر اساس کنترلر طبقه بندی می‌کند. مقادیر این جدول بصورت خودکار و توسط اینترفیس **IApiExplorer** تولید می‌شوند. در ادامه مقاله بیشتر درباره این اینترفیس صحبت خواهیم کرد. اگر کنترلر جدیدی به API خود اضافه کنید، این جدول بصورت خودکار در زمان اجرا بروز رسانی خواهد شد.

ستون "API" متدهای HTTP و آدرس نسبی را لیست می‌کند. ستون "Documentation" مستندات هر API را نمایش می‌دهد. مقادیر این ستون در ابتداء تنها **placeholder-text** است. در ادامه مقاله خواهید دید چگونه می‌توان از توضیحات XML برای تولید مستندات استفاده کرد.

هر API لینکی به یک صفحه جزئیات دارد، که در آن اطلاعات بیشتری درباره آن قابل مشاهده است. معمولاً مثالی از بدن‌های درخواست و پاسخ هم ارائه می‌شود.

GET api/Values

Documentation for 'Get'.

Response Information

Response body formats

application/json, text/json

Sample:

```
[  
    "sample string 1",  
    "sample string 2",  
    "sample string 3"  
]
```

application/xml, text/xml

Sample:

```
<ArrayOfstring xmlns:i="http://www.w3.org/2001/XMLSchema-instance"  
    xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">  
    <string>sample string 1</string>  
    <string>sample string 2</string>  
    <string>sample string 3</string>  
</ArrayOfstring>
```

افزودن صفحات راهنمایی به پروژه ای قدیمی

می‌توانید با استفاده از NuGet Package Manager صفحات راهنمایی خود را به پروژه‌های قدیمی هم اضافه کنید. این گزینه مخصوصاً هنگامی مفید است که با پروژه ای کار می‌کنید که قالب آن Web API نیست.

از منوی Tools گزینه‌های Library Package Manager, Package Manager Console را انتخاب کنید. در پنجره [Console](#) فرمان زیر را وارد کنید.

Install-Package Microsoft.AspNet.WebApi.HelpPage

این پکیج اسembly‌های لازم برای صفحات راهنمایی را به پروژه اضافه می‌کند و نماهای MVC را در مسیر Areas/HelpPage می‌سازد.

اضافه کردن لینکی به صفحات راهنمایی باید بصورت دستی انجام شود. برای اضافه کردن این لینک به یک نمای Razor از کدی مانند لیست زیر استفاده کنید.

```
@Html.ActionLink("API", "Index", "Help", new { area = "" }, null)
```

همانطور که مشاهده می‌کنید مسیر نسبی صفحات راهنمایی "Help/API" می‌باشد. همچنین اطمینان حاصل کنید که ناحیه‌ها (Areas) بدرستی登録 شوند. فایل Global.asax را باز کنید و کد زیر را در صورتی که وجود ندارد اضافه کنید.

```
protected void Application_Start()
{
    // Add this code, if not present.
    AreaRegistration.RegisterAllAreas();
}

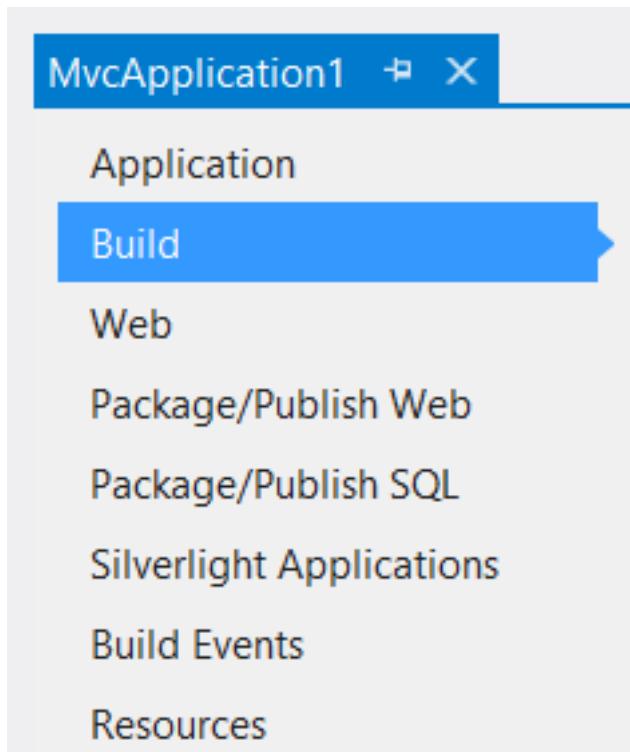
// ...
```

افزودن مستندات API

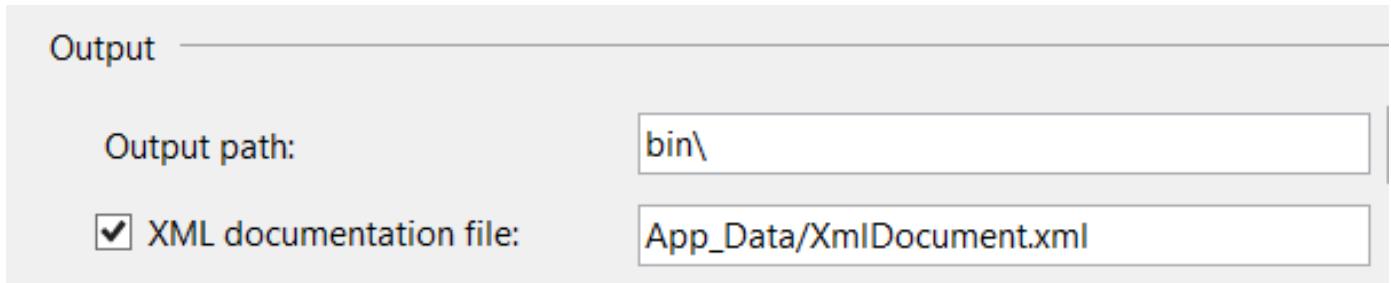
تصویرت پیش فرض صفحات راهنمایی از placeholder-text برای مستندات استفاده می‌کنند. می‌توانید برای ساختن مستندات از [توضیحات XML](#) استفاده کنید. برای فعال سازی این قابلیت فایل HelpPageConfig.cs را باز کنید و خط زیر را از حالت کامنت درآورید:

```
config.SetDocumentationProvider(new XmlDocumentationProvider(
    HttpContext.Current.Server.MapPath("~/App_Data/XMLDocument.xml")));
```

حال روی نام پروژه کلیک راست کنید و **Properties** را انتخاب کنید. در پنجره باز شده قسمت **Build** را کلیک کنید.



زیر قسمت **Output** گزینه **XML documentation file** را تیک بزنید و در فیلد روبروی آن مقدار "App_Data/XmlDocument.xml" را وارد کنید.



حال کنترلر ValuesController را از مسیر /Controllers/ValuesController.cs باز کنید و یک سری توضیحات XML به متدهای آن اضافه کنید. عنوان مثال:

```
/// <summary>
/// Gets some very important data from the server.
/// </summary>
public IEnumerable<string> Get()
{
    return new string[] { "value1", "value2" };
}

/// <summary>
/// Looks up some data by ID.
/// </summary>
/// <param name="id">The ID of the data.</param>
public string Get(int id)
{
    return "value";
}
```

اپلیکیشن را مجدداً اجرا کنید و به صفحات راهنمایی بروید. حالا مستندات API شما باید تولید شده و نمایش داده شوند.

API	Description
GET api/Values	Gets some very important data from the server.
GET api/Values/{id}	Looks up some data by ID.

صفحات راهنمایی شما را در زمان اجرا از توضیحات XML استخراج می‌کنند. دقت کنید که هنگام توزیع اپلیکیشن، فایل XML را هم منتشر کنید.

توضیحات تکمیلی
صفحات راهنمایی توسط کلاس **ApiExplorer** تولید می‌شوند، که جزوی از فریم‌ورک ASP.NET Web API است. به ازای هر API کلاس یک **ApiDescription** دارد که توضیحات لازم را در بر می‌گیرد. در اینجا منظور از "API" ترکیبی از متدهای HTTP و مسیرهای نسبی است. عنوان مثال لیست زیر تعدادی API را نمایش می‌دهد:

GET /api/products
GET /api/products/{id}
POST /api/products

اگر اکشن‌های کنترلر از متد‌های متعددی پشتیبانی کنند، هر متد را بعنوان یک API مجزا در نظر خواهد گرفت. برای مخفی کردن یک API از ApiExplorer کافی است خاصیت **ApiExplorerSettings** را به اکشن مورد نظر اضافه کنید و مقدار خاصیت **IgnoreApi** آن را به `true` تنظیم نمایید.

```
[ApiExplorerSettings(IgnoreApi=true)]
public HttpResponseMessage Get(int id) { }
```

همچنین می‌توانید این خاصیت را به کنترلرهای اضافه کنید تا تمام کنترلر از ApiExplorer از مخفی شود.

کلاس **ApiExplorer** متن مستندات را توسط اینترفیس **IDocumentationProvider** دریافت می‌کند. کد مربوطه در مسیر `/Areas/HelpPage/XMLDocumentation.cs` قرار دارد. همانطور که گفته شد مقادیر مورد نظر از توضیحات XML استخراج می‌شوند. نکته جالب آنکه می‌توانید با پیاده سازی این اینترفیس مستندات خود را از منبع دیگری استخراج کنید. برای اینکار باید متد `SetDocumentationProvider` را هم فراخوانی کنید، که در `HelpPageConfigurationExtensions` تعریف شده است.

کلاس **ApiExplorer** بصورت خودکار اینترفیس **IDocumentationProvider** را فراخوانی می‌کند تا مستندات API‌ها را دریافت کند. سپس مقادیر دریافت شده را در خاصیت **Documentation** ذخیره می‌کند. این خاصیت روی آبجکت‌های **ApiDescription** و **ApiParameterDescription** تعریف شده است.

مطالعه بیشتر

[Adding a simple Test Client to ASP.NET Web API Help Page](#)
[Making ASP.NET Web API Help Page work on self-hosted services](#)
[Design-time generation of help page \(or client\) for ASP.NET Web API](#)
[Advanced Help Page customizations](#)

نظرات خوانندگان

نویسنده: سعید شیرزادیان
تاریخ: ۱۳۹۲/۱۱/۱۸ ۱۹:۵۵

سلام؛ می‌خواستم بدونم قابلیت فوق نیز بر روی پروژه‌های asp.net وب فرمز نیز فعال می‌گردد؟ با تشکر

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۱/۱۸ ۲۰:۳۵

[Enabling ASP.NET Web API Help Pages for ASP.NET Web Forms Applications](#)

داشتن URL های تمیز و با معنا یکی از ویژگی های یک سایت خوب هست و بهترین روش برای بازنویسی URL ها یا همان URL استفاده از امکانات توکار خود ASP.NET برای این کار است. در یک مثال کوچک برای WebForm ها این کار را بررسی خواهیم کرد.

ابتدا باید فضای نام System.Web.Routing را در فایل Global.asax اضافه کنیم . سپس توسط Application_Start در RouteTable.Routes.MapPageRoute میتوانیم الگوهای تبدیل URL های خود را بنویسیم.

```
protected void Application_Start(object sender, EventArgs e)
{
    RouteTable.Routes.MapPageRoute("Product", "Product/{Name}", "~/Product.aspx");
}
```

تابع RouteTable.Routes.MapPageRoute سه ورودی می‌گیرد. ورودی اول یک نام برای الگوی تبدیل ماست که در مثال بالا، "Product" نام گرفته است. ورودی دوم، الگوی تبدیل ماست که می‌گوید URL هایی به شکل {Name} را به صفحه‌ی Product.aspx بفرست و در صفحه‌ی Product.aspx میتوانیم مقدار {Name} را توسط Page.RouteData.Values["Name"] بخواهیم . ورودی سوم نیز صفحه‌ی مقصد می‌باشد.

با این الگو میتوانیم لینک هایی مثل "لیست/Product" ایجاد کنیم و در صفحه‌ای که میخواهیم اطلاعات محصولمان را نشان دهیم از که همان مقدار لیست را بر میگرداند، استفاده کنیم .

یک مثال واقعی‌تر: مثالی که در URLs های مقالات همین سایت هم از آن استفاده شده [البته از نوع MVC](#)

```
RouteTable.Routes.MapPageRoute("PostDetail", "Post/{pi}/{pt}", "~/PostShow.aspx");
```

این الگو یو آر الی مثل "این-یک-تست-است/Post/2" را به صفحه‌ی PostShow.aspx میفرستد و در صفحه‌ی مقصد میتوانیم توسط Page.RouteData.Values["pi"] که منظور از pi همون PostId یا کد مقاله ما است، خوانده و نمایش دهیم. دلیل ارسال قسمت {pt} یا همان "این-یک-تست-است" می‌باشد و از pt منظورمان PostTitle یا عنوان مقاله‌است که در SEO سایت تاثیر زیادی دارد.

نحوه استفاده از این تبدیل هم اینطور می‌تواند باشد.
ساخت لینک :

```
<a href='<%# string.Format("/Post/{0}/{1}", Eval("PostID"), Eval("PostTitle").ToString().Replace(" ", "-")) %>'>
```

وقتی روی لینک کلیک می‌شود، به طور مثال چنین آدرسی در قسمت آدرس مرورگر ظاهر می‌شود. دلیل استفاده از `Replace(" ", "-")` هم برای زیباتر شدن URL هست. چون فضای خالی در URL زیبا نیست.

چگونه-طراحان-وب-به-جهنم-می-روند؟ / 12/Post

و در صفحه‌ی PostShow.aspx به این نحو میتوانیم اطلاعات مقاله‌ی مورد نظر را بدست بیاریم:

```
int pi = int.Parse(Page.RouteData.Values["pi"].ToString());
Posts post = (from p in con.Posts where p.PostID == pi select p).FirstOrDefault();
```

یا حتی برای نشان دادن اطلاعات از این روش استفاده کنیم:

```
<asp:EntityDataSource ID="EntityDataSource1" runat="server"
    AutoGenerateWhereClause="True" ConnectionString="name=WebWorkEntities"
    DefaultContainerName="WebWorkEntities" EnableFlattening="False"
```

```
EntitySetName="Posts" EntityTypeFilter="Posts" Where="" Select="">
<WhereParameters>
    <asp:RouteParameter Name="PostID" RouteKey="pi" DbType="Int32"
        DefaultValue="0" />
</WhereParameters>
</asp:EntityDataSource>
```

منظورم استفاده از `asp:RouteParameter` در `<WhereParameters>` برای انتخاب مقاله است.
و برای زیباتر شدن کد نویسی، بهتر است الگوهای تبدیل را در یک تابع جدا بنویسیم:

```
protected void Application_Start(object sender, EventArgs e)
{
    RoutingSite(RouteTable.Routes);
}

public static void RoutingSite(RouteCollection route)
{
    route.MapPageRoute("PostDetail", "Post/{pi}/{pt}", "~/PostShow.aspx");
    route.MapPageRoute("RouteAbout", "About-Me", "~/About.aspx");
}
```

[مطالعه بیشتر](#) [ASP.NET 4.0 Web forms URL Routing](#)

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۱/۱۰ ۱۰:۴۶

با تشکر از شما. تنها حذف فضای خالی برای تولید اصطلاحا `slug` کافی نیست. متده `GenerateSlug` مطلب «[بهینه سازی برنامه های وب ASP.NET برای موتورهای جستجو \(SEO\)](#)» بهتر است مدنظر باشد.

نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۱۱/۱۰ ۱۶:۲۰

یک نمونه برنامه برای مثال [Test-Routing.zip](#)

نویسنده: مائدہ
تاریخ: ۱۳۹۲/۱۱/۱۴ ۱۹:۳۰

سوال من در مورد وب فرم هایی که توی فolder هستند هنوز پابرجاست
شکل آدرس دهی این وب فرمها چطوری میشه؟

نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۱۱/۱۴ ۱۹:۵۲

ممکنه بیشتر توضیح بدید! منظورتونو از فرم هایی که تو فolder هستند متوجه نمیشم. مثلا فرم های قسمت مدیریتی رو میگید ، پوشه Admin

نویسنده: مائدہ
تاریخ: ۱۳۹۲/۱۱/۱۴ ۲۰:۲۹

بله مثل همون پوشه ادمین
من کلا فرم هام رو توی یک پوشه جدا گذاشتم به اسم WebForms
حالا برای آدرس دهی اگه مثلا بنویسم ~/WebForms/Details/ ~ جواب نمیده اگه به صورت ~/Details/ ~ هم بنویسم بازم جواب نمیده!

نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۱۱/۱۴ ۲۱:۶

چرا جواب میده ، باید اینجوری بنویسید .

```
RouteTable.Routes.MapPageRoute("Folder", "ShowFolder/{Id}/{Name}", "~/Admin>Show.aspx");
```

در نمونه برنامه يه فolder ساختم و کاري که خواستيد و انجام دادم.

نویسنده: مائدہ
تاریخ: ۱۳۹۲/۱۱/۱۵ ۱۳:۸

ممnon
نم دقيقا همينطوری نوشتتم ولی بازم ارور 404 میده
الآن امتحان کردم در اين حالتی که شما نوشتيدي جواب میده اما اگه کل فرمها توی پوشه باشه جواب نمیده.
شما هم اگه اون فرم Default رو بذاريد توی پوشه ادمين میبینيد که جواب نمیده!
و يه سوال: چي کار کردید که وقتی برنامه اجرا ميشه اسم صفحه اول (Default) رو توی آدرس بار نشون نمیده؟

(پ.ن: همینطوری تغییرات کدها رو هم بنویسید متوجه میشم نیازی به نمونه برنامه نیست. با تشکر)

نویسنده: محسن خان
تاریخ: ۱۴:۳۷ ۱۳۹۲/۱۱/۱۵

زمانیکه فرم default رو در یک پوشه‌ی دیگر گذاشتید، برای اون یک route جدا تعریف کردید؟ الان آیا برنامه اطلاع داره زمانیکه به مسیر ریشه سایت وارد شد باید به کجا هدایت بشه؟

```
routes.MapPageRoute("defaultRoute", "{*value}", "~/folder/default.aspx");
```

نویسنده: رضا منصوری
تاریخ: ۱۷:۴۸ ۱۳۹۲/۱۱/۱۵

باید برای تعریف لینک‌ها اینگونه عمل کنید.

```
<a href='<%# string.Format("../ShowFolder/{0}/{1}", Eval("Id"), Eval("Name").ToString().Replace(" ", "-")) %>'>%# Eval("Name")%</a>
```

.. باعث میشه یک فolder به عقب برگردنه و اینجوری درست کار میکنه.
در مورد اجرا شدن صفحه default هم دوستمون اشاره کردند.

بازم نمونه برنامه ضمیمه کردم. فرم default رو تو پوشه‌ی ادمین بردم. وقتی برنامه هم اجرا میشه صفحه default در داخل پوشه‌ی ادمینو نشون میده.

نویسنده: مائدہ
تاریخ: ۲۲:۳۵ ۱۳۹۲/۱۱/۱۶

بازم ممنون. فقط یه سوال دیگه. فرض کنید من یه چیزی به آدرس اصلی صفحه اضافه کرم. حالا میخوام به این آدرس جدید دوباره یه چیز دیگه اضافه کنم. باید چی کار کنم؟ اینجا وقتی یک یا چند تا چک لیست انتخاب میشه یا مثلا چیزی از لیست باکس‌ها انتخاب میشه عبارت مربوطه اش به آدرس اضافه میشه. چطوری این کارو کرد؟ اینجوری که آدرس ثابت نیست و میتونه حالت‌های مختلفی داشته باشه! یا مثلا وقتی یک مورد انتخاب شده حذف میشه چطوری عبارت مربوطه اش از وسط آدرس حذف میشه؟!

نویسنده: وحید نصیری
تاریخ: ۰:۴۱ ۱۳۹۲/۱۱/۱۷

این نوع routing خاص سمت کلاینت برنامه‌های تک صفحه‌ای وب را با استفاده از کتابخانه‌ای مانند [path.js](#) می‌شود انجام داد.

مثال‌های آنرا به همراه سورس صفحات آن بررسی کنید.
نمونه آنرا در مطلب «[بیاده سازی دکمه «بیشتر» یا «اسکرول نامحدود» به کمک jQuery](#) در ASP.NET MVC» (کامنت آخر آن) پیاده سازی شده می‌توانید مشاهده و دریافت کنید. از این قابلیت در سایت جاری در حال استفاده است. مثلا اگر به قسمت مطالب در بالای صفحه مراجعه کنید، چنین آدرسی قابل مشاهده است:

<http://www.dotnettips.info/postsarchive #/page/1/date/desc>

در پایین صفحه اگر در اپ داون‌های مرتب سازی را تغییر دهید، نام فیلد یا صعودی و نزولی بودن آدرس تغییر می‌کنند. شماره صفحه نیز با هر بار کلیک بر روی دکمه‌ی بیشتر یکی اضافه می‌شود. همچنین این آدرس‌ها را می‌شود ذخیره و عیناً بازیابی کرد.

نویسنده: ابراهیم
تاریخ: ۱۲:۱۷ ۱۳۹۲/۱۱/۱۷

سلام: خیلی ممنون از مطلب مفیدتون.
من از دستورات route به شکل زیر استفاده میکنم

```
RouteTable.Routes.MapPageRoute("Post", "Post/{Id}/{title}", "~/showpost.aspx");
```

این دستوارت به خوبی کار میکند و مشکلی هم نیست. اما مسئله ای که وجود داره اینه که وقتی کاربر بخواهد url را دستکاری کنه و مثلا عدد id را تغییر بد، title در url تغییر نمیکنه. من سایت های زیادی رو دیدم که این ایراد رو دارن. بسیاری از سایت های خبری. اما سایت جاری چنین مشکلی نداره. یعنی وقتی شما id را در url تغییر میدید، عنوان پست در url هم تغییر میکنه. لطفا در مورد چگونگی پیاده سازی این روش توضیح بدید.

متشرک

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۷ ۱۲:۳۸

```
var id = Page.RouteData.Values["id"];
var title = Page.RouteData.Values["title"];

var blogPost = _postsService.FindPost(id);
if(blogPost == null)
{
    Response.RedirectPermanent("~/notfound");
}

if(blogPost.Title != title)
{
    Response.RedirectPermanent("~/post/" + id + "/" + blogPost.Title);
}
```

سیستم مسیریابی، اطلاعات id و title وارد شده توسط کاربر را به شما می دهد. زمانیکه می خواهید مطلب را نمایش دهید، این title دریافتی را با title واقعی مطلب مانند کدهای فوق مقایسه کنید. (البته بهتر است از [GenerateSlug](#) استفاده کنید و عنوان نهایی را بر این اساس مقایسه و اصلاح کنید)

نویسنده: ابراهیم
تاریخ: ۱۳۹۲/۱۱/۱۷ ۲۳:۳۶

سلام. سوال دیگه ای هم داشتم.

چطور جلوی دسترسی کاربران به آدرس اصلی فایل رو بگیریم؟
مثلا در مثال زیر :

```
route.MapPageRoute("PostDetail", "Post/{pi}/{pt}", "~/PostShow.aspx");
```

کاربر نتونه به فایل postshow.aspx به صورت مستقیم دسترسی داشته باشه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۸ ۱:۲۷

در ASP.NET MVC برای اینکه کسی نتواند فایل های View را مستقیما فراخوانی کند، دسترسی به آنها در فایل web.config [کلا بسته شده](#)

```
<system.web>
    <httpHandlers>
        <remove verb="*" path="*.aspx" />
        <add path="*.aspx" verb="*" type="System.Web.HttpNotFoundHandler" />
    </httpHandlers>
</system.web>

<system.webServer>
    <validation validateIntegratedModeConfiguration="false" />
    <handlers>
        <add name="BlockViewHandler" path="*.aspx" verb="*" preCondition="integratedMode"
```

```
type="System.Web.HttpNotFoundHandler" />
</handlers>
</system.webServer>
```

نویسنده: ابراهیم
تاریخ: ۲:۵ ۱۳۹۲/۱۱/۱۸

جسارت منظورم asp.net webform بود

نویسنده: وحید نصیری
تاریخ: ۲:۸ ۱۳۹۲/۱۱/۱۸

هدفم طرح یک ایده بود. امتحان کردید؟ `HttpNotFoundHandler` مشترک است بین MVC و وب فرم‌ها)

نویسنده: ابراهیم
تاریخ: ۲:۱۳ ۱۳۹۲/۱۱/۱۸

نکته‌ای که الان یادم اومد اینه که در برنامه من، کاربران به صفحات اصلی سایت بصورت routing دسترسی دارند و در قسمت داشبورد به صورت ادرس مستقیم.
چطور میشه در `webform` این دو حالت رو در یک سایت از هم تفکیک کرد؟
متشرکم

نویسنده: وحید نصیری
تاریخ: ۲:۱۷ ۱۳۹۲/۱۱/۱۸

فایل وب کافیگی را که دسترسی به `.aspx` اعلام می‌کند، داخل همان پوشه مدنظر قرار دهید. (در MVC هم دقیقاً به همین نحو عمل شده است. یک فایل `web.config` دومی را در پوشه `views` قرار داده‌اند)

نویسنده: ابراهیم
تاریخ: ۲:۳ ۱۳۹۲/۱۱/۱۹

سلام

دستورات ذکر شده رو در فایل `web.config` قرار دادم. دسترسی به فایلهای `.aspx` قطع شد و فقط میشد از طریق routing دسترسی داشت. اما مشکلی که بوجود اومد این بود که وقتی نام سایت رو در مرورگر وارد میکردم، خطای زیر رو دریافت کردم

The resource cannot be found.

Description: HTTP 404. The resource you are looking for (or one of its dependencies) could have been removed, had its name changed, or is temporarily unavailable. Please review the following URL and make sure that it is spelled correctly.

Requested URL: /Default.aspx

به ذهنم رسید که در فایل `global.asax` دستور زیر رو وارد کنم

```
routes.MapPageRoute("Default", "", "~/default.aspx", False)
```

وقتی این کارو کردم دیدم که مشکل دسترسی به فایل `default.aspx` حل شد اما دسترسی مستقیم به فایلهای `.aspx` که در فولدرهای دیگر بودند دچار مشکل شد در وب جستجو کردم و به لینکهای روبرو رسیدم ([+](#))
اما نتیجه‌ای نداد.

نویسنده: محسن خان
تاریخ: ۱۰:۱۱ ۱۳۹۲/۱۱/۱۹

دستورات not found handler را در web.config نوشته اصلی نوشته است. چون وب کانفیگ جدید در پوشه‌ای که می‌خواهد؟ در ریشه اصلی روی تمام زیر پوشه‌ها به علت ارت بری تاثیر دارد.

نویسنده: ابراهیم
تاریخ: ۲۱:۴۳ ۱۳۹۲/۱۱/۱۹

چون اون فایلها در ریشه سایت هستند، بنابراین در web.config ریشه اصلی نوشتم
چه کاری می‌تونم بکنم؟

نویسنده: زیلابی
تاریخ: ۱۵:۹ ۱۳۹۳/۰۲/۳۰

سلام؛ تمام کارهایی که شما نوشته‌ید را نوشتم و انجام دادم. کار می‌کنند ولی دو تا مشکل داره:
۱- استیل‌های صفحه رو بار گذاری نمی‌کنند. یعنی CSS‌هایی که نوشتم و در پوشه‌ای به نام CSS ذخیره کردم رو نمی‌اره. البته من از مستر استفاده نمی‌کنم. استیل‌های مستر می‌دانم اما استیل‌های شخصی اون صفحه نه.
۲- مسیر سایتم دارای یک اسم به نام publicity است:

....//www.site.com/publicity

که وقتی آدرس‌ها رو درست می‌کنم این شکل می‌شود:
www.site.com/post/1325 publicity و حذف می‌شود از آدرس که باعث می‌شود برنامه ارور بده که همچین آدرسی نیست.

حالا باید چکار کنم؟

نویسنده: وحید نصیری
تاریخ: ۱۷:۲۶ ۱۳۹۳/۰۲/۳۰

در حالت کلی در وب فرم‌ها، برای مسیردهی سازگار با Routing باید از ResolveUrl استفاده کنید.
برای اسکریپت‌ها:

```
<script type='text/javascript' src='<%= ResolveUrl("~/Scripts/test.js") %>'></script>
```

برای شیوه‌نامه‌ها:

```
<link rel="stylesheet" href="<%= ResolveUrl("~/myStyleSheet.css")%>" type="text/css" />
```

متدهای ResolveUrl به صورت خودکار مسیر صحیح را تولید می‌کند.

اما ... این روش کار کردن صحیح نیست. چون به زودی به تعداد زیادی فایل اسکریپت و CSS لینک داده شده در صفحه می‌رسید. برای یکی کردن آن‌ها یا از ScriptManager استفاده کنید (برای اسکریپت‌ها) و یا از روش‌های bundling & minification که با فایل‌های CSS و JS سازگار است. مسیریابی‌ها را هم به صورت خودکار تصحیح می‌کند.

نویسنده: زیلابی
تاریخ: ۱۰:۴۰ ۱۳۹۳/۰۲/۳۱

تشکر.

اما من روش bundling & minification رو متوجه نشدم. مطلب آموزشیش رو از کجا باید بخونم؟
منظورم مطلب آموزشی فارسی است.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۳۱ ۱۰:۵۰

روش‌های زیادی برای دسته بندی، فشرده و یکی سازی فایل‌های CSS و JS هست؛ منجمله:

- بهینه سازی فایل‌های js و css در برنامه‌های ASP.NET با استفاده از Combres - قسمت اول
- فشرده سازی فایل‌های JavaScript و CSS بصورت خودکار توسط MS Ajax Minifier
- Adding Bundling and Minification to Web Forms
- Building a basic Web Forms application using Visual Studio 2013
- Bundling and Minification in ASP.NET 4.5

نویسنده: شایان
تاریخ: ۱۳۹۳/۰۴/۲۵ ۱۱:۳۰

با سلام و تشکر بابت مقاله خوبتون .

من یه صفحه به اسم add.aspx تو فolder Admin سایتم که با asp.net web form ایجاد شده دارم و میخوام وقتی کاربر آدرس را تو آدرس بار وارد کرد بطور اتوماتیک وارد اون صفحه(Add.aspx) بشه. چطوری میتونم اینکارو بکنم ؟ با تشکر

نویسنده: رضا منصوری
تاریخ: ۱۳۹۳/۰۴/۲۵ ۱۲:۵

سلام به این صورت در فایل Global.asax.cs

```
// using System.Web.Routing; //Must Be Added
public static void RoutingSite(RouteCollection route)
{
    // تعريف Route
    route.MapPageRoute("Admin", "Admin", "~/admin/add.aspx");
}
protected void Application_Start(object sender, EventArgs e)
{
    RoutingSite(RouteTable.Routes);
}
```

نویسنده: شایان
تاریخ: ۱۳۹۳/۰۴/۲۶ ۱۳:۱۵

ممnon بابت جوابتون . من اینکارو انجام میدم ولی وقتی کاربر آدرس http://mysite/Admin را تو آدرس بار وارد میکنه اونو به دایرکتوری فolder Admin که شامل چندین صفحه است میبینه و باز کاربر باید خودش رو لینک Add.aspx کلیک کنه تا بتونه وارد اون صفحه بشه .. امکانش هست کاربر اتوماتیک وارد صفحه Add.aspx بشه ؟

نویسنده: رضا منصوری
تاریخ: ۱۳۹۳/۰۴/۲۶ ۱۳:۴۳

خواهش میکنم ، این مشکل به خاطر اینه که وقتی شما آدرس Admin را در مرورگر تایپ میکنید قبل از اینکه برنامه به کدهای روتنینگ برسه برنامه Admin رو پوشه میدونه و فکر میکنه شما میخواهید پوشه‌ی ادمین رو باز کنید و این کارو انجام میده و به کدهای روتنینگ نمیرسه. برای حل این مشکل باید نام روتنینگ را به نام دیگه ای تغییر بدید مثل این

```
public static void RoutingSite(System.Web.Routing.RouteCollection route)
{
    // تعريف Route
    route.MapPageRoute("Admin", "AdminAdd", "~/admin/add.aspx");
}
protected void Application_Start(object sender, EventArgs e)
{
    RoutingSite(System.Web.Routing.RouteTable.Routes);
```

}

با این روتینگ وقتی شما آدرس <http://mysite/AdminAdd> را میزند صفحه‌ای add.aspx در پوششی ادمین رو اجرا میکنه . یه نمونه برنامه برآتون ساختم و همین کاریو که خواستید رو انجام دادم. [WebSite1-5077f441ae044a3ca59b8d34c9c42237.zip](#)

نویسنده: زیلایی
تاریخ: ۱۳۹۳/۰۵/۲۱ ۱۳:۵۸

سلام؛ وقتی این کد رو توی کد بیهایند مستر پیج برای گرفتن ID کاربر قرار میدم بهم ارور میده `Page.RouteData.Values("pi").ToString()`

چون مستر پیج خاصیتی به اسم page نداره. برای همین `page.RouteData` رو null برミگردونه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۲۱ ۱۴:۱۱

از `this.Request.RequestContext.RouteData` استفاده کنید.

نویسنده: رضا منصوری
تاریخ: ۱۳۹۳/۰۵/۲۱ ۱۴:۱۷

سلام . معمولا در `code behind` صفحات داخلی ID را میگیرند ، نه در مستر پیج و اینکه احتمالان اشتباه میکنید. `Page` یک متاد توکار هستش و همیشه وجود داره . اگر به null برミغورید احتمالان به خاطر استفاده از `ToString` هستش . اگه مقداری که وجود نداره رو ازش `ToString` بگیریم به ارور "object reference not set to an instance of an object" بر میغوریم . در این آموزش نمونه برنامه‌های زیادی وجود داره ، ازشون کمک بگیرید.

نویسنده: رضا منصوری
تاریخ: ۱۳۹۳/۰۵/۲۱ ۱۴:۲۸

نکته‌ی دیگه ای اینه که گرفتن مقدار از یک `Route` به این صورت هستش.

```
var id = Page.RouteData.Values["pi"].ToString();
```

شما از () استفاده کردید.

نویسنده: زیلایی
تاریخ: ۱۳۹۳/۰۵/۲۱ ۱۵:۲۴

آخه من با VB مینویسم و وقتی این کد رو مینویسم ارور میده که نباید از [] استفاده بشه.
کد رو به این تغییر دادم ولی بازم ارور دارم :

```
Me.Request.RequestContext.RouteData.Values("pi")
```

ارور اینه:

`Object reference not set to an instance of an object.`

نویسنده: رضا منصوری
تاریخ: ۱۳۹۳/۰۵/۲۱ ۱۵:۵۲

یه نمونه کد با زبان بسیار شیرین VB گذاشتم . ارور هم عرض کردم این ارور به خاطر این هستش که مقدار نال رو میخواهد بخونید . موفق باشید.

[WebApplication2-b172669ecf424c98846a0291a7ada1c9.zip](#)

نویسنده: حسین فرهودی
تاریخ: ۱۰:۲۳ ۱۳۹۳/۰۵/۲۸

سلام ممکنه منو راهنمایی کنید

```
RouteTable.Routes.MapPageRoute("Gallery", "{PageName In Database}", "~/Main.aspx");
```

من جدولی در دیتابیس دارم که نام صفحات رو ذخیره میکنم چگونه میتونم مقدار `PageName in Database` چک کنه اگه داخل اون جدول بود وارد صفحه main بشه

نویسنده: رضا منصوری
تاریخ: ۱۰:۴۱ ۱۳۹۳/۰۵/۲۸

سلام.

این روتینگ رو تعریف میکنید.

```
RouteTable.Routes.MapPageRoute("Gallery", "Page/{PageName}", "~/Main.aspx");
```

در صفحه‌ی Main.aspx

```
string pi = Page.RouteData.Values["PageName"] + "";  
Pages page = (from p in context.Pages where p.PageName == pi select p).FirstOrDefault();  
if (page == null)  
    Response.Redirect("/");  
else  
    نمایش اطلاعات //
```

نویسنده: حسین فرهودی
تاریخ: ۱۵:۴۴ ۱۳۹۳/۰۵/۲۸

سلام؛ ممنونم از پاسخ خوبتون. آیا امکان دارد بتوان در `Page/{PageName}` لغت `Page` را حذف کرد و بعد دومین که زده میشود چک کند؟

نویسنده: رضا منصوری
تاریخ: ۱۸:۸ ۱۳۹۳/۰۵/۲۸

این کار شدنی هستش که چنین روتینگی داشته باشیم

```
RouteTable.Routes.MapPageRoute("Gallery", "{PageName}", "~/Main.aspx");
```

ولی تمام آدرس‌ها به این صفحه ارجاع داده میشه و شما هر آدرسی که وارد کنید به صفحه‌ی ذکر شده میره و این مشخصه که سایتی که دارید مینویسید فقط همین صفحه رو نداره.
البته این نوع روتینگ کاربرد خاصی هم داره و من تو چند تا سایت که میخواستیم توسعه بدیم استفاده کردم به این صورت که صفحاتی که توسط گوگل ایندکس شده بود و نمیخواستیم از دست بدیم با این روتینگ چک میکردیم و صفحاتی که ایندکس بود رو به آدرس جدیدش میفرستادیم.

نویسنده: محمد
تاریخ: ۰:۷ ۱۳۹۳/۰۵/۲۹

سلام . من دقیقا مانند کدهای شما رو نوشتم ولی یه مشکلی هست زمانی که از لینکی مثل زیر استفاده میکنم

```
<a href="article/da/name">...</a>
```

چیزی که در نوار آدرس به وجود میاد تقریبا مثل زیر خواهد بود :

```
localhost://article/da/name
```

و زمانی که میخواه در این حالت به یه آدرس دیگه برم مثل زیر

```
<a href="about">...</a>
```

درست کار نمیکنه و مقدار About را جایگزین آخرین مقدار آدرس قبلی میکنه . مثلا :

```
localhost://article/da/about
```

منظورم اینه که به روت بر نمیگردد . باید چکار کرد برای حل این مشکل

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۵/۲۶

کمی بالاتر توضیح دادن ... در حالت کلی در وب فرمها، برای مسیردهی سازگار با Routing باید از `ResolveUrl` استفاده کنید.

نویسنده: آریا
تاریخ: ۱۳۹۳/۰۸/۰۷ ۱۵:۱۰

پارامترهای صفحات من از نوع رشته هستند نه عدد بطور مثال :

```
localhost:56254/type/Brand/5 میکرولیتر با سرنگ اضافه
```

برای کلیک روی لینک تگ A در قیمت href با استفاده از `String.Format`) خط را جایگزین فاصله کردم اما اطلاعاتم را نمیخواند چون در دیتابیس فاصله جای خط وجود دارد. من برای خواندن اطلاعات از دیتالیست به همراه `EntityDataSource` استفاده کرده ام مانند :

```
<asp:EntityDataSource ID="EntityDataSource1" runat="server"
    ConnectionString="name=mainfardbEntities"
    DefaultContainerName="mainfardbEntities" EnableFlattening="False"
    EntitySetName="T_Product" Where="it.[CoName]=@category and it.[CatName]=@sub and
    it.[SubCatName]=@thi and it.[ProdName]=@det">
    <WhereParameters>
        <asp:QueryStringParameter DbType="String" Name="category" QueryStringField="category" />
        <asp:QueryStringParameter DbType="String" Name="sub" QueryStringField="sub" />
        <asp:QueryStringParameter DbType="String" Name="thi" QueryStringField="thi" />
        <asp:QueryStringParameter DbType="String" Name="det" QueryStringField="det" />
    </WhereParameters>
</asp:EntityDataSource>
```

چگونه میتوانم برای خواندن اطلاعات پارامتر خود را بصورت اولیه یعنی با فاصله بخوانم؟
منون از همه عزیزان

نویسنده: وحید نصیری

بجای خط از [Uri.EscapeDataString](#) باید استفاده کنید.

نویسنده: آریا
تاریخ: ۱۵:۵۶ ۱۳۹۳/۰۸/۰۷

ممnon از پاسخ شما
متاسفانه به نتیجه ای نرسیدم
ممکنه خودتون بیشتر توضیح بدین؟
باید بگم من از روش url rewrite استفاده میکنم

```
<rewrite url="~/type/(.+)/(.+)/(.+)/(.+)" to="~/type/Product-Details.aspx?category=$1&sub=$2&thi=$3&det=$4" />
```

من چطور میتونم به پارامتر ارسالی قبل از اینکه وارد شدن به شرط EntityDataSource دست پیدا کنم و اون رو تغییر بدم
البته من احساس میکنم مطلب گذاشته شده شما رو نتونستم درک بکنم

نویسنده: وحید نصیری
تاریخ: ۱۸:۰ ۱۳۹۳/۰۸/۰۷

هنگام تولید URL، اجزای قسمت‌های مختلف آن باید encode شوند؛ در غیر اینصورت عموماً قابل پردازش نخواهد بود.

```
var url = "<a href='http://site/page/" + Uri.EscapeDataString(item.Category) + "'>Item1</a>";
```

برای مثال فاصله تبدیل می‌شود به 20% و امثال آن. از متد [Uri.EscapeDataString](#) برای encode کردن اجزای مختلف URL باید استفاده کنید تا به نتیجه‌ی صحیحی برسید. زمانیکه این encoding انجام شد، طرف کدهای دات نتی که این اجزاء یا کوئری استرینگ‌ها را دریافت می‌کند، نیازی به هیچ تغییری یا اصلاحی نخواهد بود، چون به صورت خودکار قابلیت پردازش آدرس دریافتی را دارد (در پشت صحنه از متد [HttpUtility.UrlDecode](#) استفاده می‌کند). برای نمونه مقادیر دریافتی [Page.RouteData.Values](#) را بررسی کنید.

نویسنده: هادی
تاریخ: ۱۰:۸ ۱۳۹۳/۰۸/۲۴

سلام
واسه من تصاویر سایت هم نشان داده نمیشه ، برای تصاویر هم باید همین کار رو بکنم؟

نویسنده: وحید نصیری
تاریخ: ۱۰:۳۷ ۱۳۹۳/۰۸/۲۴

```
<img src='<%=ResolveUrl("~/images/brain.png")%>' />
<asp:ImageButton ID="ImageButton1" ImageUrl "~/images/brain.png" runat="server" />
<asp:Image runat="server" ID="imgHelp" ImageUrl "~/Images/myImage.jpg" />
```

نویسنده: هادی
تاریخ: ۱۱:۴ ۱۳۹۳/۰۸/۲۴

ممnon
حالا فکر کنید من مطالب کاربر که حاوی تصاویر نیز هست در دیتابیس ذخیره کرده ام و می‌خوام در صفحه نمایش بدم ، حالا چجور میتونم تصاویری که تو مطلب هست که در زمان اجرا در صفحه ریخته میشه نشون بدم؟
ببخشید یک سوال دیگه ، اصلا میشه گفت فقط فایلهایی که به پسوند aspx ختم میشنند رو از دسترسی مستقیم خارج کن و بقیه

فایلها آزاد باشند؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۲۴ ۱۱:۵۲

- این نوع تصاویر را بجای آدرس نسبی، با آدرس مطلق ذخیره کنید.
- از برای ویرایش و اصلاح `src` در حین رندر نهایی استفاده کنید.
- امکان خارج کردن منابع از سیستم URL rewrite وجود دارد :

```
<rule name="RewriteUserFriendlyURL1" stopProcessing="true">
  <match url="^([^\+])/?$" />
  <conditions>
    <add input="{URL}" negate="true" pattern=".png$" />
  </conditions>
</rule>
```

نویسنده: هادی
تاریخ: ۱۳۹۳/۰۸/۲۶ ۱۳:۳

با سلام؛ من از این روش از تکنیک ای جکس استفاده می‌کنم، ولی نمیدونم الان که Url‌ها را با تکنیک شما بازنویسی کردم چگونه توابع رو فراخوانی کنم؟ به وب سرویس دسترسی دارم ولی تابع داخلش اجرا نمیشه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۲۶ ۱۳:۱۱

از اینجا شروع کنید «نحوه استفاده از افزونه Firebug برای دیبگ برنامه‌های ASP.NET» مبتنی بر jQuery + Ajax و دریافت خطای 401 Unauthorized در حین عملیات ASP.NET FriendlyUrls

نویسنده: موسوی
تاریخ: ۱۳۹۳/۰۹/۲۰ ۱۱:۳۹

با سلام؛ وسایتم خوب کار میکنه فقط وقتی این قسمت که به جای فضای خالی _ بزار رو گذاشت مشکل پیدا کردم چطور درستش کنم.

```
<asp:HyperLink ID="HyperLink1" runat="server" NavigateUrl='<%# Eval("NewsID", "ShowNews.aspx?id={0}")+  
Eval("NewsTitle", "&_title={0}")%>.ToString().Replace(" ", "_")%>'
```

این خطهای اضافی دنباله خبر ماله چیه که درست شده اخه من برای درج خبر کاراکتر اضافی وارد نکردم که؟ کلا چرا بعد از خبر هام این قدر ----- میزاره

http://localhost:28625>ShowNews.aspx?id=25&_title=%D8%A2%D8%AE%D8%B1%DB%8C%D9%86_%D8%AA%D8%B5%D8%A7%D9%88%DB%8C%D8%B1_%D8%A7%D8%B2_%22%D9%85%D8%B1%D8%AA%D8%B6%DB%8C_%D9%BE%D8%A7%D8%B4%D8%A7%DB%8C%DB%8C%22__

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۲۰ ۱۱:۴۹

- از روش‌های ابداعی برای encoding رشته‌ها در وب استفاده نکنید. کمی بالاتر توضیح دادم که باید از [Uri.EscapeDataString](#) استفاده کنید؛ از این جهت که حروفی که باید escape شوند، به یک فاصله‌ی ساده ختم نمی‌شوند. یک لیست طولانی دارد.
- اگر قصدتان تولید slug است، از متده [GenerateSlug](#) مطرح شده در مطلب «[بهینه سازی برنامه‌های وب ASP.NET](#) برای [موتورهای جستجو \(SEO\)](#)» ایده بگیرید. در انتهای آن یک متده [Trim](#) هم فراخوانی شده است (برای حذف فواصل خالی در ابتداء و انتهای رشته).

نویسنده: موسوی
تاریخ: ۱۳۹۳/۰۹/۲۲ ۱۹:۹

سلام برای خوندن اطلاعات از کوئری استرینگ باید چطور دستور سلکت رو نوشت من وقتی از روش بالایی که گفتم استفاده میکردم میگفتم که مقدار `id` رو بخون از کوئری استرینگ الان نوشتم `where newsid=@pi` بعدم تو `qedita` سورس گفتم از کوئری استرینگ مقدار `pi` رو بخون ولی ارور میده الان بگم چی رو بخون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۲۲ ۱۹:۲۴

```
SqlDataSource1.SelectParameters.Add("@param1", value1);
```

نویسنده: موسوی
تاریخ: ۱۳۹۳/۰۹/۲۲ ۱۹:۴۵

مقداری که از کوئری استرینگ میخونه چیه همون `pi`? آخه من قبل `id` میگفتم بخون وجواب میداد اما الان گفتم `pi` رو بخون ولی جواب نداد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۲۲ ۱۹:۴۹

- یکبار متن و نظرات را کامل مطالعه کنید. اگر از روش جدید مسیریابی استفاده کنید، دیگر نباید کوئری استرینگی در URL شما مشاهده شود (یکی از اهداف اصلی سیستم مسیریابی). در اینجا یک الگو را باید در متده [RouteTable.Routes.MapPageRoute](#) ثبت کنید.

- `pi` بر اساس همان الگوی ثبت شدهی `{pt}/{pi}` است؛ در مثالی که در متن زده شده.

نویسنده: موسوی
تاریخ: ۱۳۹۳/۰۹/۲۳ ۹:۸

سلام من تو `global.asax` اینجوری نوشتم

```
<%@ Application Language="C#" %>
<script runat="server">

    void Application_Start(object sender, EventArgs e)
    {
        // Code that runs on application startup
        System.Web.Routing.RouteTable.Routes.MapPageRoute("NewsDatils", "News/{pi}/{pt}",
        "~/ShowNews.aspx");
    }

    void Application_End(object sender, EventArgs e)
    {
        // Code that runs on application shutdown
    }

```

}

بعد تو صفحه لینک اینجوری نوشتم

```
<asp:HyperLink ID="HyperLink2" runat="server" NavigateUrl='<%# string.Format("/{0}/{1}", Eval("NewsID"), Eval("NewsTitle").ToString().Replace(" ","-")) %>' Text='<%# Eval("NewsTitle") %>'></asp:HyperLink>
```

و تو `shownews` الان مقدار `pi` برابر `null`

```
int p = int.Parse(Page.RouteData.Values["pi"].ToString());
```

اینم اروری که نشون میده

Cannot use a leading .. to exit above the top directory.

نویسنده: وحید نصیری
تاریخ: ۹:۳۸ ۱۳۹۳/۰۹/۲۳

به احتمال زیاد در صفحه یک چنین آدرسی را دارید:

```
<asp:Image ImageUrl="..\foo.jpg" />
```

در اینجا .. به یک سطح بالاتر از ریشه سایت اشاره می‌کند که معتبر نیست. بجای .. از ~ استفاده کنید یا از روش `ResolveUrl` که [کمی بالاتر در نظرات توضیح داده شد](#).

نویسنده: موسوی
تاریخ: ۱۵:۲۸ ۱۳۹۳/۰۹/۲۳

ممنون. تو صفحه دوتا ادرس فایل‌های `dz` رو به این صورت .. /داده بودم و عوض کردم با ~ و مشکل حل شد.

نویسنده: موسوی
تاریخ: ۱۶:۳۲ ۱۳۹۳/۰۹/۲۳

سلام؛ چطور یه لینک بسازه به خانه برگرده. آخه وقتی مثل قبل امتحان کردم اینطور:

```
localhost:28625/News/26/~/default.aspx
```

شد؟

نویسنده: رضا منصوری
تاریخ: ۲۳:۲۴ ۱۳۹۳/۰۹/۲۳

سلام ، به این صورت

```
<a href=<%= ResolveUrl("~/") %> >Home</a>
<a href="/" >Home</a>
```

نویسنده: موسوی

با سلام؛ همه چیز عالی کار کرده فقط یه مشکل دارم اونه که تو url به کاراکترهای / ، : ، گیر میده و iis اخطار میگیره راهشم اینه که تو ادرس دهی بیام به جای این کاراکترها - یا + بزارم تو این لینک چطور بنویسم که مشکلم حل شه

```
NavigateUrl='<%# string.Format("/{0}/{1}",Eval("NewsID"),Eval("NewsTitle").ToString().Replace("-", "-"))'
```

به جای کد زیر چطور بنویسم که درست کار کنه. تشکر

```
Replace(" ", "-")
```

نویسنده: وحید نصیری
تاریخ: ۱۳:۱۶ ۱۳۹۳/۰۹/۲۶

کمی بالاتر پاسخ گرفتید « [از روش‌های ابداعی برای encoding رشته‌ها در وب استفاده نکنید ...](#) »

نویسنده: امیر نوروزیان
تاریخ: ۱۲:۱۶ ۱۳۹۳/۱۰/۱۷

با سلام؛ جدولی دارم که لیست دانلودها رو نشون میده. وقتی روی یک لینک از جدول کلیک میکنی، میخام چنین آدرسی رو نشون بدی:

آشنایی با قوانین کار /ReDownload/1127
حالا برای اینکه لیست دانلودها که یکسری مقاله هست، توی جستجوی گوگل بیشتر به چشم بخوره، او مدم آدرس بالا رو تبدیل به این آدرس کردم:

/ReDownload/1127.docx

ولی متاسفانه وقتی روی لینک کلیک میکنی `http error 404 -not found`
حالا سوال اصلی اینه، کلا روش من غلطه یا باید تنظیمات خاصی رو اعمال کنم؟

نویسنده: محسن خان
تاریخ: ۱۳:۳۲ ۱۳۹۳/۱۰/۱۷

[مرجع رسمی](#) مسیریابی در وب فرم‌ها عنوان می‌کنه که URL routing allows you to configure an application to accept request URLs that do not map to physical files . یعنی چون آدرس شما الان پسوند پیدا کرده، دیگه وارد سیستم مسیریابی نمیشه و به صورت یک فایل فیزیکی پردازش میشه. یعنی در مسیر و پوشه‌ای شبیه به آدرسی که نوشتید به دنبال اون فایل می‌گردد (که نیست و به همین جهت خطای 404 رو دریافت می‌کنید). البته با تنظیم `RouteTable.Routes = RouteExistingFiles = true` امکان تغییر این پیش فرض هست. در این حالت درخواست تمام فایل‌های فیزیکی وارد سیستم مسیریابی میشون. البته در این مورد خاص باید یک [IRouteHandler](#) بنویسید تا این درخواست فایل رسیده رو پردازش کنه.

نویسنده: امیر نوروزیان
تاریخ: ۱۴:۱۵ ۱۳۹۳/۱۰/۱۷

فایل فیزیکی وجود نداره پسوندم برای جستجو تو گوگل اضافه کردم. تنظیمان مورد نظرم رو اضافه کردم در اربطه `handler` بیشترم لطفا توضیح بدین ممنون میشم

[stackoverflow](#)

از توضیحات بالام استفاده کردم ولی جوابی نگرفتم

نویسنده: محسن خان
تاریخ: ۱۶:۰ ۱۳۹۳/۱۰/۱۷

گوگل اینقدر هوشمند هست که بتونه تشخیص بده الان درخواست یک فایل word رو داده ولی جوابش html هست یعنی یکجا کار

مشکل داره و نمره‌ی منفی بهت می‌ده. بهتره وقت بذاری و [مباحث اصلی SEO](#) رو در سایت بخونی.

نوبنده: امیر نوروزیان
تاریخ: ۱۷ ۱۳۹۳/۱۰/۱۷ ۱۶:۱۷

من دنبال آدرس جعلی نیستم. تو بعضی سایت‌ها مثل این somesite.ir/thesis/pdf/1.pdf لینک رو انطوری داده کلیک که می‌کنی دانلود می‌کنه. گوگل هم این لینک رو به عنوان یه فایل pdf می‌شناسه.

نوبنده: آرش مصیر
تاریخ: ۱۲:۲۹ ۱۳۹۳/۱۲/۲۳

نکته: (نقطه) در مسیری مانند: ASP.NET /products/.net را وادر می‌کند تا با این درخواست همانند درخواست یک صفحه رفتار کند در صورتی که صفحه‌ای در کار نیست و منظور محصولی با عنوان .net می‌باشد. جهت حل این مشکل از / در آخر آدرس باید استفاده شود.

برای تهیه تصاویر سایت‌های معرفی شده در قسمت [اشتراک‌های سایت](#)، پیشتر از کنترل WebBrowser دات نت که در پشت صحنه از امکانات IE کمک می‌گیرد، استفاده می‌کرد. بسیار ناپایدار است؛ به روز رسانی مشکلی داشته و وابسته است به سیستم عامل جاری سیستم. برای مثال مرتبا برای تهیه تصاویر بند انگشتی (Thumbnails) سایت‌های تهیه شده با بوت استرپ کرش می‌کرد و این کرش چون از نوع [unmanaged code](#) است، عمل پروسه IIS و ب سایت را از کار می‌انداخت و در این حالت سایت را استارت بعدی IIS در دسترس نبود. برای حل این مشکل و بهبود کیفیت تصاویر تهیه شده، از پروژه [Awesomium](#) که در حقیقت مرورگر کروم را جهت استفاده در انواع و اقسام زبان‌های برنامه نویسی محصور می‌کند، کمک گرفته شد؛ که شرح آن را در ادامه ملاحظه خواهید کرد.

دریافت و نصب Awesomium

پروژه Awesomium دارای یک SDK است که [از اینجا قابل دریافت](#) می‌باشد. بعد از نصب آن در مسیر Awesomium SDK\1.7.3.0\wrappers\Awesomium.NET\Assemblies\Packed در اینجا، استفاده از DLL‌های فشرده شده‌ی آن است که در مسیر Packed SDK\1.7.3.0\build\bin\packed native کپی شده‌اند. بنابراین برای توزیع این نوع برنامه‌ها نیاز است اسambilی دات نتی Awesomium.Core.dll به همراه دو فایل بومی awesomium.dll و icudt.dll ارائه شوند.

تهیه تصاویر سایت‌ها به کمک Awesomium.NET

پس از نصب Awesomium SDK اگر به مسیر Documents\Awesomium Samples\1.7.3.0\Awesomium.NET\Samples\Core\CSharp\BasicSample مراجعه کنید، مثالی را در مورد تهیه تصاویر سایت‌ها به کمک Awesomium.NET مشاهده خواهید کرد. خلاصه‌ی آن چند سطر ذیل است:

```

try
{
    using (WebSession mywebsession = WebCore.CreateWebSession(
new WebPreferences() { CustomCSS = "::-webkit-scrollbar { visibility: hidden; }" }))
    {
        using (var view = WebCore.CreateWebView(1240, 1000, mywebsession))
        {
            view.Source = new Uri("https://site.com/");
            bool finishedLoading = false;
            view.LoadingFrameComplete += (s, e) =>
            {
                if (e.IsMainFrame)
                    finishedLoading = true;
            };
            while (!finishedLoading)
            {
                Thread.Sleep(100);
                WebCore.Update();
            }
            using (var surface = (BitmapSurface)view.Surface)
            {
                surface.SaveToJPEG("result.jpg");
            }
        }
    }
    finally
    {
        WebCore.Shutdown();
    }
}

```

کار با ایجاد یک WebSession شروع می‌شود. سپس با مقدار دهی CustomCSS، اسکرول بار صفحات را جهت تهیه تصاویری بهتر مخفی می‌کنیم. سپس یک WebView آغاز شده و منبع آن به Url مدنظر تنظیم می‌شود. در ادامه باید اندکی صبر کنیم تا بارگذاری سایت خاتمه یافته و نهایتاً می‌توانیم سطح View را به صورت یک تصویر ذخیره کنیم.

مشکل! این روش در برنامه‌های ASP.NET کار نمی‌کند!

مثال همراه آن یک مثال کنسول ویندوزی است و به خوبی کار می‌کند؛ اما در برنامه‌های وب پس از چند روز سعی و خطای مشخص شد که:

(الف) WebCore.Shutdown فقط باید در پایان کار یک برنامه فراخوانی شود. یعنی اصلاً نیازی نیست تا در برنامه‌های وب فراخوانی شود.

```
System.InvalidOperationException: You are attempting to re-initialize the WebCore.  
The WebCore must only be initialized once per process and must be shut down only when the process  
exits.
```

(ب) Awesomium فقط در یک ترد کار می‌کند. به این معنا که اگر کدهای فوق را در یک صفحه‌ی وب فراخوانی کنید، بار اول کار خواهد کرد. بار دوم برنامه کرش می‌کند؛ با این پیغام خطای:

```
System.AccessViolationException: Attempted to read or write protected memory.  
This is often an indication that other memory is corrupt. at  
Awesomium.Core.NativeMethods.WebCore_CreateWebView_1(HandleRef jarg1, Int32 jarg2, Int32 jarg3,  
HandleRef jarg4)
```

چون هر صفحه‌ی وب در یک ترد مجزا اجرا می‌شود، عمل استفاده‌ی مستقیم از Awesomium در آن ممکن نیست. خطای فوق هم از آن نوع خطاهایی است که پروسه‌ی IIS را درجا خاموش می‌کند.

استفاده از Awesomium در یک ترد پس زمینه

راه حلی که نهایتاً پاسخ داد و به خوبی و پایدار کار می‌کند، شامل ایجاد یک ترد مجزای Awesomium در زمان آغاز برنامه‌ی وب و زنده نگه داشتن آن تا زمان پایان کار برنامه است.

```
using System;  
using System.Collections.Concurrent;  
using System.IO;  
using System.Threading;  
using System.Web;  
using Awesomium.Core;  
  
namespace AwesomiumWebModule  
{  
    public class AwesomiumModule : IHttpModule  
    {  
        private static readonly Thread WorkerThread = new Thread(awesomiumWorker);  
        private static readonly ConcurrentQueue<AwesomiumRequest> TaskQueue = new  
ConcurrentQueue<AwesomiumRequest>();  
        private static bool _isRunning = true;  
  
        static AwesomiumModule()  
        {  
            WorkerThread.Start();  
        }  
  
        private static void awesomiumWorker()  
        {  
            while (_isRunning)  
            {  
                if (TaskQueue.Count != 0)  
                {  
                    AwesomiumRequest outRequest;  
                    if (TaskQueue.TryDequeue(out outRequest))  
                    {  
                        var img = AwesomiumThumbnail.FetchWebPageThumbnail(outRequest);  
                    }  
                }  
            }  
        }  
    }  
}
```

```
        File.WriteAllBytes(outRequest.SavePath, img);
        Thread.Sleep(500);
    }
    Thread.Sleep(5);
}
}

public void Dispose()
{
    _isRunning = false;
    WebCore.Shutdown();
}

public void Init(HttpApplication context)
{
    context.EndRequest += endRequest;
}

static void endRequest(object sender, EventArgs e)
{
    var url = HttpContext.Current.Items[Constants.AwesomiumRequest] as AwesomiumRequest;
    if (url!=null)
    {
        TaskQueue.Enqueue(url);
    }
}
```

در اینجا اگر در برنامه‌های وب فرم، از طریق `HttpContext.Current.Items.Add` و یا در برنامه‌های MVC به کمک `this.HttpContext.Items.Add` یک آیتم جدید، با کلید `Constants.AwesomiumRequest` و از نوع کلاس `AwesomiumRequest` دریافت گردد، مقدار آن به یک `ConcurrentQueue` اضافه خواهد شد. این صفت در یک ترد مجزا مدام در حال بررسی است. اگر مقداری به آن اضافه شده‌است، از صفت `Харж_Шед` و پردازش خواهد شد.

```
<?xml version="1.0"?>
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.0" />
    <httpModules>
      <add name="AwesomiumWebModule" type="AwesomiumWebModule.AwesomiumModule"/>
    </httpModules>
  </system.web>
  <system.webServer>
    <validation validateIntegratedModeConfiguration="false"/>
    <modules>
      <add name="AwesomiumWebModule" type="AwesomiumWebModule.AwesomiumModule"/>
    </modules>
  </system.webServer>
</configuration>
```

سیس باید تنها مدیریت `HttpContext.Current.Items` در سمت برنامه صورت گیرد:

```
protected void btnStart_Click(object sender, EventArgs e)
{
    var host = new Uri(txtUrl.Text).Host;
    HttpContext.Current.Items.Add(Constants.AwesomiumRequest, new AwesomiumRequest
    {
        Url = txtUrl.Text,
        SavePath = Path.Combine(HttpContext.Current.AppDomain.AppPath, "App_Data\\Thumbnails\\" + host +
".jpg"),
        TempDir = Path.Combine(HttpContext.Current.AppDomain.AppPath, "App_Data\\Temp")
    });
    lblInfo.Text = "Please wait. Your request will be served shortly.";
}
```

در اینجا کاربر درخواست خود را در صفحه می‌دهد. پس از مدتی کار آن در WorkerThread مازلول تهیه شده انجام گردیده و

تصویر نهایی تهیه می‌شود.

Url، آدرس وب سایتی است که می‌خواهید تصویر آن تهیه شود. SavePath مسیر کامل فایل jpg نهایی است که قرار است دریافت و ذخیره گردد. TempDir محل ذخیره سازی فایل‌های موقتی Awesomium است. یک سری کوکی، تصاویر و فایل‌های هر سایت را به این ترتیب کش کرده و در دفعات بعدی سریعتر عمل می‌کند.

پروژه‌ی کامل آنرا از اینجا می‌توانید دریافت کنید:

[AwesomiumWebApplication_V1.0.zip](#)

نظرات خوانندگان

نویسنده: سعید شیرزادیان
تاریخ: ۱۹:۱۳ ۱۳۹۲/۱۱/۱۶

پروژه فوق در ویزوال استودیو ۲۰۱۲ باز نمی‌شود پیغام عدم سازگاری می‌دهد. آیا راهنمایی می‌توانید بکنید با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۹:۲۳ ۱۳۹۲/۱۱/۱۶

فایل‌های `.sln` و `.csproj` و آنرا حذف کنید. بعد دو پروژه وب فرم و `MVC` خالی درست کنید و فایل‌های موجود را به آن‌ها اضافه کنید. ماژول آن هم یک `class library` ساده است.

نویسنده: Raees
تاریخ: ۲۰:۳ ۱۳۹۳/۰ ۱۰/۰۵

Hi Vahid

You have done an excellent job. But I have one query after generating one image the second one does not generate i mean do i need to refresh the page or do something special. The files in the folder gets hold by the worker process and hence the subsequent request are not processed can you help us in this

نویسنده: وحید نصیری
تاریخ: ۲۰:۵۷ ۱۳۹۳/۰ ۱۰/۰۵

Remove the `AwesomiumModule->Dispose` method's body. It's not necessary

همان طور که می‌دانید نقشه‌ی سایت علاوه بر استفاده از Tag MetaTag ها و URL Routing ... یکی از نکات اصلی برای سایت شماست که در نتایج گوگل برای جستجو کنندگان نمایش داده شود.

در این مقاله من قصد دارم که به شما چگونگی ایجاد و کار با یک SiteMap داینامیک را آموزش دهم. منظور از SiteMap داینامیک این است که به ازای هر مطلبی که شما در سایت تان ایجاد، ویرایش یا حذف می‌کنید چنانچه دارای صفحه‌ای در سایت باشد SiteMap تغییر می‌کند.

ساختار یک SiteMap به صورت زیر است:

```
<?xml version="1.0" encoding="utf-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>Url Page 1</loc>
    <lastmod>2014-02-20</lastmod>
    <changefreq>always</changefreq>
    <priority>1</priority>
  </url>
  <url>
    <loc>Url Page 2</loc>
    <lastmod>2014-02-20</lastmod>
    <changefreq>always</changefreq>
    <priority>1</priority>
  </url>
</urlset>
```

به ازای هر مطلبی که به سایتتان اضافه می‌کنید چنانچه آن مطلب جهت نمایش دارای URL باشد، باید یک تگ loc به SiteMap اضافه شود.

تگ loc موجود در تگ URL آدرس صفحه را مشخص می‌کند.
تگ lastmod تاریخ اضافه کردن یا آخرین ویرایش را نمایش می‌دهد.
تگ changefreq که دوره‌ی بروز رسانی صفحه را مشخص می‌کند.
تگ priority الویت صفحه را مشخص می‌کند.

که من در کد نویسی تگ priority را always و تگ changefreq را 1 قرار دادم.

در فایل ضمیمه یک کلاس به اسم updateSiteMap.cs وجود دارد که تابع آن شامل دو پارامتر ورودی مانند زیر است:

```
public void UpdateSiteMap(string Addr, string NewOpr)
```

پارامتر Addr که آدرس صفحه‌ای است که شما می‌خواهید به SiteMap اضافه شود.
پارامتر NewOpr که می‌تواند شامل یکی از سه مقدار زیر باشد: add, edit, delete.

اگر پارامتر NewOpr دارای مقدار add باشد یعنی مقدار موجود در پارامتر Addr را به SiteMap اضافه کن.
اگر پارامتر NewOpr دارای مقدار edit باشد یعنی مقدار موجود در تگ lastmod را ویرایش کن. یعنی تاریخ آخرین بروزرسانی تغییر می‌کند.
اگر پارامتر NewOpr دارای مقدار delete باشد یعنی تگ loc ای که محتوای آن برابر است با مقدار موجود در پارامتر

را از SiteMap Addr حذف کن.

این بخش از کد موجود در فایل ضمیمه updateSiteMap.cs قسمت edit و delete نقشه‌ی سایت را انجام می‌دهد.

```
if (NewOpr != "add")
{
XmlElement xmlElement = xmlDoc.DocumentElement;
if (xmlElement.ChildNodes != null)
{
foreach (XmlElement myElement in xmlDoc.DocumentElement)
{
if (myElement.ChildNodes[0].InnerText == Addr)
{
if (NewOpr != "delete")
myElement.ChildNodes[1].InnerText = DateTime.Now.ToString("yyyy-MM-dd");
else
myElement.ParentNode.RemoveChild(myElement);
break;
}
}
}
```

و بخش else دستور بالا قسمت Add را انجام می‌دهد. یعنی کدهای زیر:

```
else
{
string ns="http://www.sitemaps.org/schemas/sitemap/0.9";
XmlNode url = xmlDoc.CreateNode(XmlNodeType.Element, "url", ns );
XmlNode loc = xmlDoc.CreateNode(XmlNodeType.Element, "loc", ns);
XmlNode lastmod = xmlDoc.CreateNode(XmlNodeType.Element, "lastmod", ns);
XmlNode changefreq = xmlDoc.CreateNode(XmlNodeType.Element, "changefreq", ns);
XmlNode priority = xmlDoc.CreateNode(XmlNodeType.Element, "priority", ns);
loc.InnerText = Addr;
lastmod.InnerText = DateTime.Now.ToString("yyyy-MM-dd");
changefreq.InnerText = "always";
priority.InnerText = "1";
url.AppendChild(loc);
url.AppendChild(lastmod);
url.AppendChild(changefreq);
url.AppendChild(priority);
xmlDoc.DocumentElement.AppendChild(url);
}
```

اگر اطلاعاتی را به جدول اضافه می‌کنید و می‌خواهید Url صفحه‌ی مربوط به آن اطلاعات برای شما در SiteMap اضافه شود بعد از ذخیره شدن اطلاعات در جدول بلافصله کد زیر را اضافه می‌کنید:

```
//
Add Info In Table
//
updateSiteMap updateSiteMap = new updateSiteMap();
updateSiteMap.UpdateSiteMap("Url Page", "add");
```

برای قسمت ویرایش هم پس از آنکه اطلاعات را ویرایش کردید چنانچه برای آن اطلاعات صفحه‌ای را در SiteMap درج کرده اید کد زیر را می‌نویسید:

```
updateSiteMap updateSiteMap = new updateSiteMap();
updateSiteMap.UpdateSiteMap("Url Page", "edit");
```

برای قسمت حذف هم اگر شما اطلاعاتی را از جدول حذف می‌کنید چنانچه برای آن اطلاعات صفحه‌ای در SiteMap درج کرده اید کد زیر را می‌نویسید:

```
updateSiteMap updateSiteMap = new updateSiteMap();
updateSiteMap.UpdateSiteMap("Url Page", "delete");
```

موفق باشید .

[Files.zip](#)

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۰۲:۴۶

چند نکته تکمیلی

- نسخه‌ی MVC این بحث را [در اینجا](#) می‌توانید مطالعه کنید. این نسخه را می‌توان تبدیل به یک فایل `ashx` و ب فرم‌ها نیز کرد.
 - آن قابل انتقال است و با کد انتهای بحث یکی است.
 - در وب فرم‌ها الزاماً نیازی نیست تا حتماً یک فایل XML ثابت را روی سخت دیسک ذخیره کنید. می‌شود با استفاده از [مباحث context](#) در وب فرم‌ها، فایل‌ی را که وجود خارجی ندارد، ایجاد کرد:

```
void Application_Start(object sender, EventArgs e)
{
    RouteTable.Routes.Add(new Route(
        "sitemap.xml", new PageRouteHandler("~/sitemap.ashx")));
}
```

- سیس، برای تولید فایل XML به صورت بوسیله sitemap.ashx استفاده کنید به همراه مباحث caching اطلاعات در مرورگر وارد کند، مدیریت آن توسط sitemap.ashx انجام خواهد شد.

```
public class Sitemap : IHttpHandler
{
    public void ProcessRequest(HttpContext context) {
        context.Response.ContentType = "text/xml";
        مابقی نکات بحث در اینجا//
        context.Response.End();
    }

    public bool IsReusable {
        get { return false; }
    }
}
```

نويسنده: صابر فتح الهي
تاریخ: ۱۳۹۲/۱۲/۰۲

متشرک از زحمتی که برای این مطلب کشیدید.
یک ایراد ویرایشی کوچک:

که من در کد نویسی تگ changefreq را 1 و تگ priority را always قرار دادم.
ظاهر اجرا نشوند.

نويسنده: سلمان کاظمی
تاریخ: ۱۳۹۲/۱۲/۲۸

من یه مشکلی با سایت مپ دارم و اون اینه که من یک فرم دارم که ممکنه چند فرم مختلف این فرم را نمایش میدهند. حالا این روش شما مشکل را حل می‌کنه؟ مثلاً اگه اسم فرم من A باشه می‌خواهم وقتی از فرم B فرم A را نمایش میدهم سایت مپ بصورت $B \rightarrow A$ نشون داده بشه و اگه از فرم A را نشون میدم شکل، سایت مپ بصورت $A \rightarrow C$ باشه

نویسنده: وحید نصیری تاریخ: ۱۳۹۲/۱۲/۲۸

[sitemap](#) معرفی شده در مطلب حاره، صرفهای، متورهای، هستجو نوشته شده است و کاربرد مدنظر شما، اندارد.

نوسینده: محسن حفظی

من از این روش (تولید فایل XML به صورت پویا از طریق sitemap.ashx) استفاده کردم ولی متصفحانه خطای زیر را میده (از دات نت 4.5 استفاده می‌کنم)

Type 'sitemap' does not inherit from 'System.Web.UI.Page'.

نویسنده: وحید نصیری

تاریخ: ۱۰:۱۳ ۱۳۹۳/۰۵/۲۸

نیاز به کمی اصلاح دارد. جنریک هندرها برای معرفی به سیستم مسیریابی باید از طریق یک IRouteHandler معرفی شوند:

```
// نحوهی معرفی جنریک هندر به سیستم مسیریابی
public class RouteHandler : System.Web.Routing.IRouteHandler
{
    public IHttpHandler GetHttpHandler(System.Web.Routing.RequestContext requestContext)
    {
        return new Sitemap();
    }
}

// Global.asax file
void Application_Start(object sender, EventArgs e)
{
    RouteTable.Routes.Add(new Route("sitemap.xml", new RouteHandler()));
}
```

[خلاصه آن به صورت یک متد الحاقی](#)

طمئناً اکثر شما برنامه نویسان با معماری IIS و ASP.NET کمابیش آشنایی دارید Request از سمت کلاینت به IIS ارسال می‌شود، و عموماً بسته به نوع درخواست کلاینت یا به یک Static File مپ می‌شود (مثلاً به یک عکس)، و یا به یک ISAPI

کدی است که عموماً با C++ نوشته می‌شود، و برای درخواست آمده از سمت کلاینت کاری را انجام می‌دهد یکی از این ISAPI‌ها برای ASP.NET است، که درخواست کلاینت را به یک کد مبتنی بر .NET مپ می‌کند (به همین علت به آن ASP.NET می‌گویند)

نکته‌ای که در خطوط فوق به وضوح دیده می‌شود، وابستگی شدید ASP.NET به IIS است بدیهتاً کدی که بر روی بستر ASP.NET نوشته می‌شود نیز وابستگی فوق العاده ای به IIS دارد، که یکی از بدترین نوع این وابستگی‌ها در ASP.NET Web Forms می‌شود.

خب، این مسئله چه مشکلاتی را ایجاد می‌کند ؟

مشکل اول که شاید کمتر به چشم بیاید، بحث کندي اجرای بار اول برنامه‌های ASP.NET است.

اما مشکل دوم عدم توانایی در نوشتن کد برنامه، بدون وابستگی به وب سرور (در اینجا IIS) است، که این مشکل دوم روز به روز در حال جدی‌تر شدن است.

این مشکل دوم را برنامه نویسان جاوا سالهای است که با آن درگیرند، نکته این است که بین دو وب سرور در نحوه پردازش یک درخواست کلاینت تفاوت هایی وجود دارد، که بالطبع این تفاوت‌ها در نحوه اجرای کد بالاخره خودش را جاهایی نشان می‌دهد، این که بگوییم رفتار وب سرورها نباید متفاوت باشد کمی مسخره است، زیرا تفاوت آنها با یکدیگر باعث شده که سرعت یکسان و امکانات یکسانی نداشته باشند و هر کدام برای یک سناریوی خاص مناسب‌تر باشند این مسئله برای ما نیز روز به روز دارای اهمیت بیشتری می‌شود، دیگر این که Web Server ما فقط IIS صرف باشد، سناریوی متداول در پروژه‌های Enterprise نیست در چه جاهایی می‌توان یک برنامه را هاست کرد ؟

IIS به همراه ASP.NET

IIS بدون ASP.NET (می‌خواهیم برنامه بر روی IIS هاست شود، ولی کاری با ASP.NET نداریم)

و وب سرورهای لینوکسی در صورت اجرای برنامه بر روی Mono

و ...

هم اکنون به میزان زیادی مشکل شفاف شده است، مطابق با معماری فعلی داریم

Request >> IIS >> aspnet_isapi.dll >> System.Web.dll >> Your codes

مشکل دیگری که وجود دارد این است که اگر تیمی بخواهد فریم ورکی برای برنامه نویسان نهایی فراهم کند، باید آنرا بر روی اکثر گزینه‌های هاست موجود سازگار کنید، برای مثال مشاهده می‌کنید که ASP.NET SignalR را هم می‌توان بر روی IIS و ASP.NET هاست کرد و هم بر روی یک App Domain کاملاً معمولی و علاوه بر این که تیم SignalR باید این هزینه مضاعف را پرداخت کند، خروجی برای ما نیز چندان خوشایند نیست، برای مثال اجرای همزمان ASP.NET SignalR و ASP.NET Web API اگر چه که بر روی هاستی به غیر از ASP.NET نیز امکان پذیر است، اما متأسفانه به عنوان دو بازیگر جدا از هم کار می‌کنند و عملاً تعاملی با یکدیگر ندارند، مگر این که بر روی ASP.NET هاست شوند، و یا بسیاری از امکانات Routing موجود در WCF بر روی بستری غیر از ASP.NET کار نمی‌کند. بدیهی است که این بازار پر آشوب به نفع هیچ کس نیست. و اما راه حل چیست ؟ تعدادی از برنامه نویسان حرفه‌ای.NET دور یکدیگر جمع شدند و طی بررسی هایشان به این نتیجه رسیدند که هاست‌های مختلف نقاط اشتراک بسیار زیادی دارند و تفاوت‌ها نباید باعث این میزان مشکل شود.

پس استانداری را طراحی کردند با نام **OWIN** یا **Open Web Interface for .NET**.

این استاندارد به صورت کاملاً ریز به طراحی هر چیزی را که شما به آن فکر کنید پرداخته است، Request, Cookie, Response, ... و Web Sokcet

اما همانطور که از نامش مشخص است این یک استاندارد است و پیاده سازی ندارد، و هر هاستی باید یک بار این استاندارد را بر

روی خود پیاده سازی کند

خبر خوش این است که تا این لحظه اکثر هاستهای مهم این استاندارد را پیاده سازی کرده اند و یا در دست پیاده سازی دارند پروژه Helios برای IIS

پروژه [Katana](#) برای IIS به در کنار و سازگار با ASP.NET برای پروژه هایی که تا این لحظه از امکانات سطح پایین ASP.NET استفاده زیادی کرده اند و فرست تغییر ساختاری ندارند

پروژه هایی برای App Domains و ...

مرحله‌ی بعدی این است که فریم ورک‌ها خوشان را با Owin سازگار کنند

معروف‌ترین فریم ورک هایی که تا این لحظه اقدام به انجام این کار کرده اند، عبارتند از:

ASP.NET Web API

ASP.NET MVC

ASP.NET Identity

(در حال حاضر Signal R فقط بر روی Owin قابلیت استفاده دارد)

بدیهی است که زمانی که پروژه ASP.NET Web API بر روی استاندارد OWIN نوشته می‌شود، دیگر نیازی به تحمل هزینه مضاعف برای سازگاری خود با انواع هاست‌ها ندارد و این مسئله توسط Katana، Helios و ... انجام شده است، که بالطبع بزرگترین نفع آن

برای ما جلوگیری از چند باره کاری توسط تیم Web API و ... است که بالطبع در زمان کمتر امکانات بیشتری را به ما ارائه می‌دهند. البته واضح است فریم ورک هایی که با کلاینت و درخواست‌ها کاری ندارند، با این مقولات کاری ندارند، بسیار Entity Framework و

... از این داستان مستثنی هستند. علاوه بر این فریم ورک هایی با طراحی اشتباه و قدیمی مانند ASP.NET Web Forms به صورت کلی قابلیت سازنگار شدن با این استاندارد را ندارند، زیرا کاملاً به ASP.NET وابسته هستند

و در نهایت در مرحله‌ی بعدی لازم است شما نیز از فریم ورک هایی استفاده کنید که مبتنی بر OWIN هستند، یعنی برای مثال پروژه بعدی، تان را مبتنی بر ASP.NET Identity و ASP.NET Web API و ASP.NET MVC ساخته سازی کنید، در این صورت شما می‌توانید ب‌نامه

ا، بنویسید که به Web Server هیچ گونه واستگ ندارد.

به این صورت کد زدن چند مزیت بزرگ دیگر هم دارد که از کم اهمیت‌ترین آنها شروع می‌کنیم:

۱- سرعت بسیار بالاتر برنامه در هاستهای غیر ASP.NET ای، مانند زمانی که شما از IIS به صورت مستقیم و بدون وابستگی به System.Web.dll استفاده می‌کنید.

توجه کنید که حتی در این حالت هم می‌توانید از ASP.NET Web API و SignalR و Identity استفاده کنید و تا 25% سرعت بیشتری داشته باشید (بسته به سناریو) 2- قابلیت توسعه آسانتر و با قابلیت نگهداری بالاتر پروژه‌های Enterprise، برای مثال در

یکی از پروژه‌ها من مجبور بودم از ASP.NET Web API به صورتی استفاده کنم که هم توسط کلاینت JavaScript ای استفاده شود، و هم توسط کدهای Controller MVC (بدون استفاده مستقیم از کد سرویس با رفرانس زدن به سرویس‌ها البته) که خوب‌بختانه

AppDomain به خوبی از پس این کار برآمد، و عملایک سرویس Web API را هم بر روی IIS هاست کردم و هم داخل یک 3- در چند سال آینده که اکثریت مطلق سایتها از این روش استفاده کنند (شما چه بدانید و چه ندانید اگر در برنامه خودتان از

Signal R نسخه 2 دارید استفاده می کنید، حتما از OWIN استفاده کرده اید)، مایکروسافت می تواند دست به تغییرات اساسی تری بزند، برای مثال معماری جدیدی از IIS ارائه دهد که مشکلات ساختاری فراوان فعلی IIS را که از حوصله توضیح این مقاله خارج

است را نداشته باشد، و فقط یک پیاده سازی OWIN جدید بر روی آن ارائه دهد و برنامه های ما بدون تغییر بر روی آن نیز کار کنند، و یا این که بتواند تعدادی از فریم ورک های با طراحی قدیمی را راحت تر از دور خارج کند، مانند Web Forms

نکته پایانی، اگر هم اکنون پروژه‌ای دارید که در داخل آن از ASP.NET استفاده شده، و برای مثال تعدادی فرم ASP.NET Web Forms نیز دارد، نگران نباشید، کماکان می‌توانید از Owin برای سایر قسمت‌ها مانند Web API استفاده کنید، البته در این حالت تاثیری در

نظرات خوانندگان

نوبت‌دهنده: ناظم
تاریخ: ۱۳۹۲/۱۲/۰۳ ۱۰:۵۱

سلام
ممnon بابت مطلب مفیدتون.

بدون وابستگی به IIS یعنی هر web server-ی که OWIN را پیاده سازی کند امکان اجرای برنامه‌هایی که مثلاً با ASP.NET MVC نوشته شدن رو خواهد داشت؟
همین که مثلاً با ASP.NET MVC برنامه نوشته شده به معنی این هست که برنامه بر اساس استاندارد OWIN هست؟ یا کارهایی برای این منظور باید انجام داد؟

نوبت‌دهنده: مسعود پاکدل
تاریخ: ۱۳۹۲/۱۲/۰۳ ۱۱:۴۸

بدون وابستگی به IIS یعنی شما امکان هاست کردن سرویس‌های Web API را به صورت Windows Service یا پروژه هم خواهید داشت.
به صورت پیش فرض یک پروژه MVC بدون وابستگی به Owin پیاده سازی می‌شود و برای این منظور می‌توانید یکی از موارد زیر را انجام دهید:
«امکان هاست سرویس‌ها روی IIS. در این صورت Owin فقط به صورت یک Middleware عمل خواهد نمود و در این حالت دیگر نیاز به نوشتن `HttpModule`-ها نخواهد داشت. البته این روش به `System.Web` وابستگی دارد (`Microsoft.Owin.Host.SystemWeb`)»
«استفاده از `OwinHost.Exe` که در واقع یک پیاده سازی دیگر برای Owin است و عملیات bootstrapping را بر عهده خواهد داشت. در نتیجه شما فقط موارد مربوط به `middleWare` در application انجام خواهید داد.»
«استفاده از Owin Self Hosting برای هاست سرویس‌ها در قالب برنامه Console یا Windows Service (`Microsoft.Owin.Host.HttpListener`)

نوبت‌دهنده: یاسر مرادی
تاریخ: ۱۳۹۲/۱۲/۰۳ ۱۲:۱۳

بله، به همین معنی است
البته دقت کنید، پیاده سازی OWIN کار ساده‌ای نیست، و به سرعت نمی‌توان شاهد پیاده سازی آن بر روی هاست‌های مختلف بود، و این پروسه با سرعت فعلی از نظر من مدتی طول خواهد کشید.
برای مثال Katana که یک پیاده سازی قابل استفاده و خوب از آن به شمار می‌رود کار شرکت مایکروسافت است و سایر پیاده سازی Open Source سایر تیم‌ها که بالطبع امکان مانور شرکت مایکروسافت را ندارند، کمی طول می‌کشد تا واقعاً آماده استفاده شود.
و همچنین پیاده سازی‌های فعلی در قسمت هایی مانند `Web Socket`ها و سایر مسائل پیچیده دارای ضعف‌هایی هستند.

درست مانند استاندارد HTML 5 که بر روی مرورگرهای مختلف به میزان‌های مختلفی پیاده سازی شده است.
به بیان دیگر پیاده سازی OWIN صفر و صدی نیست، بلکه هر پیاده سازی ممکن است در داخل خود دارای ضعف‌ها و یا نواقصی باشد.
علاوه بر این اگر شما در کد نویسی ASP.NET MVC خود، بی‌جهت به امکانات پایه ASP.NET ایجاد وابستگی کنید، نیز در این عمل دچار مشکل خواهید شد، برای همین بدیهتا کاری را که می‌توانید با Action Filter انجام دهید را نباید با یک `Http Module` انجام دهید و ...
مهمنترین کار طراحی برنامه‌هایی که می‌نویسید به صورت سازگار با OWIN است که در پست‌های بعدی قرار است به همین قسم از مطالب بپردازیم
البته من آینده خوبی برای OWIN قائلم، و نفع آن در کوتاه مدت و بلند مدت کاملاً آشکار و واضح است، کما این که در مطلب به آن اشاره شد.

برای مشاهده پیاده سازی های مختلف OWIN می توانید به سایت owin.org مراجعه کنید.
موفق و پایدار باشید

نویسنده: یاسر مرادی
تاریخ: ۱۳۹۲/۱۲/۰۳ ۱۹:۲۶

ممnon از پاسختون، البته این رو در نظر داشته باشد که استفاده از IIS به همراه Owin لزوماً به پیاده سازی Katana یا همان Microsoft.Owin.Host.SystemWeb مرتبط نیست، در این حالت شما هیچ گونه بهبود سرعتی را مشاهده نخواهید کرد و حتی به علت اضافه شدن Owin Middleware بر روی ASP.NET حتی کنترلر هم خواهید شد، این حالت فقط برای پروژه هایی توصیه می شود که با استفاده از مواردی مانند ASP.NET Module یا Web form به ASP.NET وابسته اند. برای پروژه های جدید استفاده از Helios که نه به System.Web احتیاجی دارد و نه به Owin.Host.SystemWeb توصیه می شود، به همراه SignalR ، MVC و Web API، که به نظر من این آنقدر کامل و کافی هستند که لزومی به استفاده از ASP.NET System.Web.dll و پیاده سازی Owin مربوطه ای که نام بر دید نباشد، تا بتوان بیشتر از مزایای Owin به خصوصیات کارآمدی بیشتر برنامه ها بهره برد

نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۲/۱۲/۰۳ ۲۱:۵۵

ممnonم.
در حال حاضر من استفاده از helios رو پیشنهاد نمی کنم چون اولین محدودیتی که در helios جلب توجه می کند Minimum system requirements مورد نظر است.
برای توسعه پروژه های Windows Server 2012 یا Windows 8 «
NET Framework 4.5.1«
Visual Studio 2013 یا Visual Studio 2012«

و برای Web Server نیز :
Windows Server 2012«
NET Framework 4.5.1«
Mord نیاز است.

البته به گفته تیم توسعه پروژه helios، احتمال رفع این محدودیت ها در آینده وجود دارد. در نتیجه به نظر من Owin Self Hosting گزینه بهتری برای Microsoft.AspNet.WebApi.OwinSelfHost است و از آن جا که در حالت Hosting هیچ گونه وابستگی به IIS و البته System.Web نیز وجود ندارد در نتیجه مشکل performance مشکل برطرف خواهد شد.

نویسنده: یاسر مرادی
تاریخ: ۱۳۹۲/۱۲/۰۴ ۰:۲

روش برنامه نویسی مایکروسافت بیش از دو سالی می شود که به این شکل شده است که هر امکان و قابلیت جدیدی بر روی آخرین نسخه .NET Framework ارائه می شود و البته سپس به نسخ قبلی نیز تعمیم می یابد، در همینجا است که می بینید اکثر امکانات 6 & 5 Entity Framework ابتدا بر روی .NET Framework 4.5 ارائه شدند، و سپس بر روی .NET Framework 4 اگر ما بخواهیم به .NET Framework نهاده باشیم، بلکه برای صدھا امکان دیگر مانند Data Flow های جدید و ... نیز باید صبر کنیم، که عملاً هزینه به فایده آن نمی صرفد. پس همیشه با فراغ بال از آخرین نسخه .NET Framework استقبال کنیم.

نکته دیگر را که باید مد نظر داشته باشیم، این است که مطابق با سیاست هایی که مایکروسافت جدیداً اتخاذ کرده است، دیگر نباید خیلی نگران نسخه های جدید .NET Framework باشیم، چون دیگر از آن نسخه دھی های پشت سر هم و با حجم تغییرات بالا خبری نیست، بلکه اکثر فریم و رکھای مهمن جدا از .NET ارائه و به روز رسانی می شوند.

علاوه بر این، ارتقا به آخرین نسخه سیستم عامل ویندوز نیز به هیچ وجه مانند قبل (6 IIS به 7 IIS) دردرسرا زانیست، و خوشبختانه این ارتقا (و یا تغییر هاست) بدون دردرسرا است.

به نظر من این ارتقاء را انجام دهید، چون نه فقط Helios که خیلی چیزهای دیگری را دارید از دست می‌دهید، مانند سرعت بالاتر توسعه برنامه بر روی Windows 8.1 و Visual Studio 2013 برای توسعه برنامه‌های وبی، سرعت و کارآمدی بسیار بالاتر.NET 4.5.1 با IIS 8.5 برای مشتری‌های برنامه و ...

به نظر من آنقدر این ارتقاء ارزشمند است، که ارزش Helios این میان کمتر ارزشش به چشم می‌آید.

یکی از دلایلی که برنامه‌های سمت وب به سرعت بر برنامه‌های دسکتاپ قدیمی چیره شدند، همین است: امکان ارتقای سرورها در مدت زمان کم و به شکل مدیریت شده و با کمترین تاثیر روی مشتری‌های نهایی، بارها این تصمیم را که در ابتدایش کمی سخت به نظر می‌آید را گرفته ام و در نهایت از مشتری تا برنامه نویس همه را راضی دیده ام، چون هیچ کسی از امکانات جدید که بدون دردسر حاصل شود بدم نمی‌آید، و خوشبختانه کیفیت محصولات مایکروسافت واقعاً بهبود یافته و دیگر آن زمانی که از.NET 2.0 به 3.5 می‌رفتیم و گرفتار چندین مشکل می‌شدیم گذشته است.

از این نگذرید که بالاخره روزی باید این مهاجرت‌ها را انجام دهید، پس چه بهتر که از سود آن زودتر بهره مند شوید، البته بی مهابا عمل کردن توصیه نمی‌شود، بد نیست زمانی شروع به ارتقاء کنید که صفحه Release Notes و سوالات موجود در سایت Stack over flow در رابطه با اشکالات رخ داده در زمان ارتقاء و Breaking Changes را از بر باشید، به این صورت عمل کنید تماماً برد کرده اید.

در این آموزش قصد دارم چگونگی ایجاد یک سیستم اعلام وضعیت آب و هوا را مشابه آنچه که در سایت [گوگل](#) می‌بینید برای شما توضیح دهم. باید توجه داشت من این آموزش را با ASP.NET MVC نوشتمن ولی شما می‌توانید با اندک تغییراتی در کدها، آنرا در ASP.NET Web فرمز نیز استفاده کنید. برای گرفتن آب و هوای هر شهر از RSS‌های اعلام وضعیت آب و هوای یاهو استفاده می‌کنم و توضیح خواهیم داد که چگونه با RSS آن کار کنید.

آب و هوای هر شهر در یاهو به صورت یک لینک یکتا می‌باشد؛ به شکل زیر:

<http://weather.yahooapis.com/forecastrss?w=WOEID&u=c>

حال می‌خواهیم کوئری استرینگ‌های این لینک را برای شما توضیح دهم. هر شهری بر روی کره زمین یک WOEID یکتا و منحصر بفرد دارد که شما به پارامتر w عدد WOEID شهر موردنظر خود را می‌دهید. بعد از مقدار دهی پارامتر w، وقتی این لینک را در آدرس بار مرورگر خود می‌زنید، RSS مربوط به آب و هوای آن شهر را به شما می‌دهد. مثلاً WOEID تهران عدد 28350859 می‌باشد. و این لینک RSS اطلاعات آب و هوای تهران را در قالب یک RSS به شما نمایش خواهد داد.

- خوب، حالا پارامتر دوم یعنی پارامتر u چکاری را انجام می‌دهد؟
- * چنانچه مقدار پارامتر u برابر c باشد، یعنی شما دمای آب و هوای شهر مدنظر را بر اساس سانتیگراد می‌خواهید.
- * اگر مقدار پارامتر u برابر f باشد، یعنی شما دمای آب و هوای آن شهر موردنظر را بر اساس فارنهایت می‌خواهید.

برای گرفتن WOEID شهرها هم به این سایت بروید <http://woeid.rosselliot.co.nz> و اسم هر شهری که می‌خواهید بزنید تا WOEID را به شما نمایش دهد.

در این مثال من از یک DropDownList استفاده کردم که کاربر با انتخاب هر شهر از DropDownList، آب و هوای آن شهر را مشاهده می‌کند. مرتبه Action مربوط به صفحه Index به صورت زیر می‌باشد:

```
[HttpGet]
public ActionResult Index()
{
    ViewBag.ProvinceList = _RPosition.Positions;
    ShowWeatherProvince(8);
    return View();
}
```

در اینجا من لیست شهرها را از جدول می‌خوانم که البته این جدول را چون بخش مهمی نبود و فقط شامل ID و نام شهرها بود در فایل ضمیمه قرار ندادم و نام شهرها و ID آنها را بر عهده خودتان گذاشتمن. حال تابعی را که آب و هوای مربوط به هر شهر را نمایش می‌دهد، به شرح زیر است:

```
public ActionResult ShowWeatherProvince(int dpProvince)
{
    XDocument rssXml=null;
    CountryName CountryName = new CountryName();
    if (dpProvince != 0)
    {
        switch (dpProvince)
        {
            case 1:
                {
                    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345768&u=c");
                }
        }
    }
}
```

```

        CountryName = new CountryName() { Country = "Iran", City = "Azarbeyejan-e
Sharqli" };
        break;
    }
    case 2:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345767&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Azarbeyejan-e
Qarbi" };
        break;
    }
    case 3:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2254335&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Ardebil" };
        break;
    }
    case 4:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=28350859&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Alborz" };
        break;
    }
    case 5:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345787&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Esfahan" };
        break;
    }
    case 6:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345775&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Ilam" };
        break;
    }
    case 7:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2254463&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Bushehr" };
        break;
    }
    case 8:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=28350859&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Tehran" };
        break;
    }
    case 9:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345769&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Chahar Mahall
va Bakhtiari" };
        break;
    }
    case 10:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=56189824&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Razavi
Khorasan" };
        break;
    }
    case 11:
    {
        rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345789&u=c");
        CountryName = new CountryName() { Country = "Iran", City = "Shomali
Khorasan" };
        break;
    }
    case 12:
    {
        rssXml =

```

ایجاد سیستم وضعیت آب و هوا مانند گوگل (بخش اول)

```
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345789&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Jonubi
Khorasan" };
        break;
    }
case 13:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345778&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Khuzestan" };
        break;
}
case 14:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2255311&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Zanjan" };
        break;
}
case 15:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345784&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Semnan" };
        break;
}
case 16:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345770&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Sistan va
Baluchestan" };
        break;
}
case 17:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345772&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Fars" };
        break;
}
case 18:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=20070200&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Qazvin" };
        break;
}
case 19:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2255062&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Qom" };
        break;
}
case 20:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345779&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Kordestan" };
        break;
}
case 21:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2254796&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Kerman" };
        break;
}
case 22:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2254797&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Kermanshah" };
        break;
}
case 23:
{
    rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345771&u=c");
    CountryName = new CountryName() { Country = "Iran", City = "Kohgiluyeh va
```

```

Buyer Ahmad" };
                    break;
                }
            case 24:
            {
                rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=20070201&u=c");
                CountryName = new CountryName() { Country = "Iran", City = "Golestan" };
                break;
            }
            case 25:
            {
                rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345773&u=c");
                CountryName = new CountryName() { Country = "Iran", City = "Gilan" };
                break;
            }
            case 26:
            {
                rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345782&u=c");
                CountryName = new CountryName() { Country = "Iran", City = "Lorestan" };
                break;
            }
            case 27:
            {
                rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345783&u=c");
                CountryName = new CountryName() { Country = "Iran", City = "Markazi" };
                break;
            }
            case 28:
            {
                rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345780&u=c");
                CountryName = new CountryName() { Country = "Iran", City = "Mazandaran" };
                break;
            }
            case 29:
            {
                rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2254664&u=c");
                CountryName = new CountryName() { Country = "Iran", City = "Hamedan" };
                break;
            }
            case 30:
            {
                rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345776&u=c");
                CountryName = new CountryName() { Country = "Iran", City = "Hormozgan" };
                break;
            }
            case 31:
            {
                rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2253355&u=c");
                CountryName = new CountryName() { Country = "Iran", City = "Yazd" };
                break;
            }
        }
        ViewBag.Location = CountryName;
        XNamespace yWeatherNS = "http://xml.weather.yahoo.com/ns/rss/1.0";
        List<YahooWeatherRssItem> WeatherList = new List<YahooWeatherRssItem>();
        for (int i = 0; i < 4; i++)
        {
            YahooWeatherRssItem YahooWeatherRssItem = new YahooWeatherRssItem()
            {
                Code = Convert.ToInt32(rssXml.Descendants("item").Elements(yWeatherNS +
"forecast").ElementAt(i).Attribute("code").Value),
                Day = rssXml.Descendants("item").Elements(yWeatherNS +
"forecast").ElementAt(i).Attribute("day").Value,
                Low = rssXml.Descendants("item").Elements(yWeatherNS +
"forecast").ElementAt(i).Attribute("low").Value,
                High = rssXml.Descendants("item").Elements(yWeatherNS +
"forecast").ElementAt(i).Attribute("high").Value,
                Text = rssXml.Descendants("item").Elements(yWeatherNS +
"forecast").ElementAt(i).Attribute("text").Value,
            };
            WeatherList.Add(YahooWeatherRssItem);
        }
    }
}

```

ایجاد سیستم وضعیت آب و هوا مانند گوگل (بخش اول)

```
        ViewBag.FeedList = WeatherList;
    }

    return PartialView("_Weather");
}
```

قسمت `SwitchCase`, `مقدار` و `Value` مربوط به هر آیتم `DropDown` را که شامل یک اسم شهر است، میگیرد و `RSS` مربوط به آن شهر را بر میگرداند.
حالا کد مربوط به خواندن فایل `Rss` را برایتان توضیح می دهم : حلقه `for` 0 تا 4 (که در کد بالا مشاهده می کنید) یعنی اطلاعات 4 روز آینده را برایم برگردان.
من تگ های `Code` , `Day` , `Low` , `High` و `text` فایل `RSS` را در این حلقه `For` می خوانم که البته مقادیر این 4 روز را در لیستی اضافه می کنم که نوع این لیست هم از نوع `YahooWeatherRssItem` می باشد. من این کلاس را در فایل ضمیمه قرار دادم. اکنون هر کدام از این تگ ها را برایتان توضیح می دهم:

`code` : هر آب و هوای کدی دارد. مثلا آب و هوای نیمه ابری یک کد ، آب و هوای آفتابی کدی دیگر و ...
`Low`: حداقل دمای آن روز را به ما می دهد .
`High`: حداکثر دمای آن روز را به ما می دهد .
`day`: نام روز از هفته را بر میگرداند مثلا شنبه ، یکشنبه و
`text`: که توضیحاتی می دهد مثلا اگر هوای آفتابی باشد مقدار `sunny` را بر میگرداند و ...

خوب، تا اینجا ما `Rss` مربوط به هر شهر را خواندیم حالا در قسمت `Design` باید چکار کنیم .
کدهای `html` صفحه `Index` ما شامل کدهای زیر است :

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

<link href("~/Content/User/Weather/Weather.css" rel="stylesheet" />
@section scripts{
    <script src="@Url.Content("~/Scripts/jquery-1.6.2.min.js")" type="text/javascript"></script>
    <script src="@Url.Content("~/Scripts/jquery.unobtrusive-ajax.min.js")"
type="text/javascript"></script>
    <script type="text/javascript">
        $("#dpProvince").change(function () {
            $(this).parents("form").submit();
        });
    </script>
}
<h2>Weather</h2>
<div id="Progress">
    <img src("~/Images/User/Other/ajax-loader.gif" />
</div>
<div id="BoxContent"> @Html.Partial("_Weather")</div>

@using (Ajax.BeginForm(actionName: "ShowWeatherProvince", ajaxOptions: new AjaxOptions {
    UpdateTargetId = "BoxContent", LoadingElementId = "Progress", InsertionMode = InsertionMode.Replace }))
{
<div style="padding-top:15px;">
    <div style="float:left; width:133px; ">Select Your Province</div>
    <div style="float:left"> @Html.DropDownList("dpProvince", new
SelectList(ViewBag.ProvinceList, "Id", "Name"), "Select Your Province", new { @class =
"webUserDropDown", @style = "width:172px" })</div>
</div>
}
```

و کدهای `Partial Weather` که است به صورت زیر است:

```
@{
    List<Weather.YahooWeatherRssItem> Feeds = ViewBag.FeedList;
}
<div>
```

```

{@
    HtmlString StartTable = new HtmlString("<table class='WeatherTable' cellspacing='0'
cellpadding='0'><tbody><tr>");
    HtmlString EndTable = new HtmlString("</tr></tbody></table>");
    HtmlString StartTD = new HtmlString("<td>");
    HtmlString EndTD = new HtmlString("</td>");
}
<div style="width: 300px;">
@{
    @StartTable
    foreach (var item in Feeds)
    {
        @StartTD
        <div>@item.Day</div>
        <div>
            @{
                string FileName = "";
                switch (item.Code)
                {
                    case 0: { FileName = "/Images/User/Weather/Tornado.png"; break; }
                    case 1: { FileName = "/Images/User/Weather/storm2.gif"; break; }
                    case 2: { FileName = "/Images/User/Weather/storm2.gif"; break; }
                    case 3: { FileName = "/Images/User/Weather/storm2.gif"; break; }
                    case 4: { FileName = "/Images/User/Weather/15.gif"; break; }
                    case 5: { FileName = "/Images/User/Weather/29.gif"; break; }
                    case 6: { FileName = "/Images/User/Weather/29.gif"; break; }
                    case 7: { FileName = "/Images/User/Weather/29.gif"; break; }
                    case 8: { FileName = "/Images/User/Weather/26.gif"; break; }
                    case 9: { FileName = "/Images/User/Weather/drizzle.png"; break; }
                    case 10: { FileName = "/Images/User/Weather/26.gif"; break; }
                    case 11: { FileName = "/Images/User/Weather/18.gif"; break; }
                    case 12: { FileName = "/Images/User/Weather/18.gif"; break; }
                    case 13: { FileName = "/Images/User/Weather/19.gif"; break; }
                    case 14: { FileName = "/Images/User/Weather/19.gif"; break; }
                    case 15: { FileName = "/Images/User/Weather/19.gif"; break; }
                    case 16: { FileName = "/Images/User/Weather/22.gif"; break; }
                    case 17: { FileName = "/Images/User/Weather/Hail.png"; break; }
                    case 18: { FileName = "/Images/User/Weather/25.gif"; break; }
                    case 19: { FileName = "/Images/User/Weather/dust.png"; break; }
                    case 20: { FileName = "/Images/User/Weather/fog_icon.png"; break; }
                    case 21: { FileName = "/Images/User/Weather/hazy_icon.png"; break; }
                    case 22: { FileName = "/Images/User/Weather/2017737395.png"; break; }
                    case 23: { FileName = "/Images/User/Weather/32.gif"; break; }
                    case 24: { FileName = "/Images/User/Weather/32.gif"; break; }
                    case 25: { FileName = "/Images/User/Weather/31.gif"; break; }
                    case 26: { FileName = "/Images/User/Weather/7.gif"; break; }
                    case 27: { FileName = "/Images/User/Weather/38.gif"; break; }
                    case 28: { FileName = "/Images/User/Weather/6.gif"; break; }
                    case 29: { FileName = "/Images/User/Weather/35.gif"; break; }
                    case 30: { FileName = "/Images/User/Weather/7.gif"; break; }
                    case 31: { FileName = "/Images/User/Weather/33.gif"; break; }
                    case 32: { FileName = "/Images/User/Weather/1.gif"; break; }
                    case 33: { FileName = "/Images/User/Weather/34.gif"; break; }
                    case 34: { FileName = "/Images/User/Weather/2.gif"; break; }
                    case 35: { FileName = "/Images/User/Weather/freezing_rain.png"; break; }
                    case 36: { FileName = "/Images/User/Weather/30.gif"; break; }
                    case 37: { FileName = "/Images/User/Weather/15.gif"; break; }
                    case 38: { FileName = "/Images/User/Weather/15.gif"; break; }
                    case 39: { FileName = "/Images/User/Weather/15.gif"; break; }
                    case 40: { FileName = "/Images/User/Weather/12.gif"; break; }
                    case 41: { FileName = "/Images/User/Weather/22.gif"; break; }
                    case 42: { FileName = "/Images/User/Weather/22.gif"; break; }
                    case 43: { FileName = "/Images/User/Weather/22.gif"; break; }
                    case 44: { FileName = "/Images/User/Weather/39.gif"; break; }
                    case 45: { FileName = "/Images/User/Weather/thundershowers.png"; break; }
                    case 46: { FileName = "/Images/User/Weather/19.gif"; break; }
                    case 47: { FileName = "/Images/User/Weather/thundershowers.png"; break; }
                    case 3200: { FileName = "/Images/User/Weather/1211810662.png"; break; }
                }
            }
            <img alt='@item.Text' title='@item.Text' src='@FileName'>
        </div>
        <div>
            <span>@item.High°</span>
            <span>@item.Low°</span>
        </div>
        @EndTD
    }
}
@EndTable
</div>

```

</div>

من عکس‌های مربوط به وضعیت آب و هوا را در فایل ضمیمه قرار دادم.
چنانچه در مورد RSS وضعیت آب و هوای یاهو اطلاعات دقیق‌تری را می‌خواهید بدانید به این [لينك](#) بروید.
در آموزش بعدی قصد دارم برایتان این بخش را توضیح دهم که بر اساس IP بازدید کننده سایت شما، اطلاعات آب و هوای شهر بازدید کننده را برایش در سایت نمایش دهد.

[Files-06bf65bac63d4dd694b15fc24d4cb074.zip](#)

موفق باشید

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۱۱ ۰:۵

ممnon از شما. چند نکته جزئی در مورد بهبود کیفیت کدها. در View‌های MVC باید کدنویسی صرفا به نمایش اطلاعات View خلاصه شود. یعنی switch داخل آن بهتر است تبدیل به یک extension method شده و نهایتاً استفاده شود (برای تمیز کردن View). همچنین استفاده از 1 و 2 switch و 10000 اصطلاحاً به magic code مشهور هستند. یعنی مشخص نیست معنای این اعداد چی هست. این‌ها را عموماً بهتر است تبدیل به enum کرد و بعد استفاده نهایی.

نویسنده: شقایق اشتربی
تاریخ: ۱۳۹۲/۱۲/۱۲ ۸:۳۹

ممnon از شما که مشکل کد نویسی من را گفتید .extension method helper‌ها منظورتون همون هاست ؟ لینکی دارید که در مورد این extension method آموزشی داده باشد چون من می‌خوام کدهامو تا جایی که می‌شود بهینه بنویسم .

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۲/۱۲/۱۲ ۹:۱۵

یک مثال در [اینجا](#)
قسمت آخر (If‌های شرطی را در View‌ها را در متدهای کمکی کپسوله کنید)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۱۲ ۹:۵۱

قسمت switch ای را که در View نوشتشد، تبدیل کنید به یک متدهای کمکی خارج از View: (مهم نیست متدهای مخصوص باشد یا خیر؛ فقط داخل View نباشد)

```
public static string GetFileName(int code)
{
    switch (code)
    {
        case 0: return "/Images/User/Weather/Tornado.png";
    //...
}
```

بعد یک خاصیت محاسباتی به نام FileName به مدل مورد استفاده اضافه کنید:

```
public class YahooWeatherRssItem
{
    public int Code { get; set; }
    //...
    public string FileName
    {
        get { return Util.GetFileName(Code); }
    }
}
```

به این صورت View از کدهای محاسبات یافتن FileName خالی می‌شود.

نویسنده: علی
تاریخ: ۱۳۹۲/۱۲/۱۲ ۱۰:۷

یه راه دیگه هم استفاده از Yahoo Query Language هست، می‌توانید با کوئری زیر اطلاعات آب و هوای شهر مورد نظر رو در قالب JSON دریافت کنید

ایجاد سیستم وضعیت آب و هوا مانند گوگل (بخش اول)

```
select * from weather.forecast where woeid in (select woeid from geo.placefinder where text="CityName")
```

نویسنده: شقایق اشتربی
تاریخ: ۱۳۹۲/۱۲/۱۰:۱۳

یه سوال دیگه که داشتم اگر بین کدهایمان ، کد html وجود داشت باز هم باید آنها را از view جدا کنیم و به صورت extension بنویسیم؟

یا اینکه جداسازی کدها از View را زمانی انجام می‌دهیم که چندین خط کد پشت سر هم نوشته باشیم و بین آنها کد html بی نباشد ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۱۰:۴۱

- نه الزاماً. اگر چندین جا استفاده و تکرار می‌شود یا منطق طولانی دارد، روش « [نوشتن HTML Helpers ویژه، به کمک امکانات Razor](#) » می‌تواند مفید باشد.

- منطق قرار گرفته در View فقط باید کار «نهایی» نمایشی را انجام دهد. اگر در View مثلا مستقیماً با دیتابیس کار می‌کنید، محاسبات مرتبط با فیلد خاصی را انجام می‌دهید و کلا هر منطقی که «نهایی» بودن نمایش اطلاعات را زیر سؤال ببرد، باید از View جدا شده و به کنترلر و زیرساخت آن منتقل شود.

نویسنده: محسن تقی پور
تاریخ: ۱۳۹۲/۱۲/۲۰:۸

میشه راجبه این روش کمی بیشتر توضیح بدین

نویسنده: Anaritius
تاریخ: ۱۳۹۲/۱۲/۲۱:۴۴

سلام
ممnon از مقاله خوبتون
اماکنش هست سورس کد webForm ش رو هم بذارید و اسه دانلود؟
ممnon

نویسنده: علی
تاریخ: ۱۳۹۲/۱۲/۲۱:۸

[اینجا](#) یک مثال از نحوه کار با Linq رو توضیح داده

نویسنده: ع پارسایی
تاریخ: ۱۳۹۳/۱۱/۱۸:۳۶

با سلام

امکان داره بگید در اکشن، قسمت ViewBag.ProvinceList = _RPosition.Positions چه کار میکنه و RPosition کارش چیه؟

نویسنده: ع پارسایی
تاریخ: ۱۳۹۳/۱۱/۲۱:۳۴

با سلام؛ در مطلب گفته شده برای گرفتن WOEID شهرها به فلان سایت بروید و هر کشوری رو بخواهیم باید بریم و دستی واردش اما من دیتابیس یا هر چیز دیگری رو میخوام که قابلیت این رو داشته باشه که کاربر با وارد کردن کشور و شهر بتونه آب و هوا رو

بینه، شما میتوانید راهنمایی ام کنید که چه کار باید بکنم و یا منبعی برای دیتابیس مختصات جغرافیایی کشورها وجود دارد؟ ممنون میشم کمک کنید. با تشکر

نوبسند: شقایق اشتربی
تاریخ: ۲:۱۳ ۱۳۹۳/۱۱/۲۶

سلام ، نام استان‌ها را در ViewBag قرار دادم.

نوبسند: شقایق اشتربی
تاریخ: ۲:۱۴ ۱۳۹۳/۱۱/۲۶

به دو لینک زیر نگاهی بیندازید:
[/https://developer.yahoo.com/yql](https://developer.yahoo.com/yql)
[/http://dev.maxmind.com/geoip/legacy/geolite](http://dev.maxmind.com/geoip/legacy/geolite)

اگر به دو مطلب استفاده از Quartz.NET و Quartz استفاده آن دقت کرده باشید به این نتیجه خواهید رسید که ... این کتابخانه‌ی در اصل جاوا این گنج طراحی شده است. در سایت جاری برای انجام کارهای زمانبندی شده (مانند ارسال ایمیل‌های روزانه خلاصه مطالب، تهیه خروجی PDF و XML سایت، تبدیل پیش‌نویس‌ها به مطالب، بازسازی ایندکس‌های جستجو و امثال آن) از یک Thread timer استفاده می‌شود که حجم نهایی کتابخانه‌ی محصور کننده و مدیریت کننده وظایف آن جملاً ۸ کیلوبایت است؛ متشکل از ... سه کلاس. در ادامه کدهای کامل و نحوه استفاده از آن را بررسی خواهیم کرد.

دریافت کتابخانه DNT Scheduler و مثال آن

[DNTScheduler.7z](#)

در این بسته، کدهای کتابخانه‌ی DNT Scheduler و یک مثال وب فرم را، ملاحظه خواهید کرد. از این جهت که برای ثبت وظایف این کتابخانه، از فایل global.asax.cs استفاده می‌شود، اهمیتی ندارد که پروژه‌ی شما وب فرم است یا MVC. با هر دو حالت کار می‌کند.

نحوه تعریف یک وظیفه‌ی جدید

کار با تعریف یک کلاس و پیاده سازی ScheduledTaskTemplate شروع می‌شود:

```
public class SendEmailsTask : ScheduledTaskTemplate
```

برای نمونه :

```
using System;
namespace DNTScheduler.TestWebApplication.WebTasks
{
    public class SendEmailsTask : ScheduledTaskTemplate
    {
        /// <summary>
        /// خاصیت ترتیب اجرای آنها را مشخص خواهد کرد
        /// </summary>
        public override int Order
        {
            get { return 1; }
        }

        public override bool RunAt(DateTime utcNow)
        {
            if (this.IsShuttingDown || this.Pause)
                return false;

            var now = utcNow.AddHours(3.5);
            return now.Minute % 2 == 0 && now.Second == 1;
        }

        public override void Run()
        {
            if (this.IsShuttingDown || this.Pause)
                return;

            System.Diagnostics.Trace.WriteLine("Running Send Emails");
        }

        public override string Name
        {
            get { return "ارسال ایمیل"; }
        }
    }
}
```

- در اینجا Order، ترتیب اجرای وظیفه‌ی جاری را در مقایسه با سایر وظیفه‌هایی که قرار است در یک زمان مشخص اجرا شوند، مشخص می‌کند.

- متدهای RunAt ثانیه‌ای یکبار فرآخوانی می‌شود (بنابراین بررسی now را فراموش نکنید). زمان ارسالی به آن UTC است و اگر برای نمونه می‌خواهید بر اساس ساعت ایران کار کنید باید 3.5 ساعت به آن اضافه نمائید. این مساله برای سرورهایی که خارج از ایران قرار دارند مهم است. چون زمان محلی آن‌ها برای تصمیم‌گیری در مورد زمان اجرای کارها مفید نیست.

در متدهای RunAt فرصت خواهید داشت تا منطق زمان اجرای وظیفه‌ی جاری را مشخص کنید. برای نمونه در مثال فوق، این وظیفه هر دو دقیقه یکبار اجرا می‌شود. یا اگر خواستید اجرای آن فقط در سال 23 و 33 دقیقه هر روز باشد، تعریف آن به نحو ذیل خواهد بود:

```
public override bool RunAt(DateTime utcNow)
{
    if (this.IsShuttingDown || this.Pause)
        return false;

    var now = utcNow.AddHours(3.5);
    return now.Hour == 23 && now.Minute == 33 && now.Second == 1;
}
```

- خاصیت IsShuttingDown موجود در کلاس پایه ScheduledTaskTemplate، توسط کتابخانه‌ی DNT Scheduler مقدار دهنده می‌شود. این کتابخانه قادر است زمان خاموش شدن پروسه‌ی فعلی IIS را تشخیص داده و خاصیت IsShuttingDown را کند. بنابراین در حین اجرای وظیفه‌ای مشخص، به مقدار IsShuttingDown دقت داشته باشید. اگر true شد، یعنی فقط 30 ثانیه وقت دارید تا کار را تمام کنید.

خاصیت Pause هر وظیفه را برنامه می‌تواند تغییر دهد. به این ترتیب در مورد توقف یا ادامه‌ی یک وظیفه می‌توان تصمیم‌گیری کرد. خاصیت ScheduledTasksCoordinator.Current.ScheduledTasks می‌دهد.

- در متدهای Run، منطق وظیفه‌ی تعریف شده را باید مشخص کرد. برای مثال ارسال ایمیل یا تهیه‌ی بک آپ.

- نیز نام وظیفه‌ی جاری است که می‌تواند در گزارشات مفید باشد.

همین مقدار برای تعریف یک وظیفه کافی است.

نحوه‌ی ثبت و راه اندازی وظایف تعریف شده

پس از اینکه چند وظیفه را تعریف کردیم، برای مدیریت بهتر آن‌ها می‌توان یک کلاس ثبت و معرفی کلی را مثلاً به نام ScheduledTasksRegistry ایجاد کرد:

```
using System;
using System.Net;

namespace DNTScheduler.TestWebApplication.WebTasks
{
    public static class ScheduledTasksRegistry
    {
        public static void Init()
        {
            ScheduledTasksCoordinator.Current.AddScheduledTasks(
                new SendEmailsTask(),
                new DoBackupTask());

            ScheduledTasksCoordinator.Current.OnUnexpectedException = (exception, scheduledTask) =>
            {
                //todo: log the exception.
                System.Diagnostics.Trace.WriteLine(scheduledTask.Name + ":" + exception.Message);
            };

            ScheduledTasksCoordinator.Current.Start();
        }

        public static void End()
        {
        }
    }
}
```

```
        ScheduledTasksCoordinator.Current.Dispose();
    }

    public static void WakeUp(string pageUrl)
    {
        try
        {
            using (var client = new WebClient())
            {
                client.Credentials = CredentialCache.DefaultNetworkCredentials;
                client.Headers.Add("User-Agent", "ScheduledTasks 1.0");
                client.DownloadData(pageUrl);
            }
        }
        catch (Exception ex)
        {
            //todo: log ex
            System.Diagnostics.Trace.WriteLine(ex.Message);
        }
    }
}
```

- شیء ScheduledTasksCoordinator.Current، نمایانگر تنها و هلهی مدیریت وظایف برنامه است.
 - توسط متد ScheduledTasksCoordinator.Current.AddScheduledTasks، تنها کافی است کلاس‌های وظایف مشتق شده از ScheduledTaskTemplate معرفی شوند.
 - به کمک متد ScheduledTasksCoordinator.Current.Start Thread timer برنامه شروع می‌شود.
 - اگر در حین اجرای متد Run، استثنایی رخ دهد، آنرا توسط یک Action delegate به نام Scheduler.OnUnexpectedException می‌توانید دریافت کنید. کتابخانه‌ی DNT برای اجرای وظایف، از یک ترد با سطح تقدم Below normal استفاده می‌کند تا در حین اجرای وظایف، برنامه‌ی جاری با اخلال و کندی مواجه نشده و بتواند به درخواست‌های رسیده پاسخ دهد. در این بین اگر استثنایی رخ دهد، می‌تواند کل پروsesه‌ی IIS را خاموش کند. به همین جهت این کتابخانه کار try/catch استثناهای متد Run را نیز انجام می‌دهد تا از این لحاظ مشکلی نباشد.
 - متد ScheduledTasksCoordinator.Current.Dispose کار مدیر وظایف برنامه را خاتمه می‌دهد.
 - از متد WakeUp تعریف شده می‌توان برای پیدار کردن مجدد برنامه استفاده کرد.

استفاده از کلاس `ScheduledTasksRegistry` تعریف شده

پس از اینکه کلاس `ScheduledTasksRegistry` را تعریف کردیم، نیاز است آن را به فایل استاندارد `global.asax.cs` برنامه به نحو ذیل معرفی کنیم:

```
using System;
using System.Configuration;
using DNTScheduler.TestWebApplication.WebTasks;

namespace DNTScheduler.TestWebApplication
{
    public class Global : System.Web.HttpApplication
    {
        protected void Application_Start(object sender, EventArgs e)
        {
            ScheduledTasksRegistry.Init();
        }

        protected void Application_End()
        {
            ScheduledTasksRegistry.End();
            // نکته مهم این روش نیاز به سرویس پینگ سایت برای زنده نگه داشتن آن است
            ScheduledTasksRegistry.WakeUp(ConfigurationManager.AppSettings["SiteRootUrl"]);
        }
    }
}
```

- متد `ScheduledTasksRegistry.Init` در حین آغاز برنامه فراخوانی می‌شود.
 - متد `ScheduledTasksRegistry.Finish` در بایان کار ب‌نامه جفت باکساز، منابع باید فراخوان، گردد.

همچنین در اینجا با فرآخوانی `ScheduledTasksRegistry.WakeUp`، می‌توانید برنامه را مجدداً زنده کنید! IIS مجاز است یک سایت ASP.NET را پس از مثلاً 20 دقیقه عدم فعالیت (فعالیت به معنای درخواست‌های رسیده به سایت است و نه کارهای پس زمینه)، از حافظه خارج کند (این عدد در `application pool` برنامه [قابل تنظیم است](#))。 در اینجا در فایل `web.config` برنامه می‌توانید آدرس یکی از صفحات سایت را برای فرآخوانی مجدد تعريف کنید:

```
<?xml version="1.0"?>
<configuration>
  <appSettings>
    <add key="SiteRootUrl" value="http://localhost:10189/Default.aspx" />
  </appSettings>
</configuration>
```

همینکه درخواست مجددی به این صفحه برسد، مجدداً برنامه توسط IIS بارگذاری شده و اجرا می‌گردد。به این ترتیب وظایف تعريف شده، در طول یک روز بدون مشکل کار خواهند کرد。

گزارشگیری از وظایف تعريف شده

برای دسترسی به کلیه وظایف تعريف شده، از خاصیت `ScheduledTasksCoordinator.Current.ScheduledTasks` استفاده نمائید:

```
var jobsList = ScheduledTasksCoordinator.Current.ScheduledTasks.Select(x => new
{
  TaskName = x.Name,
  LastRunTime = x.LastRun,
  LastRunWasSuccessful = x.IsLastRunSuccessful,
  IsPaused = x.Pause,
}).ToList();
```

لیست حاصل را به سادگی می‌توان در یک `Grid` نمایش داد.

نظرات خوانندگان

نویسنده: محمد رعیت پیشه
تاریخ: ۱۳۹۲/۱۲/۲۶ ۱۹:۲۲

آیا یکی از کاربردهای این مطلب می‌توانه مثلاً ارسال یک ایمیل یک هفته قبل از اتمام زمان شارژ کاربری باشد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۶ ۱۹:۳۷

بله. می‌توانید یک وظیفه‌ی جدید تعریف کنید که هر شب ساعت مثلاً ۱۱ و ۱۵ دقیقه اجرا شود (نحوه‌ی تعریف متده‌ی RunAt). سپس در متده‌ی Run آن یک کوئری از دیتابیس گرفته، لیست موارد مدنظر را واکشی کرده و به آن‌ها ایمیل بزنید.

نویسنده: پیمان مهریانی
تاریخ: ۱۳۹۲/۱۲/۲۷ ۱۱:۴۷

کتابخانه سبکی بود اما نگرانی‌هایی مانند پیش آمدن همزمانی در اجرای وظایف را دارد. پیش از این از کتابخانه quartz scheduler در یک پروژه بزرگ استفاده کرده بودیم و نتیجه کار بسیار راضی کننده بود، حتی می‌توان از Scheduler خود ویندوز و یا Jon‌های SQL Server هم بهره برد.

این روش چه مزایا و معایبی نسبت به روش‌های موجود دارد و آیا توصیه شده برای استفاده در پروژه‌های بزرگ هست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۷ ۱۲:۲

- «پیش آمدن همزمانی در اجرای وظایف» خاصیت Order را برای وظایفی که قرار است در یک زمان مشخص اجرا شوند، مقدار دهی کنید. ۱ و ۲ و ۳ و الی آخر.

- «حتی می‌توان از Scheduler خود ویندوز و یا Jon‌های SQL Server هم بهره برد».

بله. به شرطی که سرور در اختیار شما باشد و [دسترسی کافی برای انجام اینکار را](#) داشته باشید. البته در این حالت خاص، مدیریت آن پیکارچه با یک برنامه‌ی وب نیست.

در سرورهای اشتراکی روش ارائه شده در این مطلب بدون نیاز به سطح دسترسی خاصی کار می‌کند. ضمناً برای ASP.NET نوشته شده است و این قابلیت را دارد که به شما اعلام کند مثلاً ۳۰ ثانیه دیگر برنامه از سرور unload می‌شود؛ توسط خاصیت IsShuttingDown.

- «آیا توصیه شده برای استفاده در پروژه‌های بزرگ هست؟»

یک به اشتراک گذاری بود از قسمتی از کدهای زیر ساخت سایت حاری که هم اکنون مورد استفاده است (مقدمه بحث).

نویسنده: سلمان کاظمی
تاریخ: ۱۳۹۲/۱۲/۲۸ ۱:۳۲

سوالی که واسه من پیش اومنده اینه که من یک نرم افزار Web Form دارم . هدفم اینه که یک روز قبل از تاریخ تولد اعضا ایمیلی با عنوان تولدتان مبارک ارسال شه . حالا من باید این کدهای گفته شده را در برنامه خود بیاورم یا یک وب سرویس بنویسم که اینکارو انجام بد؟ اگه بخواهم که در برنامه انجام بشه خوب کجا باید نوشت ؟ یا به عبارتی دیگه باید حتماً برنامه اجرا بشه و اگه برنامه اصلاً Run نشه چی میشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۸ ۸:۵۹

- وب سرویس فقط با یک درخواست رسیده کار می‌کند. کار کتابخانه‌ی فوق، اجرا در پس زمینه‌ی برنامه به صورت مداوم است.
- فقط دو حالت وجود دارد که برنامه اجرا نشود:

- الف) `protected void Application_End` فراخوانی شود. متدهای `WakeUp` نوشته شده برای این منظور و راه اندازی مجدد برنامه توسط آن، کافی است.
- ب) کل سرور ری استارت شود (نه فقط برنامه). در این حالت کافی است آدرس برنامه را به یکی از سرویس‌هایی که هر از چندگاهی برنامه را ping می‌کنند، معرفی کنید.

نویسنده: امیرحسین جلوداری
تاریخ: ۲۳:۲۳ ۱۳۹۳/۰۱/۰۱

خیلی خیلی ممنون ... کتابخانه‌ی ساده و مفیدی است.
سوال من اینه که میشه کاری کرد که آدرس روت تعیین شده در فایل کانفیگ چهت بیدار شدن IIS را در کد نوشت مثل کد زیر :

```
var urlToWakeUp = Request.Url.Scheme + Uri.SchemeDelimiter + Request.Url.Host +  
(Request.Url.IsDefaultPort ? ":" : ":" + Request.Url.Port);
```

که مجبور نباشی در هر سایتی که طراحی میکنیم آدرس رو در فایل کانفیگ سست کنیم !
پی نوشت : من این کار رو در متدهای `Application_End` نوشتیم و خطای در دسترس نبودن `Request` رو دریافت کردم ! (هر چند که میدونستم این خطای را میده !)

نویسنده: وحید نصیری
تاریخ: ۰۵:۵۱ ۱۳۹۳/۰۱/۰۲

می‌شود در `Application_BeginRequest` اطلاعات آدرس ریشه سایت را در یک متغیر استاتیک ذخیره کرد. مقدار آن در `Application_End` قابل استفاده است:

```
namespace TestApp  
{  
    public static class App  
    {  
        public static string SiteRootUrl;  
    }  
  
    public class TestApplication : HttpApplication  
    {  
        protected void Application_BeginRequest(object sender, EventArgs e)  
        {  
            if (string.IsNullOrWhiteSpace(App.SiteRootUrl))  
            {  
                App.SiteRootUrl = Request.Url.GetLeftPart(UriPartial.Authority) +  
Request.ApplicationPath;  
            }  
        }  
  
        protected void Application_End()  
        {  
            // use App.SiteRootUrl  
        }  
    }  
}
```

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳:۲ ۱۳۹۳/۰۱/۰۲

خیلی خیلی ممنون :) ... اگه میشه نسخه‌ی nuget کتابخانه را هم بذارید

نویسنده: مهدی پایرونده
تاریخ: ۱۴:۳۱ ۱۳۹۳/۰۱/۲۵

1. برای زنده نگهداشتن سایت میشه از خود همین کتابخانه استفاده کرد؟
 2. برای چه تعداد جاب بنظرتون منطقیه که استفاده بشه؟
- ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۲۵ ۱۶:۴۳

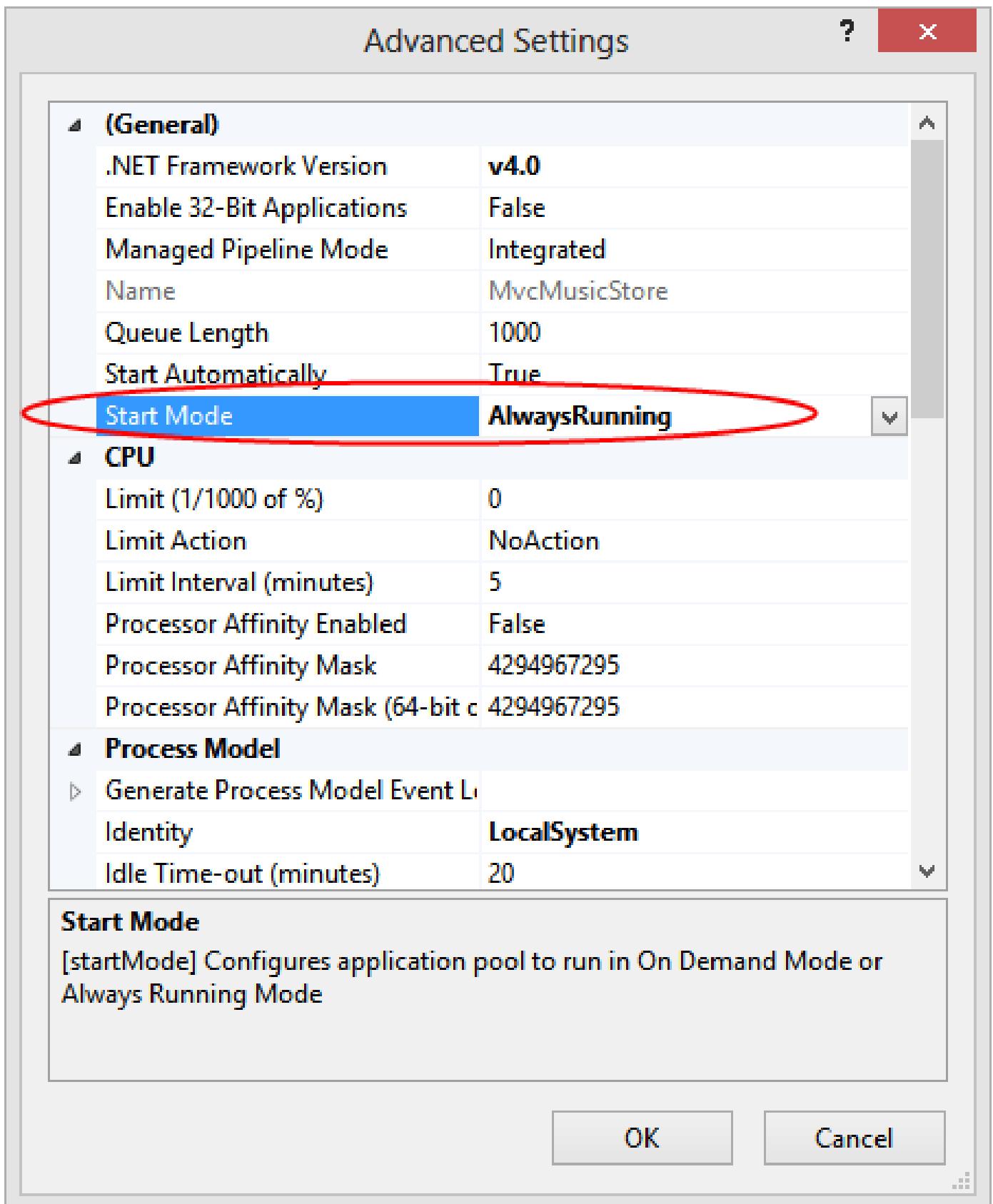
- کمی بالاتر توضیح دادم «فقط دو حالت وجود دارد که برنامه اجرا نشود: ...»
- این مورد فقط بستگی به توان سرور شما دارد.

نویسنده: fss
تاریخ: ۱۳۹۳/۰۱/۲۸ ۹:۶

در حالتی که "کل سرور ری استارت شود (نه فقط برنامه)"، بعد از بالا آمدن سرور و شروع به کار IIS، تابع Application_Start اجرا نمی‌شود؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۲۸ ۹:۱۴

نه تا زمانیکه اولین درخواستی به برنامه برسد.
کل سرور ری استارت شده IIS برنامه را فقط زمانی مجددا بارگذاری می‌کند که درخواست نمایش یکی از قسمت‌های سایت به آن ارسال شود.
البته IIS‌های جدید قابلیت Auto-Start هم دارند؛ ولی باید در تنظیمات Application pool برنامه انتخاب شود:



همچنین نیز برای اجرا خودکار برنامه پس از ریاستارت سرور [طراحی شده](#).

نویسنده: وحید نصیری
تاریخ: ۹:۶ ۱۳۹۳/۰۲/۱۷

یک نکته‌ی تکمیلی

در دات نت 4.5.2 ، متدهی به نام `HostingEnvironment.QueueBackgroundWorkItem` اضافه شده است تا درخواست اجرای کارهای پس زمینه در ASP.NET به سادگی و همچنین با اطمینان بیشتری قابل انجام باشد.

نویسنده: مهرداد
تاریخ: ۱۳:۴۰ ۱۳۹۳/۰۲/۱۷

یعنی با استفاده از این متدهی که در دات نت 4.5.2 اضافه شده ، دیگه نیازی به Quartz.net و یا DNT scheduler نیست ؟ و میتوان برای کارهایی که نیاز به زمانبندی دارند از این متدهی استفاده کرد ؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۴۲ ۱۳۹۳/۰۲/۱۷

فقط قسمت مدیریت تردهای آن با یک سطر `QueueBackgroundWorkItem` جایگزین می‌شود. مابقی قسمتهای آن (مانند اینکه یک `Queue` چه زمانی در `WorkItem` قرار گیرد) تفاوتی نمی‌کند و مانند قبل است.

نویسنده: مهرداد
تاریخ: ۱۴:۱۰ ۱۳۹۳/۰۲/۱۷

منظور شما این خط هست ؟

```
ScheduledTasksCoordinator.Current.AddScheduledTasks(  
    new SendEmailsTask(),  
    new DoBackupTask());
```

نویسنده: وحید نصیری
تاریخ: ۱۴:۲۳ ۱۳۹۳/۰۲/۱۷

این‌ها باقی خواهند ماند. قسمت `new Thread` و مدیریت آن جایگزین می‌شود.

نویسنده: وحید نصیری
تاریخ: ۹:۲۵ ۱۳۹۳/۰۲/۲۹

یک نکته

مثالی از نحوه استفاده از متدهی `QueueBackgroundWorkItem`
+ یک مثال رسمی

نویسنده: Aria
تاریخ: ۲۳:۱۰ ۱۳۹۳/۰۳/۱۹

با سلام

من روی یه پروژه مالی دارم با ASP WebForm کار می‌کنم و میخواستم در اول هر ماه یک سری دستورات ویرایش یا `update` را که به صورت رکورد در یک جدول ذخیره می‌کنم انجام بشن. سوالم اینه که برای این که مجموعه دستورات من انجام بشن باید برنامه جتما `Run` باشه و چه جوری میشه اوно به صورت پس زمینه در حال اجرا نگه داشت.

با تشکر

نویسنده: وحید نصیری
تاریخ: ۰:۱۹ ۱۳۹۳/۰۳/۲۰

- به اندازه کافی در نظرات این بحث در مورد زنده نگه داشتن یک برنامه ASP.NET بحث شده. کمی وقت بگذارید و آن‌ها را مطالعه کنید.

+ اگر برنامه مالی است، احتمالاً دسترسی کاملی به سرور و همچنین SQL Server (اگر با آن کار می‌کنید) دارید. در این حالت برای به روز رسانی زمانبندی شده‌ی چند رکورد شاید بهتر باشد از سرویس معروف و همیشه در حال اجرای [SQL Server agent](#) استفاده کنید. در اینجا نیز می‌شود یک job را که متشکل از دستورات T-SQL است، در فواصل زمانی مشخصی اجرا کرد.

نویسنده: میثم
تاریخ: ۱۳۹۳/۰۳/۲۵ ۱۳:۴۸

سلام.

منظورتون فقط همین یک خط توی متدهای [ScheduledTasksCoordinator](#) کلاس Start بود که تغییر میکنه؟
یعنی فقط متدهای Start به شکل زیر بازنویسی میشه دیگه؟

```
public void Start()
{
    timer.OnTimerCallback = () =>
    {
        var now = DateTime.UtcNow;
        var taskToRun = _tasks.Where(x => !x.IsRunning && x.RunAt(now)).OrderBy(x => x.Order).ToList();
        if (_isShuttingDown || !taskToRun.Any())
            return;

        HostingEnvironment.QueueBackgroundWorkItem(x => taskAction(taskToRun));
    };
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۲۵ ۱۵:۱

- بله. قسمتهای [IRegisteredObject](#) و [HostingEnvironment.RegisterObject](#) آن هم باید حذف شوند چون در [QueueBackgroundWorkItem](#) وجود دارند و یک [CancellationToken](#) را تنظیم می‌کند.

+ زمانیکه از DNTScheduler استفاده می‌کنید، عمل نیازی به [QueueBackgroundWorkItem](#) ندارید. چون نکته‌ی [IRegisteredObject](#) و [HostingEnvironment.RegisterObject](#) در آن لحاظ شده. این نکته که خاموش شدن IIS را گزارش می‌کند، چند سال قبل، توسط یکی از اعضای قبلى تیم [ASP.NET منتشر شده بود](#). دقیقاً از همین نکته در استفاده شده. [QueueBackgroundWorkItem](#)

به صورت خلاصه، DNTScheduler با دات نت 4 به بعد سازگار است و نکات [QueueBackgroundWorkItem](#) دات نت 4.5.2 را به صورت توكار پیاده سازی کرده است.

نویسنده: حسین
تاریخ: ۱۳۹۳/۰۴/۱۷ ۱۱:۲

سلام. اگر بخواهیم یک کار نسبتاً زمانبر که IO هم هست را توسط این کتابخانه در فواصل زمانی معین اجرا کنیم، میشه از `async` و `await` استفاده کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۷ ۱۱:۲۰

بله. از روش و کتابخانه معرفی شده در مطلب «[استفاده از async و await در برنامه‌های کنسول و سرویس‌های ویندوز NT](#)» استفاده کنید.

نویسنده: حمید حسین وند
تاریخ: ۱۹/۰۵/۱۳۹۳

سلام

من برای این سری کارها از ویندوز سرویس استفاده می‌کنم. مثلاً ویندوز سرویس من از ساعت 8 صبح شروع به کار می‌کنه و رویدادهای مثل سالروز تولد رو با استفاده از پیامک به کاربران پیام تبریک ارسال می‌کنه. مهمترین عاملی که باعث شد من از ویندوز سرویس استفاده کنم اجرای مداوم و همیشگی بدون ارسال درخواست به وب سایت من بود. ولی فکر می‌کنم این کتابخانه شما هم مثل ویندوز سرویس عمل می‌کنه و خودش همیشه در حال اجراست. حالا به نظرتون آیا از ویندوز سرویس استفاده کنم بهتره و یا اینکه از این کتابخانه استفاده کنم؟ ممنون

نویسنده: محسن خان
تاریخ: ۰۹/۰۵/۱۳۹۳

فرض کن داری از یک هاست اشتراکی استفاده می‌کنی. دسترسی ادمین هم روی سرور نداری برای اجرا سرویس ویندوز یا فرض کن در یک سازمان بہت گفتن ما فقط اجازه می‌دیم فایل‌های سایت رو روی سرور کپی کنی. دسترسی بیشتری بہت نمی‌دیم. اون وقت چکار می‌کنی؟

نویسنده: Mohammad
تاریخ: ۱۵/۰۵/۱۳۹۳

سلام. ممنون.

دوستان برای من وقتی به صورت ساعت می‌درست کار نمی‌کنه. مثلاً من می‌خوام در ساعت دو و بیست و چهار دقیقه بامداد يه کاری رو انجام بده اینجوری نوشتم: در صورتی اگه بگم دو دقیقه دو دقیقه درست درست انجام می‌شه. به نظرتون مشکل از کجاست؟ ممنون

```
var now = utcNow.AddHours(3.5);
return now.Hour == 2 && now.Minute == 24 && now.Second == 1;
```

نویسنده: وحید نصیری
تاریخ: ۱۵/۰۵/۱۳۹۳

مشکل از تنظیم نبودن ساعت سرور است. مقدار `DateTime.UtcNow` را روی سرور بررسی کنید و با مقدار واقعی تطابق بدید. بعد اختلافش را باید در همینجا اعمال کنید. مثلاً اگر پس از بررسی متوجه شدید ساعت سرور یک ساعت عقب هست، `now.Hour == 2` می‌شود و امثال این نوع محاسبات.

نویسنده: علی
تاریخ: ۱۵/۰۶/۱۳۹۳

ممنون. مفید بود. اگه بخواهیم یه کاری مثلاً هر سه روز یکبار انجام بشه باید چه جوری زمان رو تعیین کنیم؟ ممنون

نویسنده: وحید نصیری
تاریخ: ۱۵/۰۶/۱۳۹۳

```
public override bool RunAt(DateTime utcNow)
{
    if (this.IsShuttingDown || this.Pause)
        return false;

    var now = utcNow.AddHours(3.5);
    return (now.Day % 3 == 0) && (now.Hour == 0 && now.Minute == 1 && now.Second == 1);
}
```

نویسنده: مهدی

تاریخ: ۲۲:۲۲ ۱۳۹۳/۰۶/۱۷

یه سوال: اختلاف زمانی ساعت گرینویچ با تهران همیشه ۳:۳۰ هست؟

نویسنده: وحید نصیری

تاریخ: ۲۳:۸ ۱۳۹۳/۰۶/۱۷

اگر سرور شما تنظیمات [daylight saving time](#) صحیحی داشته باشد، بله. اگر نه، خودتان این یک ساعت تفاوت را ۶ ماه یکبار باید محاسبه و اعمال کنید.

نویسنده: آرام

تاریخ: ۹:۵۷ ۱۳۹۳/۰۶/۱۸

سلام.

چرا هیچ ارجاعی به متد Stop در کلاس `JobsRunnerTimer` وجود نداره؟ برای این کار دلیل دارد؟

نویسنده: وحید نصیری

تاریخ: ۱۰:۱۴ ۱۳۹۳/۰۶/۱۸

نیازی نیست. چون طول عمر کل این مژول دقیقاً معادل طول عمر برنامه‌ی وب است. خاتمه‌ی آن هم به صورت خودکار با از حافظه خارج کردن `AppDomain` برنامه توسط IIS انجام می‌شود. تا زمانیکه برنامه در حال اجرا است این مژول هم به همین ترتیب. هر زمان که IIS تصمیم به خاتمه‌ی برنامه گرفت، نه این مژول، هیچ مژول دیگری هم فرصت مقاومت پیدا نمی‌کند و راسا به همراه `AppDomain` جاری خاتمه می‌یابد.

نویسنده: سیروان عفیفی

تاریخ: ۱۸:۴۳ ۱۳۹۳/۰۸/۰۷

ممnon، عالیه

فقط یه سوال:

چطور می‌توینیم یک Task خاص رو خودمون فراخوانی کنیم، مثلًا بعد از اینکه کاربر در سایت ثبت‌نام کرد به صورت هفتگی برآش ایمیل ارسال بشه یا مواردی از این دست، در مطلب بالا این طوری که من متوجه شدم، توسط دستور `Application_Start()`; در `ScheduledTasksRegistry.Init` تمام وظایف به صورت همزمان شروع می‌شون، در این حالت باید برای هر وظیفه یک کلاس برای ریجستر کردن اون بنویسیم؟ درسته؟

نویسنده: وحید نصیری

تاریخ: ۱۸:۴۸ ۱۳۹۳/۰۸/۰۷

بله. هر وظیفه نیاز به طراحی مجزا دارد. ثبت و معرفی آن توسط متد `ScheduledTasksCoordinator.Current.AddScheduledTasks` در `ScheduledTasksRegistry.Init` انجام می‌شود.

نویسنده: سیروان عفیفی

تاریخ: ۲۳:۵۰ ۱۳۹۳/۰۸/۰۷

ممnon،

آیا انجام چنین کاری توسط این کتابخانه میسر است؟
به عنوان مثال: ابتدا یک وظیفه را توسط `ScheduledTasksRegistry.Init` اجرا می‌کنم، این وظیفه قرار است به مدت ۶ روز، روزانه کاری را انجام دهد، بعد از پایان روز ۶ یعنی در روز هفتم یک وظیفه دیگر شروع خواهد شد (تعداد روزها توسط کاربر انجام می‌گیرد). و در نهایت وظایف یکی می‌شوند (بعد زمان تعیین شده برای آنها متوقف خواهند شد).

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۰۸

در متدهای RunAt بازه زمانی را مشخص کنید. در این مثال‌ها یک سری عدد ثابت است؛ شما آن‌ها را از دیتابیس بخوانید. فقط دقت داشته باشید که خروجی true/false این متدهای ثانیه‌ای یکبار بررسی می‌شود. بنابراین مباحث caching را جهت واکشی اطلاعات از بانک اطلاعاتی در متدهای RunAt اعمال کنید تا بجهت رفت و برگشت اضافی صورت نگیرد.

نویسنده: حسن
تاریخ: ۱۳۹۳/۰۸/۰۸ ۱:۴۵

سلام
ایا میشه از این کتابخانه به صورت ثانیه‌ای استفاده کرد؟ در هر ثانیه چک کنه و در صورت داشتن وظایف در این ساعت و دقیقه و ثانیه (که لیست اونها در دیتابیس هست) بهشون عمل کنه!
مثلاً اینطوری اگر کار کنه مشکل پیش نمیاد؟

```
return now.Second % 1 == 0 && now.Second == 1;
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۰۸ ۱:۵۴

برای این حالت در متدهای RunAt فقط بنویسید return true. چون این متدهای ثانیه‌ای یکبار فراخوانی می‌شود.

نویسنده: سعید شیرعلی
تاریخ: ۱۳۹۳/۰۹/۰۲ ۲۳:۳۳

سلام؛ فایلی که برای دانلود گذاشتین پر از اشکاله. وقتی بازش میکنم با asp زیر خیلی از کداش خط قرمز میکشه، مشکل از کجاست، آیا سورس مشکل داره؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۰۳ ۰:۲۸

دقیقاً چه خطایی گرفتید؟

نویسنده: سعید شیرعلی
تاریخ: ۱۳۹۳/۰۹/۰۳ ۱۰:۱۱

چند بار بازو بستش کردم خیلی از خطاهاش رفتم نمیدونم دلیلش چیه اما هنوز یک خط داره
خط اول Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"  
Inherits="DNTScheduler.TestWebApplication.Default" %>
```

خطا
Error 1 Could not load type 'DNTScheduler.TestWebApplication.Default'.
C:\Users\KING\Desktop\DNTScheduler\DNTScheduler.TestWebApplication\Default.aspx 1

نویسنده: سعید شیرعلی
تاریخ: ۱۳۹۳/۰۹/۰۳ ۱۰:۵۵

مشکل را حل کردم. وارد قسمت پروپرتبیس شدم و استارت رشو عوض کردم درست شد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۰۳ ۱۰:۵۵

- این پروژه باید یکبار کامل کامپایل شود و همچنین پوششی bin آن موجود باشد.
- خطای Could not load type صرفا به معنای عدم کامپایل شدن برنامه و یا نبود فایل‌های dll مرتبط در پوششی bin است. مراجعه کنید به منوی Build و گزینه‌ی Rebuild solution را انتخاب کنید.

نویسنده: سعید شیرعلی
تاریخ: ۱۳۹۳/۰۹/۰۳ ۱۲:۴۶

سلام؛ میخواهم هر 15 ثانیه یک بار کدم اجرا بشه. این کارو کردم اما نشد:

```
return now.Second % 15 == 0 && now.Second == 1;
```

لطفا راهنمایی کنید.

```
<add key="SiteRootUrl" value="http://localhost:10189/Default.aspx" />
```

«همینکه درخواست مجددی به این صفحه برسد، مجددا برنامه توسط IIS بارگذاری شده و اجرا می‌گردد. به این ترتیب وظایف تعريف شده، در طول یک روز بدون مشکل کار خواهند کرد.»

- یعنی صفحه ایندکس سایت را بهش بدیم مشکلی پیش نمی‌اد.
- اگر برنامه در حال انجام کار باشه بعد یک کاربری وارد سایت بشه کد در حال اجرا از کار میفته یا فقط تایمرو زنده نگه میداره

با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۰۳ ۱۲:۵۲

- در شرطی که نوشته شد «now.Second == 1» اضافی است.
- خیر؛ مشکلی نخواهد بود.
- این کارها پس زمینه هستند. اصلاً ارتباطی به UI برنامه ندارند و در ترد دیگری اجرا می‌شوند.

نویسنده: مهدی خیراندیش
تاریخ: ۱۳۹۳/۰۹/۰۹ ۱۶:۳

ممnon؛ اما من یه مشکلی دارم ساعت اجرا رو 1 بامداد تنظیم کردم ساعت سرور رو هم چک کردم تنظیمه اما فکر کنم اجرای ترد به اون موقع نمیرسه! اوایل ترد اجرا میشه اما همیشه موقعی که مت بیدارکننده! رو صدا میزنه خطاهای Server 503 و Internal Server Error 500 Unavailable throw پرت.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۰۹ ۱۸:۱۳

- از [ELMAH](#) برای ثبت خودکار ریز جزئیات خطاهای مدیریت نشده و بررسی آن‌ها استفاده کنید.
- همچنین در کتابخانه‌ی فوق به قسمت ScheduledTasksCoordinator.Current.OnUnexpectedException هم دقت داشته باشد؛ مطابق مثال ارائه شده.

در اکثر برنامه‌ها ما نیازمند این موضوع هستیم که بتوانیم اطلاعاتی را به کاربر نشان دهیم. در بعضی از موارد این اطلاعات بسیار زیاد هستند و نیاز است در این حالت از صفحه بندی اطلاعات یا Data Paging استفاده کنیم. در ASP.NET برای ارائه اطلاعات به کاربر معمولاً از کنترلهای ListView، GridView و امثال‌هم استفاده می‌شود. مشکل اساسی این کنترل‌ها این است که آنها اطلاعات را به صورت کامل از سرور دریافت کرده، سپس اقدام به نمایش صفحه بندی شده آن می‌نمایند که این موضوع باعث استفاده بی مورد از حافظه سرور شده و هزینه زیادی برای برنامه ما خواهد داشت.

صفحه بندی در سطح پایگاه داده بهترین روش برای استفاده بهینه از منابع است. برای رسیدن به این مقصود ما نیاز به یک کوئری خواهیم داشت که فقط همان صفحه مورد نیاز را به کنترل تحویل دهد.
با استفاده از متدهای زیر می‌توان به این مقصود دست یافت:

```

/// <summary>
/// صفحه بندی کوئری
/// </summary>
/// <param name="query"></param>
/// <param name="pageNum"></param>
/// <param name="pageSize"></param>
/// <param name="orderByProperty"></param>
/// <param name="isAscendingOrder"><c>true</c></param>
/// <param name="rowsCount"></param>
/// <returns></returns>
private static IQueryable<T> PagedResult<T, TResult>(IQueryable<T> query, int pageNum, int pageSize,
    Expression<Func<T, TResult>> orderByProperty, bool isAscendingOrder, out int rowsCount)
{
    if (pageSize <= 0) pageSize = 20;
    // مجموع ردیف‌های به دست آمده
    rowsCount = query.Count();
    // اگر شماره صفحه کوچکتر از ۰ بود صفحه اول نشان داده شود
    if (rowsCount <= pageSize || pageNum <= 0) pageNum = 1;
    // محاسبه ردیف‌های که نسبت به سایز صفحه باید از آنها گذشت
    int excludedRows = (pageNum - 1) * pageSize;
    query = isAscendingOrder ? query.OrderBy(orderByProperty) :
        query.OrderByDescending(orderByProperty);
    // ردشدن از ردیف‌های اضافی و دریافت ردیف‌های مورد نظر برای صفحه مربوطه
    return query.Skip(excludedRows).Take(pageSize);
}
  
```

نحوه استفاده :

فرض کنید که کوئری مورد نظر قرار است تا یکسری از مطالب را از جدول Articles نمایش دهد. برای دریافت ۲۰ ردیف اول جهت استفاده در صفحه اول، از کد زیر استفاده می‌کنیم:

```

var articles = (from article in Articles
    where article.Author == "Abc"
    select article);

int totalArticles;

var firstPageData = PagedResult(articles, 1, 20, article => article.PublishedDate, false, out
totalArticles);
  
```

یا به صورت ساده‌تر و قابل اجرا به صورت کلی‌تر:

```

var context = new ArticleEntityModel();
var query = context.ArticlesPagedResult(articles, <pageNumber>, 20, article => article.PublishedDate,
false, out totalArticles);
  
```

نظرات خوانندگان

نویسنده: محمد رضا ایزدی
تاریخ: ۱۳۹۳/۰۱/۲۶ ۱۱:۰

یه سوالی خط آخر چطوری اجرایی شده
شما تو کانتکس اون متد رو اوردین ؟

+

```
Helper.PagedResult(t, 1, 10, o=>true, false, out i);
```

یه اور لود هم اینطوری میشه براش نوشت اینطوری نیاز نیست حتما order در نظر گرفته شود

نویسنده: میثم ۹۹
تاریخ: ۱۳۹۳/۰۱/۲۶ ۱۱:۲۳

بسلام

مطلوبت بسیار مفیدی بود. فقط اگر بتوانی خود صفحه بندی را هم قرار دهی بسیار عالی میشود.
منظورم چند تا لینک که صفحه اول و آخر و صفحه جاری و تعداد دارد.
ممnon

نویسنده: محسن عباس آبادعربی
تاریخ: ۱۳۹۳/۰۱/۲۶ ۱۱:۴۷

ضمن تشکر از مطلب فوق
اگر شما در ASP.net استفاده میکنید میتوانی از کنترل ObjectDataSource استفاده کنی که چند مزیت دارد
1 سرعت بالایی دارد
2 امکان sql cache dependency را فعال میکنه یعنی فقط در هنگامی که شما اطلاعات رو در داخل گرید لود میکنید برای دفعه بعد
اگر اطلاعات در دیتابیس تغییری نکرده باشد دیگر به سمت دیتابیس مراجعه نمیکند و اطلاعات از cache خوانده میشود
2 امکان paging سمت سرور رو به شما میدهد .
3 برای پروژهای با دیتای بزرگ تست شده و جواب داده

نویسنده: میثم ۹۹
تاریخ: ۱۳۹۳/۰۱/۲۶ ۱۲:۹

در mvc هم یک هلپر برای اینکار ساخته شده است که خیلی خوب کار میکند.
ولی منظور من یک مازول ساده دست ساز بود که با توجه به سایت بتوان به هر شکل دلخواهی آنرا تغییر داد. در بعضی از
پروژهها واقعا یک همچین چیزی بدرد میخورد

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۲۶ ۱۲:۳۳

برای طراحی Pager سازگار با بوت استرپ این مطلب مفید است:
[A simple Bootstrap Pager Html Helper](#)

نویسنده: ایزدی
تاریخ: ۱۳۹۳/۰۱/۲۶ ۱۴:۱۸

ObjectDataSource میشه یه توضیحی در موردش بدین یا یه مقاله معرفی کنید

نویسنده: وحید نصیری
تاریخ: ۱۴:۲۷ ۱۳۹۳/۰ ۱/۲۶

ObjectContainerDataSource Control

نویسنده: محمد رضا صفری
تاریخ: ۲۰:۳۲ ۱۳۹۳/۰ ۱/۲۶

کسانی که Table بی دردر و Ajax ای میخوان از این پلاگین استفاده کنند :
[/http://www.jtable.org](http://www.jtable.org)

با MVC هم کاملا سازگار هست و نمونه هم دارد .

[آموزش کامل : http://www.codeproject.com/Articles/277576/AJAX-based-CRUD-tables-using-ASP-NET-MVC-and-jTa](http://www.codeproject.com/Articles/277576/AJAX-based-CRUD-tables-using-ASP-NET-MVC-and-jTa) . [هست Datatable](#)

نویسنده: محسن عباس آبادعربی
تاریخ: ۱۲:۲۲ ۱۳۹۳/۰ ۱/۲۷

کنترل [ObjectContainerDataSource Control](#) مربوط به فزیم ورک WCSF می باشد می توانید از سایت مايكروسافت دانلود نمایید.

نویسنده: ایزدی
تاریخ: ۱۶:۵۶ ۱۳۹۳/۰ ۶/۰۸

سلام

من یک برنامه تولید لایه business نوشتم از کد شما هم استفاده کردم با اجازتون یه تعییر کوچیک دادم تو ش خواستم اینجا هم بذارم که اگر کسی خواست استفاده کنه

```
private static IQueryables<T> PagedResult<T, TResult>(IQueryables<T> query, int pageNum, int pageSize,
                                                       Expression<Func<T, TResult>>
orderByProperty,
                                                       bool isAscendingOrder, out int rowsCount,
                                                       Expression<Func<T, bool>> whereClause =
null)
{
    if (pageSize <= 0) pageSize = 20;
    مجموع ردیفهای به دست آمد//  

    rowsCount = query.Count();
    اگر شماره صفحه کوچکتر از ۰ بود صفحه اول نشان داده شود //  

    if (rowsCount <= pageSize || pageNum <= 0) pageNum = 1;
    محاسبه ردیف هایی که نسبت به سایز صفحه باید از آنها گذشت //  

    int excludedRows = (pageNum - 1) * pageSize;
    query = isAscendingOrder ? query.OrderBy(orderByProperty) :
query.OrderByDescending(orderByProperty);
    جستجو را در صورت لزوم انجام می دهد//  

    query = whereClause == null ? query : query.Where(whereClause);
    ردشدن از ردیفهای اضافی و دریافت ردیفهای مورد نظر برای صفحه مربوطه //  

    return query.Skip(excludedRows).Take(pageSize);
}
```

و برای فراخوانی هم اینطور استفاده کردم

```
public static List<t_Prodcts> GetPaging(int currentPage, int pageSize, out int count,
                                         Expression<Func<t_Prodcts, bool>> search = null)
{
    using (var db = new asusIranDBConnection())
    {
        return PagedResult(db.t_Prodcts, currentPage, pageSize, o => true, false, out count,
search).ToList();
    }
}
```

نویسنده: ابراهیم
تاریخ: ۱۶:۲۹ ۱۳۹۴/۰ ۱/۰۷

سلام

وقتی کدها Order می‌شود تمامی اطلاعات از بانک فرآخونی میشه ؟

نویسنده: محمد رضا صفری
تاریخ: ۱۶:۴۲ ۱۳۹۴/۰ ۱/۰۷

خیر ، زمانی کهToList می‌کنید تازه دستور اجرا میشه و نتیجه برگشت داده میشه .

شاید شما هم قصد داشته باشید تا از برخی درخواست‌ها به وب سایت یا اپلیکیشن خود ممانعت عمل بیاورید. نظیر درخواست‌های SQL Injection یا برخی Query String های خاص یا برخی درخواست‌های مزاحم.

یکی از مزاحمت‌هایی که گریبانگر وب سایت‌هاست، Bot‌های متفاوتی است که برای کپی اطلاعات، درج کامنت به صورت خودکار و مواردی از این دست، به آنها مراجعه می‌کنند. شاید در نگاه اول بد نباشد که این Bot‌ها به سراغ وب سایت ما بینند و باعث افزایش تعداد ویزیت سایتمان شوند؛ ولی ضررها ناشی از کپی و سرقـت مطالب سایت، آنهم با سرعت بالا، بیشتر از منافع ناشی از بالا رفتن رنک سایت است. به طور مثال همین سایت [NET Tips](#). دارای تعداد زیادی مقالات مفید است که افراد متعددی در نگارش و تهیه آن‌ها زحمت کشیده‌اند، یا وب سایتی برای جلب اعتماد مشتریان جهت درج اطلاعاتشان و یا آگهی‌هایشان زحمت زیادی کشیده است، Bot‌های آماده‌ی زیادی وجود دارد که با چند دقیقه صرف وقت جهت تنظیم شدن آماده می‌شوند تا مطالب را طبق ساختار تعیین شده، مورد به مورد کپی کنند.

برای خلاصی از این موارد روش‌های متعددی وجود دارد که از جمله آنها می‌توان به تنظیمات فایل `htaccess` در وب سرورهای Apache و یا `web.config` در IIS اشاره کرد. در این مقاله این امکان را با IIS مرور می‌کنیم و برای فعال سازی آن کافی است در:

IIS 7.5 و بالاتر، همراه با انتخاب Request Filtering در مراحل نصب IIS

IIS 7.0 پس از نصب بسته آپدیت [Microsoft Knowledge Base Article 957508](#).

IIS 6.0 با نصب URLScan 3.0

در بخش `<system.webServer>` و سپس `<security>`، تگ `requestFiltering` را استفاده کنیم، در این تگ دستورالعمل‌های ویژه‌ی پالایشگر درخواست‌ها را مینویسیم (`filteringRules`) هر دستورالعمل پالایش دارای خصیصه‌های (Attributes) زیر است:

`denyUnescapedPercent`

مقدار Boolean و انتخابی

اگر برابر با `true` تنظیم گردد، درخواست‌هایی که دارای کاراکتر "درصد" (%) هستند و به وسیله `escape character` ها پوشش داده نشده باشند، رد می‌شوند. (جهت جلوگیری از حملات XSS و...) مقدار پیش فرض `true` است.

`name`

عنوان دستورالعمل.

مقدار پیش فرض نداشته و درج کردن آن اجباری است.

`scanAllRaw`

مقدار Boolean و انتخابی

اگر برابر با `true` تنظیم گردد، پالایشگر درخواست‌ها موظف است تا با بررسی متن `header` های درخواست، در صورت یافتن یکی از واژه‌هایی که در خصیصه `denyStrings` ذکر کرده‌اید، درخواست را رد کند. مقدار پیش فرض `false` است.

`scanQueryString`

مقدار Boolean و انتخابی

اگر برابر با true تنظیم گردد، پالایشگر درخواست‌ها موظف است تا Query string را بررسی کند تا در صورتی که یکی از واژه‌های درج شده در خصیصه denyStrings را بباید، درخواست را رد کند.

اگر خصیصه‌ی unescapeQueryString از تگ <requestFiltering> دوبار بررسی می‌شود: یکبار متن query string برای یافتن عبارات ممنوعه و بار دیگر برای یافتن کarakترهای بدون پوشش scaped. مقدار پیشفرض false است.

scanUrl

مقدار Boolean و انتخابی

اگر برابر با true تنظیم گردد، پالایشگر درخواست‌ها URL را برای یافتن واژه‌های ممنوعه‌ی ذکر شده در خصیصه denyStrings بررسی می‌نماید.

مقدار پیش فرض false است.

چند مثال:

مثال 1: در این مثال عنوان User-Agent هایی را که در موارد متعدد برای وب سایت‌هایی که روی آنها کار میکردم مزاحمت ایجاد میکردند را پالایش میکنیم. (لیست این Bot‌ها آپدیت میشود)

```
<requestFiltering>
  <filteringRules>
    <filteringRule name="BlockSearchEngines" scanUrl="false" scanQueryString="false">
      <scanHeaders>
        <clear />
        <add requestHeader="User-Agent" />
      </scanHeaders>
      <appliesTo>
        <clear />
      </appliesTo>
      <denyStrings>
        <clear />
        <add string="Python UrlLib" />
        <add string="WGet" />
        <add string="Apache HttpClient" />
        <add string="Unknown Bot" />
        <add string="Yandex Spider" />
        <add string="libwww-perl" />
        <add string="Nutch" />
        <add string="DotBot" />
        <add string="CCBot" />
        <add string="Majestic 12 Bot" />
        <add string="Java" />
        <add string="Link Checker" />
        <add string="Baiduspider" />
        <add string="Exabot" />
        <add string="PHP" />
      </denyStrings>
    </filteringRule>
  </filteringRules>
</requestFiltering>
```

مثال 2: ممانعت از SQL Injection

```
<requestFiltering>
  <filteringRules>
    <filteringRule name="SQLInjection" scanUrl="false" scanQueryString="true">
      <appliesTo>
        <clear />
        <add fileExtension=".asp" />
        <add fileExtension=".aspx" />
        <add fileExtension=".php" />
      </appliesTo>
      <denyStrings>
        <clear />
        <add string="--" />
        <add string=";" />
      </denyStrings>
    </filteringRule>
  </filteringRules>
</requestFiltering>
```

```
<add string="/*" />
<add string="@" />
<add string="char" />
<add string="alter" />
<add string="begin" />
<add string="cast" />
<add string="create" />
<add string="cursor" />
<add string="declare" />
<add string="delete" />
<add string="drop" />
<add string="end" />
<add string="exec" />
<add string="fetch" />
<add string="insert" />
<add string="kill" />
<add string="open" />
<add string="select" />
<add string="sys" />
<add string="table" />
<add string="update" />
</denyStrings>
<scanHeaders>
  <clear />
</scanHeaders>
</filteringRule>
</filteringRules>
</requestFiltering>
```

مثال 3: ممانعت از درخواست انواع خاصی از فایل ها

```
<requestFiltering>
  <filteringRules>
    <filteringRule name="Block Image Leeching" scanUrl="false" scanQueryString="false"
scanAllRaw="false">
      <scanHeaders>
        <add requestHeader="User-agent" />
      </scanHeaders>
      <appliesTo>
        <add fileExtension=".zip" />
        <add fileExtension=".rar" />
        <add fileExtension=".exe" />
      </appliesTo>
      <denyStrings>
        <add string="leech-bot" />
      </denyStrings>
    </filteringRule>
  </filteringRules>
</requestFiltering>
```

نظرات خوانندگان

نویسنده: امین مصباحی
تاریخ: ۱۳۹۳/۰۲/۰۱ ۶:۳

تکمیلی ۱: هم از جمله‌ی Bot‌های مزاحم است، لذا:

</ "add string="AhrefsBot>

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۰۱ ۹:۳۵

لیست User-Agent‌هایی است که من در این سایت بستم تا امروز (از لاغ‌های خطای برنامه استخراج شدند):

[bots.txt](#)

در تاریخ 20 مارچ 2014 تیم ASP.NET نسخه نهایی 2.0 Identity را منتشر کردند. نسخه جدید برخی از ویژگی‌های درخواست شده پیشین را عرضه می‌کند و در کل قابلیت‌های احراز هویت و تعیین سطح دسترسی ارزش‌های ای را پشتیبانی می‌کند. این فریم ورک در تمام اپلیکیشن‌های ASP.NET می‌تواند بکار گرفته شود.

فریم ورک Identity در سال 2013 معرفی شد، که دنباله سیستم ASP.NET Membership بود. سیستم قبلی گرچه طی سالیان استفاده می‌شد اما مشکلات زیادی هم بهمراه داشت. بعلاوه با توسعه دنیای وب و نرم افزار، قابلیت‌های مدرنی مورد نیاز بودند که باید پشتیبانی می‌شدند. فریم ورک Identity در ابتدا سیستم ساده و کارآمدی برای مدیریت کاربران بوجود آورد و مشکلات پیشین را تا حد زیادی برطرف نمود. عنوان مثال فریم ورک جدید مبتنی بر EF Code-first است، که سفارشی کردن سیستم عضویت را بسیار آسان می‌کند و به شما کنترل کامل می‌دهد. یا مثلاً احراز هویت مبتنی بر پروتوكول OAuth پشتیبانی می‌شود که به شما اجازه استفاده از فراهم کنندگان خارجی مانند گوگل، فیسبوک و غیره را می‌دهد.

نسخه جدید این فریم ورک ویژگی‌های زیر را معرفی می‌کند (بعلاوه مواردی دیگر):

مدل حساب‌های کاربری توسعه داده شده. مثلاً آدرس ایمیل و اطلاعات تماس را هم در بر می‌گیرد احراز هویت دو مرحله‌ای (Two-Factor Authentication) توسط اطلاع رسانی ایمیلی یا پیامکی. مشابه سیستمی که گوگل، مایکروسافت و دیگران استفاده می‌کنند تایید حساب‌های کاربری توسط ایمیل (Account Confirmation) مدیریت کاربران و نقش‌ها (Administration of Users & Roles) قفل کردن حساب‌های کاربری در پاسخ به Invalid log-in attempts

تامین کننده شناسه امنیتی (Security Token Provider) برای بازتولید شناسه‌ها در پاسخ به تغییرات تنظیمات امنیتی (مثلاً هنگام تغییر کلمه عبور)

بهبود پشتیبانی از Social log-ins
یکپارچه سازی ساده با Claims-based Authorization

Identity 2.0 تغییرات چشم‌گیری نسبت به نسخه قبلی به وجود آورده است. به نسبت ویژگی‌های جدید، پیچیدگی‌های نیز معرفی شده‌اند. اگر به تازگی (مانند خودم) با نسخه 1 این فریم ورک آشنا شده و کار کرده اید، آماده شوید! گرچه لازم نیست از صفر شروع کنید، اما چیزهای بسیاری برای آموختن وجود دارد.

در این مقاله نگاهی اجمالی به نسخه‌ی جدید این فریم ورک خواهیم داشت. کامپوننت‌های جدید و اصلی را خواهیم شناخت و خواهیم دید هر کدام چگونه در این فریم ورک کار می‌کنند. بررسی عمیق و جزئی این فریم ورک از حوصله این مقاله خارج است، بنابراین به این مقاله تنها عنوان یک نقطه شروع برای آشنایی با این فریم ورک نگاه کنید.

اگر به دنبال اطلاعات بیشتر و بررسی‌های عمیق‌تر هستید، لینک‌هایی در انتهای این مقاله نگاشت شده‌اند. همچنین طی هفته‌های آینده چند مقاله تخصصی‌تر خواهیم نوشت تا از دید پیاده سازی بیشتر با این فریم ورک آشنا شوید.

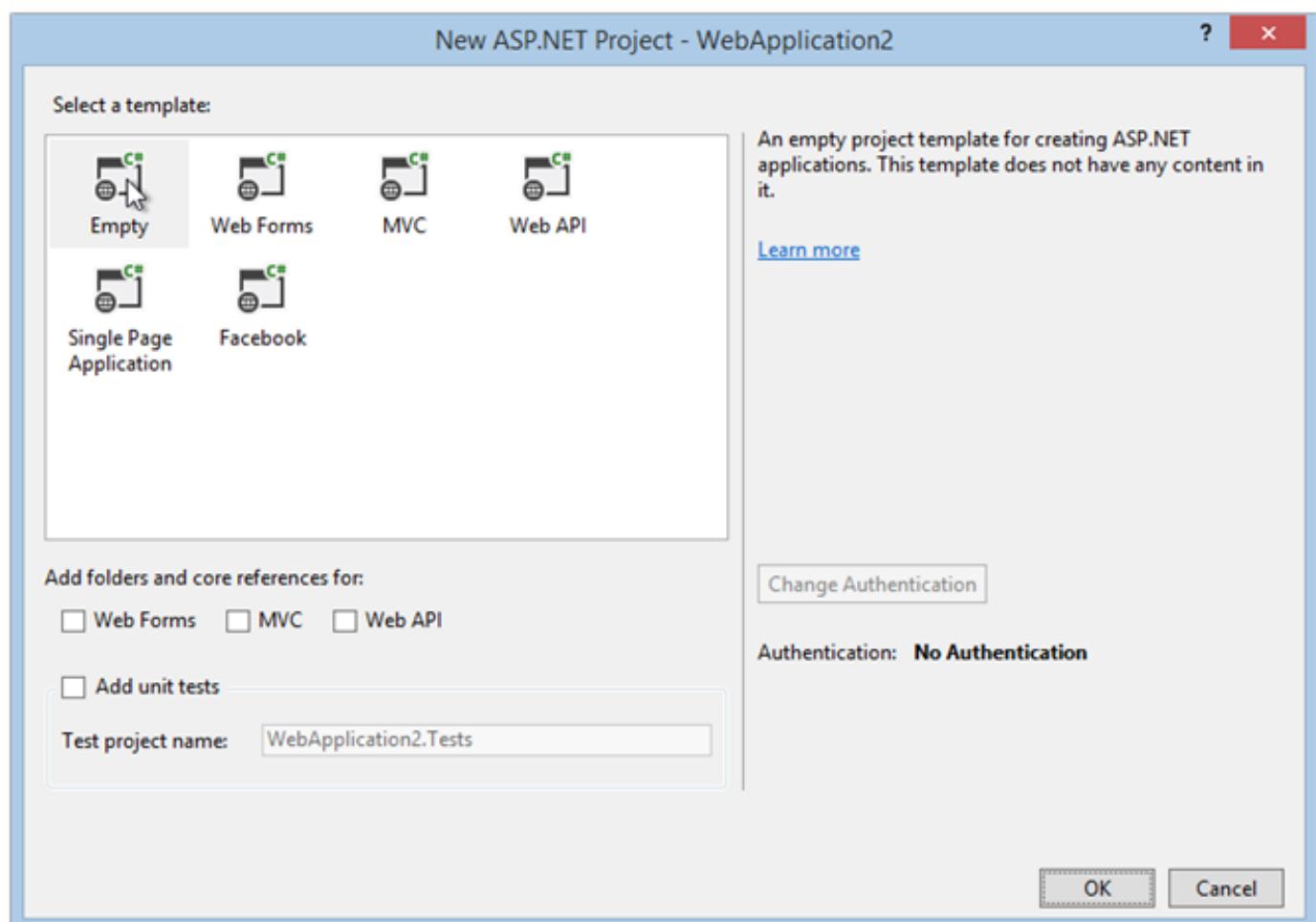
در این مقاله با مقدار قابل توجهی کد مواجه خواهید شد. لازم نیست تمام جزئیات آنها را بررسی کنید، تنها با ساختار کلی این فریم ورک آشنا شوید. کامپوننت‌ها را بشناسید و بدانید که هر کدام در کجا قرار گرفته‌اند، چطور کار می‌کنند و اجزای کلی سیستم چگونه پیکربندی می‌شوند. گرچه، اگر به برنامه نویسی دات نت (#ASP.NET) تسلط دارید و با نسخه قبلی Identity هم کار کرده‌اید، درک کدهای جدید کار ساده‌ای خواهد بود.

با نسخه قبلی سازگار نیست

اپلیکیشن هایی که با نسخه 1.0 این فریم ورک ساخته شده اند نمی توانند بسادگی به نسخه جدید مهاجرت کنند. قابلیت هایی جدیدی که پیاده سازی شده اند تغییرات چشمگیری در معماری این فریم ورک بوجود آورده اند، همچنین API مورد استفاده در اپلیکیشن ها نیز دستخوش تغییراتی شده است. مهاجرت از نسخه 1.0 به 2.0 نیاز به نوشتگری کدهای جدید و اعمال تغییرات متعددی دارد که از حوصله این مقاله خارج است. فعلاً همین قدر بدانید که این مهاجرت نمی تواند بسادگی در قالب *Plug-in and play* صورت پذیرد!

شروع به کار : پروژه مثال ها را از NuGet دریافت کنید

در حال حاضر (هنگام نوشتگر این مقاله) قالب پروژه استانداردی برای اپلیکیشن های ASP.NET MVC که از 2.0 Identity استفاده کنند وجود ندارد. برای اینکه بتوانید از نسخه جدید این فریم ورک استفاده کنید، باید پروژه مثال را توسط NuGet دریافت کنید. ابتدا پروژه جدیدی از نوع ASP.NET Web Application با سازنده Empty و قالب Web Application را در دیالوگ تنظیمات انتخاب کنید.



کنسول Package Manager را باز کنید و با اجرای فرمان زیر پروژه مثال ها را دانلود کنید.

```
PM> Install-Package Microsoft.AspNet.Identity.Samples -Pre
```

پس از آنکه NuGet کار خود را به اتمام رساند، ساختار پروژه ای مشابه پروژه های استاندارد MVC مشاهده خواهد کرد. پروژه شما شامل قسمت های Models، Views، Controllers و کامپوننت های دیگری برای شروع به کار است. گرچه در نگاه اول ساختار پروژه بسیار شبیه به پروژه های استاندارد ASP.NET MVC به نظر می آید، اما با نگاهی دقیق تر خواهد دید که تغییرات جدیدی ایجاد شده اند و پیچیدگی هایی نیز معرفی شده اند.

پیکربندی Identity : دیگر به سادگی نسخه قبلی نیست

به نظر من یکی از مهم ترین نقاط قوت فریم ورک Identity یکی از مهم ترین نقاط ضعفش نیز بود. سادگی نسخه 1.0 این فریم ورک کار کردن با آن را بسیار آسان می کرد و به سادگی می توانستید ساختار کلی و روند کار کردن کامپوننت های آن را درک کنید. اما همین سادگی به معنای محدود بودن امکانات آن نیز بود. بعنوان مثال می توان به تایید حساب های کاربری یا پشتیبانی از احراز هویت های دو مرحله ای اشاره کرد.

برای شروع نگاهی اجمالی به پیکربندی این فریم ورک و اجرای اولیه اپلیکیشن خواهیم داشت. سپس تغییرات را با نسخه 1.0 مقایسه می کنیم.

در هر دو نسخه، فایلی بنام Startup.cs در مسیر ریشه پروژه خواهد یافت. در این فایل کلاس واحدی بنام Startup تعریف شده است که متدها ConfigureAuth() را فراخوانی می کند. چیزی که در این فایل مشاهده نمی کنیم، خود متدهای ConfigurationAuth است. این بدين دلیل است که مابقی کد کلاس Startup در یک کلاس پاره ای (Partial) تعریف شده که در پوشش App_Start قرار دارد. نام فایل مورد نظر Startup.Auth.cs است که اگر آن را باز کنید تعاریف یک کلاس پاره ای بهمراه متدهای ConfigurationAuth را خواهد یافت. در یک پروژه که از نسخه 1.0 استفاده می کند، کد متدهای ConfigurationAuth مطابق لیست زیر است.

```
public partial class Startup
{
    public void ConfigurationAuth(IApplicationBuilder app)
    {
        // Enable the application to use a cookie to
        // store information for the signed in user
        app.UseCookieAuthentication(new CookieAuthenticationOptions
        {
            AuthenticationType = DefaultAuthenticationTypes.ApplicationCookie,
            LoginPath = new PathString("/Account/Login")
        });

        // Use a cookie to temporarily store information about a
        // user logging in with a third party login provider
        app.UseExternalSignInCookie(DefaultAuthenticationTypes.ExternalCookie);

        // Uncomment the following lines to enable logging
        // in with third party login providers
        // app.UseMicrosoftAccountAuthentication(
        //     clientId: "",
        //     clientSecret: "");

        // app.UseTwitterAuthentication(
        //     consumerKey: "",
        //     consumerSecret: "");

        // app.UseFacebookAuthentication(
        //     appId: "",
        //     appSecret: "");

        // app.UseGoogleAuthentication();
    }
}
```

در قطعه کد بالا پیکربندی لازم برای کوکی ها را مشاهده می کنید. همچنین کدهایی بصورت توضیحات وجود دارد که به منظور استفاده از تامین کنندگان خارجی مانند گوگل، فیسبوک، توییتر و غیره استفاده می شوند. حال اگر به کد این متدهای نسخه 2.0 دقت کنید خواهید دید که کد بیشتری نوشته شده است.

```
public partial class Startup {
    public void ConfigurationAuth(IApplicationBuilder app) {
        // Configure the db context, user manager and role
        // manager to use a single instance per request
    }
}
```

```

    app.CreatePerOwinContext(ApplicationUserManager.Create);
    app.CreatePerOwinContext<ApplicationRoleManager>(ApplicationRoleManager.Create);

    // Enable the application to use a cookie to store information for the
    // signed in user and to use a cookie to temporarily store information
    // about a user logging in with a third party login provider
    // Configure the sign in cookie
    app.UseCookieAuthentication(new CookieAuthenticationOptions {
        AuthenticationType = DefaultAuthenticationTypes.ApplicationCookie,
        LoginPath = new PathString("/Account/Login"),
        Provider = new CookieAuthenticationProvider {
            // Enables the application to validate the security stamp when the user
            // logs in. This is a security feature which is used when you
            // change a password or add an external login to your account.
            OnValidateIdentity = SecurityStampValidator
                .OnValidateIdentity<ApplicationUserManager, ApplicationUser>(
                    validateInterval: TimeSpan.FromMinutes(30),
                    regenerateIdentity: (manager, user)
                        => user.GenerateUserIdentityAsync(manager))
        }
    });
}

app.UseExternalSignInCookie(DefaultAuthenticationTypes.ExternalCookie);

// Enables the application to temporarily store user information when
// they are verifying the second factor in the two-factor authentication process.
app.UseTwoFactorSignInCookie(
    DefaultAuthenticationTypes.TwoFactorCookie,
    TimeSpan.FromMinutes(5));

// Enables the application to remember the second login verification factor such
// as phone or email. Once you check this option, your second step of
// verification during the login process will be remembered on the device where
// you logged in from. This is similar to the RememberMe option when you log in.
app.UseTwoFactorRememberBrowserCookie(
    DefaultAuthenticationTypes.TwoFactorRememberBrowserCookie);

// Uncomment the following lines to enable logging in
// with third party login providers
// app.UseMicrosoftAccountAuthentication(
//     clientId: "",
//     clientSecret: "");

// app.UseTwitterAuthentication(
//     consumerKey: "",
//     consumerSecret: "");

// app.UseFacebookAuthentication(
//     appId: "",
//     appSecret: "");

// app.UseGoogleAuthentication();
}
}

```

اول از همه به چند فراخوانی متد `app.CreatePerOwinContext` بر می‌خوریم. با این فراخوانی‌ها `Callback`‌هایی را登録 می‌کنیم که آبجکت‌های مورد نیاز را بر اساس نوع تعریف شده توسط `type arguments` و هله سازی می‌کنند. این وله‌ها سپس توسط فراخوانی متد `(context.Get<T>()` قابل دسترسی خواهند بود. این به ما می‌گوید که حالا `Owin` بخشی از اپلیکیشن ما است و `Identity` را برای ارائه قابلیت هایش استفاده می‌کند.

مورد بعدی ای که جلب توجه می‌کند فراخوانی‌های دیگری برای پیکربندی احراز هویت دو مرحله‌ای است. همچنین پیکربندی‌های جدیدی برای کوکی‌ها تعریف شده است که در نسخه قبلي وجود نداشتند.

تا اینجا پیکربندی‌های اساسی برای اپلیکیشن شما انجام شده است و می‌توانید از اپلیکیشن خود استفاده کنید. بکارگیری فراهم کنندگان خارجی در حال حاضر غیرفعال است و بررسی آنها نیز از حوصله این مقاله خارج است. این کلاس پیکربندی‌های اساسی IdentityConfig.cs را انجام می‌دهد. کامپونت‌های پیکربندی و کدهای کمکی دیگری نیز وجود دارند که در کلاس `IdentityConfig.cs` تعریف شده اند.

پیش از آنکه فایل `IdentityConfig.cs` را بررسی کنیم، بهتر است نگاهی به کلاس `ApplicationUser` بیاندازیم که در پوشه `Models` قرار گرفته است.

کلاس جدید `ApplicationUser` در Identity 2.0

اگر با نسخه 1.0 این فریم ورک اپلیکیشنی ساخته باشید، ممکن است متوجه شده باشید که کلاس پایه `IdentityUser` محدود و شاید ناکافی باشد. در نسخه قبلی، این فریم ورک پیاده سازی `IdentityUser` را تا حد امکان ساده نگاه داشته بود تا اطلاعات پروفایل کاربران را معرفی کند.

```
public class IdentityUser : IUser
{
    public IdentityUser();
    public IdentityUser(string userName);

    public virtual string Id { get; set; }
    public virtual string UserName { get; set; }

    public virtual ICollection<IdentityUserRole> Roles { get; }
    public virtual ICollection<IdentityUserClaim> Claims { get; }
    public virtual ICollection<IdentityUserLogin> Logins { get; }

    public virtual string PasswordHash { get; set; }
    public virtual string SecurityStamp { get; set; }
}
```

بین خواص تعریف شده در این کلاس، تنها `Id`, `UserName` و `Roles` برای ما حائز اهمیت هستند (از دید برنامه نویسی). مابقی خواص عمده توسط منطق امنیتی این فریم ورک استفاده می‌شوند و کمک شایانی در مدیریت اطلاعات کاربران به ما نمی‌کنند.

اگر از نسخه 1.0 Identity استفاده کرده باشید و مطالعاتی هم در این زمینه داشته باشید، می‌دانید که توسعه کلاس کاربران بسیار ساده است. مثلاً برای افزودن فیلد آدرس ایمیل و اطلاعات دیگر کافی بود کلاس `ApplicationUser` را ویرایش کنیم و از آنجا که این فریم ورک مبتنی بر EF Code-first است بروز رسانی دیتابیس و مابقی اپلیکیشن کار چندان مشکلی نخواهد بود.

با ظهور نسخه 2.0 Identity نیاز به برخی از این سفارشی سازی‌ها از بین رفته است. گرچه هنوز هم می‌توانید بسادگی مانند گذشته کلاس `ApplicationUser` را توسعه و گسترش دهید، تیم ASP.NET تعییراتی بوجود آورده اند تا نیازهای رایج توسعه دهنده‌گان را پاسخگو باشد.

اگر به کد کلاس‌های مربوطه دقت کنید خواهید دید که کلاس `ApplicationUser` همچنان از کلاس پایه `IdentityUser` ارث بری می‌کند، اما این کلاس پایه پیچیده‌تر شده است. کلاس `ApplicationUser` در پوشه `Models` و در فایلی بنام `IdentityModels.cs` تعریف شده است. همانطور که می‌بینید تعاریف خود این کلاس بسیار ساده است.

```
public class ApplicationUser : IdentityUser {
    public async Task<ClaimsIdentity> GenerateUserIdentityAsync(
        UserManager< ApplicationUser > manager) {
        // Note the authenticationType must match the one
        // defined in CookieAuthenticationOptions.AuthenticationType
        var userIdentity =
            await manager.CreateIdentityAsync(this,
                DefaultAuthenticationTypes.ApplicationCookie);

        // Add custom user claims here
        return userIdentity;
    }
}
```

حال اگر تعاریف کلاس `ApplicationUser` را بازیابی کنید (با استفاده از قابلیت Go To Definition در VS) خواهید دید که این کلاس خود از کلاس پایه دیگری ارث بری می‌کند. اگر به این پیاده سازی دقت کنید کاملاً واضح است که ساختار این کلاس به کلی نسبت به نسخه قبلی تغییر کرده است.

```
public class IdentityUser< TKey, TLogin, TRole, TClaim > : IUser< TKey >
    where TLogin : Microsoft.AspNet.Identity.EntityFramework.IdentityUserLogin< TKey >
```

```

where TRole : Microsoft.AspNet.Identity.EntityFramework.IdentityUserRole< TKey >
where TClaim : Microsoft.AspNet.Identity.EntityFramework.IdentityUserClaim< TKey >
{
    public IdentityUser();

    // Used to record failures for the purposes of lockout
    public virtual int AccessFailedCount { get; set; }
    // Navigation property for user claims
    public virtual ICollection< TClaim > Claims { get; }
    // Email
    public virtual string Email { get; set; }
    // True if the email is confirmed, default is false
    public virtual bool EmailConfirmed { get; set; }
    // User ID (Primary Key)
    public virtual TKey Id { get; set; }
    // Is lockout enabled for this user
    public virtual bool LockoutEnabled { get; set; }
    // DateTime in UTC when lockout ends, any
    // time in the past is considered not locked out.
    public virtual DateTime? LockoutEndDateUtc { get; set; }

    // Navigation property for user logins
    public virtual ICollection< TLogin > Logins { get; }
    // The salted/hashed form of the user password
    public virtual string PasswordHash { get; set; }
    // PhoneNumber for the user
    public virtual string PhoneNumber { get; set; }
    // True if the phone number is confirmed, default is false
    public virtual bool PhoneNumberConfirmed { get; set; }
    // Navigation property for user roles
    public virtual ICollection< TRole > Roles { get; }

    // A random value that should change whenever a users
    // credentials have changed (password changed, login removed)
    public virtual string SecurityStamp { get; set; }
    // Is two factor enabled for the user
    public virtual bool TwoFactorEnabled { get; set; }
    // User name
    public virtual string UserName { get; set; }
}

```

اول از همه آنکه برخی از خواص تعریف شده هنوز توسط منطق امنیتی فریم ورک استفاده می‌شوند و از دید برنامه نویسی مربوط به مدیریت اطلاعات کاربران نیستند. اما به هر حال فیلدهای Email و PhoneNumber نیاز به ویرایش تعریف پیش فرض موجودیت کاربران را از بین می‌برد.

اما از همه چیز مهم‌تر امضا (Signature) خود کلاس است. این آرگومانهای جنریک چه هستند؟ به امضای این کلاس دقت کنید.

```

public class IdentityUser< TKey, TLogin, TRole, TClaim > : IUser< TKey >
    where TLogin : Microsoft.AspNet.Identity.EntityFramework.IdentityUserLogin< TKey >
    where TRole : Microsoft.AspNet.Identity.EntityFramework.IdentityUserRole< TKey >
    where TClaim : Microsoft.AspNet.Identity.EntityFramework.IdentityUserClaim< TKey >

```

نسخه جدید IdentityUser انواع آرگومانهای جنریک را پیاده سازی می‌کند که انعطاف پذیری بسیار بیشتری به ما می‌دهند. بعنوان مثال بیاد بیاورید که نوع داده فیلد Id در نسخه 1.0 رشته (string) بود. اما در نسخه جدید استفاده از آرگومانهای جنریک (در اینجا T) به ما اجازه می‌دهد که نوع این فیلد را مشخص کنیم. همانطور که مشاهده می‌کنید خاصیت Id در این کلاس نوع داده را باز می‌گرداند. TKey

```
public virtual TKey Id { get; set; }
```

یک مورد حائز اهمیت دیگر خاصیت Roles است که بصورت زیر تعریف شده.

```
public virtual ICollection< TRole > Roles { get; }
```

همانطور که می‌بینید نوع TRole بصورت جنریک تعریف شده و توسط تعاریف کلاس IdentityUser مشخص می‌شود. اگر به

محدودیت‌های پیاده سازی این خاصیت دقت کنید می‌بینید که نوع این فیلد به `IdentityUserRole< TKey >` محدود (constraint) شده است، که خیلی هم نسبت به نسخه 1.0 تغییری نکرده. چیزی که تفاوت چشمگیری کرده و باعث `breaking changes` می‌شود تعریف خود `IdentityUserRole` است.

در نسخه 1.0 کلاس `IdentityUserRole` بصورت زیر تعریف شده بود.

```
public class IdentityUserRole
{
    public IdentityUserRole();
    public virtual IdentityRole Role { get; set; }
    public virtual string RoleId { get; set; }
    public virtual IdentityUser User { get; set; }
    public virtual string UserId { get; set; }
}
```

این کلاس را با پیاده سازی نسخه 2.0 مقایسه کنید.

```
public class IdentityUserRole< TKey >
{
    public IdentityUserRole();
    public virtual TKey RoleId { get; set; }
    public virtual TKey UserId { get; set; }
}
```

پیاده سازی پیشین ارجاعاتی به آبجکت‌هایی از نوع `IdentityUser` و `IdentityRole` داشت. پیاده سازی نسخه 2.0 تنها شناسه‌ها را ذخیره می‌کند. اگر در اپلیکیشنی که از نسخه 1.0 استفاده می‌کند سفارشی سازی هایی انجام داده باشد (مثلاً تعریف کلاس `Role` را توسعه داده باشد، یا سیستمی مانند `Group-based Roles` را پیاده سازی کرده باشد) این تغییرات سیستم قبلی شما را خواهد شکست.

بررسی پیاده سازی جدید `IdentityUser` از حوصله این مقاله خارج است. فعلاً همین قدر بدانید که گرچه تعاریف پایه کلاس کاربران پیچیده‌تر شده است، اما انعطاف پذیری بسیار خوبی بدست آمده که شایان اهمیت فراوانی است.

از آنجا که کلاس `ApplicationUser` از `IdentityUser` ارث بری می‌کند، تمام خواص و تعاریف این کلاس پایه در `ApplicationUser` قابل دسترسی هستند.

کامپوننت‌های پیکربندی 2.0 و کدهای کمکی

گرچه متدهای `ConfigAuth()` در کلاس `Startup` محلی است که پیکربندی `Identity` در زمان اجرا صورت می‌پذیرد، اما در واقع کامپوننت‌های موجود در فایل `IdentityConfig.cs` هستند که اکثر قابلیت‌های 2.0 را پیکربندی کرده و نحوه رفتار آنها در اپلیکیشن ما را کنترل می‌کنند.

اگر محتوای فایل `IdentityConfig.cs` را بررسی کنید خواهید دید که کلاس‌های متعددی در این فایل تعریف شده اند. می‌توان تک تک این کلاس‌ها را به فایل‌های مجزایی منتقل کرد، اما برای مثال جاری کدها را بهمین صورت رها کرده و نگاهی اجمالی به آنها خواهیم داشت. بهرحال در حال حاضر تمام این کلاس‌ها در فضای نام `ApplicationName.Models` قرار دارند.

Application Role Manager و Application User Manager

اولین چیزی که در این فایل به آنها بر می‌خوریم دو کلاس `ApplicationRoleManager` و `ApplicationUserManager` هستند. آماده باشید، مقدار زیادی کد با انواع داده جنریک در پیش روست!

```
public class ApplicationUserManager : UserManager< ApplicationUser >
{
    public ApplicationUserManager(IUserStore< ApplicationUser > store)
        : base(store)
    {
    }
```

```

public static ApplicationUserManager Create(
    IdentityFactoryOptions<ApplicationUserManager> options,
    IWinContext context)
{
    var manager = new ApplicationUserManager(
        new UserStore<ApplicationUser>(
            context.Get<ApplicationDbContext>()));

    // Configure validation logic for usernames
    manager.UserValidator =
        new UserValidator<ApplicationUser>(manager)
    {
        AllowOnlyAlphanumericUserNames = false,
        RequireUniqueEmail = true
    };

    // Configure validation logic for passwords
    manager.PasswordValidator = new PasswordValidator
    {
        RequiredLength = 6,
        RequireNonLetterOrDigit = true,
        RequireDigit = true,
        RequireLowercase = true,
        RequireUppercase = true,
    };

    // Configure user lockout defaults
    manager.UserLockoutEnabledByDefault = true;
    manager.DefaultAccountLockoutTimeSpan = TimeSpan.FromMinutes(5);
    manager.MaxFailedAccessAttemptsBeforeLockout = 5;

    // Register two factor authentication providers. This application uses
    // Phone and Emails as a step of receiving a code for verifying
    // the user. You can write your own provider and plug in here.
    manager.RegisterTwoFactorProvider("PhoneCode",
        new PhoneNumberTokenProvider<ApplicationUser>
    {
        MessageFormat = "Your security code is: {0}"
    });

    manager.RegisterTwoFactorProvider("EmailCode",
        new EmailTokenProvider<ApplicationUser>
    {
        Subject = "SecurityCode",
        BodyFormat = "Your security code is {0}"
    });

    manager.EmailService = new EmailService();
    manager.SmsService = new SmsService();

    var dataProtectionProvider = options.DataProtectionProvider;

    if (dataProtectionProvider != null)
    {
        manager.UserTokenProvider =
            new DataProtectorTokenProvider<ApplicationUser>(
                dataProtectionProvider.Create("ASP.NET Identity"));
    }

    return manager;
}

public virtual async Task<IdentityResult> AddUserToRolesAsync(
    string userId, IList<string> roles)
{
    var userRoleStore = (IUserRoleStore<ApplicationUser, string>)Store;
    var user = await FindByIdAsync(userId).ConfigureAwait(false);

    if (user == null)
    {
        throw new InvalidOperationException("Invalid user Id");
    }

    var userRoles = await userRoleStore
        .GetRolesAsync(user)
        .ConfigureAwait(false);

    // Add user to each role using UserRoleStore
    foreach (var role in roles.Where(role => !userRoles.Contains(role)))
    {
        await userRoleStore.AddToRoleAsync(user, role).ConfigureAwait(false);
    }
}

```

```
// Call update once when all roles are added
return await UpdateAsync(user).ConfigureAwait(false);
}

public virtual async Task<IdentityResult> RemoveUserFromRolesAsync(
    string userId, IList<string> roles)
{
    var userRoleStore = (IUserRoleStore<ApplicationUser, string>) Store;
    var user = await FindByIdAsync(userId).ConfigureAwait(false);

    if (user == null)
    {
        throw new InvalidOperationException("Invalid user Id");
    }

    var userRoles = await userRoleStore
        .GetRolesAsync(user)
        .ConfigureAwait(false);

    // Remove user to each role using UserRoleStore
    foreach (var role in roles.Where(userRoles.Contains))
    {
        await userRoleStore
            .RemoveFromRoleAsync(user, role)
            .ConfigureAwait(false);
    }

    // Call update once when all roles are removed
    return await UpdateAsync(user).ConfigureAwait(false);
}
```

قطعه کد بالا گرچه شاید به نظر طولانی بیاید، اما در کل، کلاس ApplicationUserManager توابع مهمی را برای مدیریت کاربران فراهم می‌کند: ایجاد کاربران جدید، انتساب کاربران به نقش‌ها و حذف کاربران از نقش‌ها. این کلاس بخودی خود از کلاس `UserManager< ApplicationUser >` ارث بری می‌کند بنابراین تمام قابلیت‌های `UserManager` در این کلاس هم در دسترس است. اگر به متدهای `Create()` دقت کنید می‌بینید که وله‌ای از نوع `ApplicationUserManager` باز می‌گرداند. بیشتر تنظیمات پیکربندی و تنظیمات پیش‌فرض کاربران شما در این متدهای اتفاق می‌افتد.

مورد حائز اهمیت بعدی در متد `Create()` فراخوانی `context.Get<ApplicationDbContext>()` است. بیاد بیاورید که پیشتر نگاهی به متد `Create()` داشتیم که چند فراخوانی `CreatePerOwinContext` داشت که توسط آنها `Callback` های راجیسiter می کردیم. فراخوانی متد `(())` `context.Get<ApplicationDbContext>` این `Callback` ها را صدا می زند، که در اینجا فراخوانی متد `ApplicationDbContext.Create()` استانیک خواهد بود. در ادامه بیشتر در باره این قسمت خواهید خواهند.

اگر دقت کنید می‌بینید که احراز هویت، تعیین سطوح دسترسی و تنظیمات مدیریتی و مقادیر پیش فرض آنها در متدهای `Create()`، `Update()` و `Delete()` کلاس `ApplicationUserManager` انجام می‌شوند و سپس وله ای از نوع خود کلاس `ApplicationUser` بازگشت داده می‌شود. همچنین سرویس‌های احراز هویت دو مرحله‌ای نیز در همین مرحله پیکربندی می‌شوند. اکثر پیکربندی‌ها و تنظیمات نیازی به توضیح ندارند و قابل درک هستند. اما احراز هویت دو مرحله‌ای نیاز به بررسی عمیق‌تری دارد. در ادامه به این قسمت خواهیم پرداخت. اما پیش از آن نگاه، به کلاس `ApplicationRoleManager` ساندایم.

```
public class ApplicationRoleManager : RoleManager<IdentityRole>
{
    public ApplicationRoleManager(IRoleStore<IdentityRole, string> roleStore)
        : base(roleStore)
    {
    }

    public static ApplicationRoleManager Create(
        IdentityFactoryOptions<ApplicationRoleManager> options,
        IOWinContext context)
    {
        var manager = new ApplicationRoleManager(
            new RoleStore<IdentityRole>(
                context.Get<ApplicationDbContext>()));

        return manager;
    }
}
```

```

    }
}
```

مانند کلاس `RoleManager<IdentityRole>` `ApplicationRoleManager` مشاهده می‌کنید که کلاس `ApplicationUserManager` از ارث بری می‌کند. بنابراین تمام قابلیت‌های کلاس پایه نیز در این کلاس در دسترس هستند. یکبار دیگر متدهای `Create()` را مشاهده می‌کنید که وله‌ای از نوع خود کلاس بر می‌گرداند.

سروریس‌های ایمیل و پیامک برای احراز هویت دو مرحله‌ای و تایید حساب‌های کاربری

دو کلاس دیگری که در فایل `IdentityConfig.cs` وجود دارند کلاس‌های `EmailService` و `SmsService` هستند. بصورت پیش فرض این کلاس‌ها تنها یک wrapper هستند که می‌توانید با توسعه آنها سرویس‌های مورد نیاز برای احراز هویت دو مرحله‌ای و تایید حساب‌های کاربری را بسازید.

```

public class EmailService : IIIdentityMessageService
{
    public Task SendAsync(IdentityMessage message)
    {
        // Plug in your email service here to send an email.
        return Task.FromResult(0);
    }
}
```

```

public class SmsService : IIIdentityMessageService
{
    public Task SendAsync(IdentityMessage message)
    {
        // Plug in your sms service here to send a text message.
        return Task.FromResult(0);
    }
}
```

دقت کنید که هر دو این کلاس‌ها قرارداد واحدی را بنام `IIIdentityMessageService` پیاده سازی می‌کنند. همچنین قطعه کد زیر را از متدهای `ApplicationUserManager.Create()` بیاد آورید.

```

// Register two factor authentication providers. This application uses
// Phone and Emails as a step of receiving a code for verifying
// the user. You can write your own provider and plug in here.
manager.RegisterTwoFactorProvider("PhoneCode",
    new PhoneNumberTokenProvider< ApplicationUser >
{
    MessageFormat = "Your security code is: {0}"
});

manager.RegisterTwoFactorProvider("EmailCode",
    new EmailTokenProvider< ApplicationUser >
{
    Subject = "SecurityCode",
    BodyFormat = "Your security code is {0}"
});

manager.EmailService = new EmailService();
manager.SmsService = new SmsService();
```

همانطور که می‌بینید در متدهای `Create()` کلاس‌های `EmailService` و `SmsService` وله‌ای سازی شده و توسط خواص مرتبط به وله‌ای ارجاع می‌شوند.

SignIn کمکی

هنگام توسعه پروژه مثال `Identity`, تیم توسعه دهندگان کلاسی کمکی برای ما ساخته‌اند که فرامین عمومی احراز هویت کاربران و ورود آنها به اپلیکیشن را توسط یک API ساده فراهم می‌سازد. برای آشنایی با نحوه استفاده از این متدهای می‌توانیم به کنترلر `AccountController` در پوشش `Controllers` مراجعه کنیم. اما پیش از آن بگذارید نگاهی به خود کلاس `SignInHelper` داشته باشید.

```

public class SignInHelper
{
    public SignInHelper(
        ApplicationUserManager userManager,
        IAuthenticationManager authManager)
    {
        UserManager = userManager;
        AuthenticationManager = authManager;
    }

    public ApplicationUserManager UserManager { get; private set; }
    public IAuthenticationManager AuthenticationManager { get; private set; }

    public async Task SignInAsync(
        ApplicationUser user,
        bool isPersistent,
        bool rememberBrowser)
    {
        // Clear any partial cookies from external or two factor partial sign ins
        AuthenticationManager.SignOut(
            DefaultAuthenticationTypes.ExternalCookie,
            DefaultAuthenticationTypes.TwoFactorCookie);

        var userIdentity = await user.GenerateUserIdentityAsync(UserManager);

        if (rememberBrowser)
        {
            var rememberBrowserIdentity =
                AuthenticationManager.CreateTwoFactorRememberBrowserIdentity(user.Id);

            AuthenticationManager.SignIn(
                new AuthenticationProperties { IsPersistent = isPersistent },
                userIdentity,
                rememberBrowserIdentity);
        }
        else
        {
            AuthenticationManager.SignIn(
                new AuthenticationProperties { IsPersistent = isPersistent },
                userIdentity);
        }
    }

    public async Task<bool> SendTwoFactorCode(string provider)
    {
        var userId = await GetVerifiedUserIdAsync();

        if (userId == null)
        {
            return false;
        }

        var token = await UserManager.GenerateTwoFactorTokenAsync(userId, provider);

        // See IdentityConfig.cs to plug in Email/SMS services to actually send the code
        await UserManager.NotifyTwoFactorTokenAsync(userId, provider, token);

        return true;
    }

    public async Task<string> GetVerifiedUserIdAsync()
    {
        var result = await AuthenticationManager.AuthenticateAsync(
            DefaultAuthenticationTypes.TwoFactorCookie);

        if (result != null && result.Identity != null
            && !String.IsNullOrEmpty(result.Identity.GetUserId()))
        {
            return result.Identity.GetUserId();
        }

        return null;
    }

    public async Task<bool> HasBeenVerified()
    {
        return await GetVerifiedUserIdAsync() != null;
    }
}

```

```

public async Task<SignInStatus> TwoFactorSignIn(
    string provider,
    string code,
    bool isPersistent,
    bool rememberBrowser)
{
    var userId = await GetVerifiedUserIdAsync();

    if (userId == null)
    {
        return SignInStatus.Failure;
    }

    var user = await UserManager.FindByIdAsync(userId);

    if (user == null)
    {
        return SignInStatus.Failure;
    }

    if (await UserManager.IsLockedOutAsync(user.Id))
    {
        return SignInStatus.LockedOut;
    }

    if (await UserManager.VerifyTwoFactorTokenAsync(user.Id, provider, code))
    {
        // When token is verified correctly, clear the access failed
        // count used for lockout
        await UserManager.ResetAccessFailedCountAsync(user.Id);
        await SignInAsync(user, isPersistent, rememberBrowser);

        return SignInStatus.Success;
    }

    // If the token is incorrect, record the failure which
    // also may cause the user to be locked out
    await UserManager.AccessFailedAsync(user.Id);

    return SignInStatus.Failure;
}

public async Task<SignInStatus> ExternalSignIn(
    ExternalLoginInfo loginInfo,
    bool isPersistent)
{
    var user = await UserManager.FindAsync(loginInfo.Login);

    if (user == null)
    {
        return SignInStatus.Failure;
    }

    if (await UserManager.IsLockedOutAsync(user.Id))
    {
        return SignInStatus.LockedOut;
    }

    return await SignInOrTwoFactor(user, isPersistent);
}

private async Task<SignInStatus> SignInOrTwoFactor(
    ApplicationUser user,
    bool isPersistent)
{
    if (await UserManager.GetTwoFactorEnabledAsync(user.Id) &&
        !await AuthenticationManager.TwoFactorBrowserRememberedAsync(user.Id))
    {
        var identity = new ClaimsIdentity(DefaultAuthenticationTypes.TwoFactorCookie);
        identity.AddClaim(new Claim(ClaimTypes.NameIdentifier, user.Id));

        AuthenticationManager.SignIn(identity);
        return SignInStatus.RequiresTwoFactorAuthentication;
    }

    await SignInAsync(user, isPersistent, false);
    return SignInStatus.Success;
}

public async Task<SignInStatus> PasswordSignIn(

```

```

        string userName,
        string password,
        bool isPersistent,
        bool shouldLockout)
{
    var user = await UserManager.FindByNameAsync(userName);

    if (user == null)
    {
        return SignInStatus.Failure;
    }

    if (await UserManager.IsLockedOutAsync(user.Id))
    {
        return SignInStatus.LockedOut;
    }

    if (await UserManager.CheckPasswordAsync(user, password))
    {
        return await SignInOrTwoFactor(user, isPersistent);
    }

    if (shouldLockout)
    {
        // If lockout is requested, increment access failed
        // count which might lock out the user
        await UserManager.AccessFailedAsync(user.Id);

        if (await UserManager.IsLockedOutAsync(user.Id))
        {
            return SignInStatus.LockedOut;
        }
    }

    return SignInStatus.Failure;
}
}

```

کد این کلاس نسبتاً طولانی است، و بررسی عمیق آنها از حوصله این مقاله خارج است. گرچه اگر به دقت یکبار این کلاس را مطالعه کنید می‌توانید برآحتی از نحوه کارکرد آن آگاه شوید. همانطور که می‌بینید اکثر متدهای این کلاس مربوط به ورود کاربران و مسئولیت‌های تعیین سطوح دسترسی است.

این متدها ویژگی‌های جدیدی که در Identity 2.0 عرضه شده اند را در بر می‌گیرند. متند آشنایی بنام (`SignInAsync()`) و متدهای دیگری که مربوط به احراز هویت دو مرحله‌ای و `external log-ins` می‌شوند. اگر به متدها دقت کنید خواهید دید که برای ورود کاربران به اپلیکیشن کارهای بیشتری نسبت به نسخه پیشین انجام می‌شود.

عنوان مثال متدهای احراز هویت در کنترلر `Login` در `AccountController` را باز کنید تا نحوه مدیریت احراز هویت در Identity 2.0 را ببینید.

```

[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Login(LoginViewModel model, string returnUrl)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }

    // This doesn't count login failures towards lockout only two factor authentication
    // To enable password failures to trigger lockout, change to shouldLockout: true
    var result = await SignInHelper.PasswordSignIn(
        model.Email,
        model.Password,
        model.RememberMe,
        shouldLockout: false);

    switch (result)
    {
        case SignInStatus.Success:
            return RedirectToLocal(returnUrl);
        case SignInStatus.LockedOut:
            return View("Lockout");
    }
}

```

```

        case SignInStatus.RequiresTwoFactorAuthentication:
            return RedirectToAction("SendCode", new { ReturnUrl = returnUrl });
        case SignInStatus.Failure:
        default:
            ModelState.AddModelError("", "Invalid login attempt.");
    }
}
}

```

مقایسه Sign-in با نسخه 1.0

در نسخه 1.0 این فریم ورک، ورود کاربران به اپلیکیشن مانند لیست زیر انجام می‌شد. اگر متده Login در کنترلر AccountController را باز کنید چنین قطعه کدی را می‌بینید.

```

[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Login(LoginViewModel model, string returnUrl)
{
    if (ModelState.IsValid)
    {
        var user = await UserManager.FindAsync(model.UserName, model.Password);

        if (user != null)
        {
            await SignInAsync(user, model.RememberMe);
            return RedirectToLocal(returnUrl);
        }
        else
        {
            ModelState.AddModelError("", "Invalid username or password.");
        }
    }

    // If we got this far, something failed, redisplay form
    return View(model);
}

```

در قطعه کد بالا متده SignInHelper در کلاس UserManager را فراخوانی می‌کنیم که مشابه قطعه کدی است که در کلاس SignInHelper دیدیم. همچنین متده SignInAsync را فراخوانی می‌کنیم که مستقیماً روی کنترلر AccountController تعریف شده است.

```

private async Task SignInAsync(ApplicationUser user, bool isPersistent)
{
    AuthenticationManager.SignOut(
        DefaultAuthenticationTypes.ExternalCookie);

    var identity = await UserManager.CreateIdentityAsync(
        user, DefaultAuthenticationTypes.ApplicationCookie);

    AuthenticationManager.SignIn(
        new AuthenticationProperties() { IsPersistent = isPersistent }, identity);
}

```

مسلمان با عرضه قابلیت‌های جدید در Identity 2.0 و تغییرات معماری این فریم ورک، پیچیدگی‌هایی معرفی می‌شود که حتی در امور ساده‌ای مانند ورود کاربران نیز قابل مشاهده است.

ApplicationDbContext

اگر از نسخه پیشین Identity در اپلیکیشن‌های ASP.NET MVC استفاده کرده باشید با کلاس ApplicationDbContext آشنا هستید. این کلاس پیاده‌سازی پیش فرض EF فریم ورک است، که اپلیکیشن شما توسط آن داده‌های مربوط به Identity را ذخیره و بازیابی می‌کند.

در پروژه مثال‌ها، تیم Identity این کلاس را بطور متفاوتی نسبت به نسخه 1.0 پیکربندی کرده‌اند. اگر فایل IdentityModels.cs را باز کنید تعاریف کلاس ApplicationDbContext را مانند لیست زیر خواهد یافت.

```

public class ApplicationDbContext : IdentityDbContext<ApplicationUser> {
    public ApplicationDbContext()
        : base("DefaultConnection", throwIfV1Schema: false) {
    }

    static ApplicationDbContext() {
        // Set the database initializer which is run once during application start
        // This seeds the database with admin user credentials and admin role
        Database.SetInitializer<ApplicationDbContext>(new ApplicationDbContextInitializer());
    }

    public static ApplicationDbContext Create() {
        return new ApplicationDbContext();
    }
}

```

قطعه کد بالا دو متد استاتیک تعریف می‌کند. یکی (`Create()`) و دیگری (`ApplicationContext()`) که سازنده دیتابیس (`database`) را تنظیم می‌کند. این متد هنگام اجرای اپلیکیشن فراخوانی می‌شود و هر پیکربندی ای که در کلاس `initializer` تعریف شده باشد را اجرا می‌کند. اگر به فایل `IdentityConfig.cs` مراجعه کنیم می‌توانیم تعاریف `ApplicationDbInitializer` این کلاس را مانند لیست زیر بباییم.

```

public class ApplicationDbContextInitializer
    : DropCreateDatabaseIfModelChanges<ApplicationDbContext>
{
    protected override void Seed(ApplicationDbContext context)
    {
        InitializeIdentityForEF(context);
        base.Seed(context);
    }

    public static void InitializeIdentityForEF(ApplicationDbContext db)
    {
        var userManager = HttpContext
            .Current.GetOwinContext()
            .GetUserManager<ApplicationUserManager>();

        var roleManager = HttpContext.Current
            .GetOwinContext()
            .Get<ApplicationRoleManager>();

        const string name = "admin@admin.com";
        const string password = "Admin@123456";
        const string roleName = "Admin";

        //Create Role Admin if it does not exist
        var role = roleManager.FindByName(roleName);

        if (role == null)
        {
            role = new IdentityRole(roleName);
            var roleresult = roleManager.Create(role);
        }

        var user = userManager.FindByName(name);

        if (user == null)
        {
            user = new ApplicationUser { UserName = name, Email = name };

            var result = userManager.Create(user, password);
            result = userManager.SetLockoutEnabled(user.Id, false);
        }

        // Add user admin to Role Admin if not already added
        var rolesForUser = userManager.GetRoles(user.Id);

        if (!rolesForUser.Contains(role.Name))
        {
            var result = userManager.AddToRole(user.Id, role.Name);
        }
    }
}

```

پیکربندی جاری در صورتی که مدل موجودیت‌ها تغییر کنند دیتابیس را پاک کرده و مجدداً ایجاد می‌کند. در غیر اینصورت از دیتابیس موجود استفاده خواهد شد. اگر بخواهیم با هر بار اجرای اپلیکیشن دیتابیس از نو ساخته شود، می‌توانیم کلاس مربوطه را به `DropCreateDatabaseAlways<ApplicationDbContext>` تغییر دهیم. عنوان مثال هنگام توسعه اپلیکیشن و بمنظور تست می‌توانیم از این رویکرد استفاده کنیم تا هر بار با دیتابیس (تقریباً) خالی شروع کنیم.

نکته حائز اهمیت دیگر متدهای `InitializeIdentityForEF()` است. این متدهای مشابه متدهای `Seed()` انجام می‌دهد که هنگام استفاده از مهاجرت‌ها (`Migrations`) از آن استفاده می‌کنیم. در این متدهای توانید رکوردهای اولیه ای را در دیتابیس ثبت کنید. همانطور که مشاهده می‌کنید در قطعه کد بالا نقشی مدیریتی بنام `Admin` ایجاد شده و کاربر جدیدی با اطلاعاتی پیش فرض ساخته می‌شود که در آخر به این نقش منتنسب می‌گردد. با انجام این مراحل، پس از اجرای اولیه اپلیکیشن کاربری با سطح دسترسی مدیر در اختیار خواهیم داشت که برای تست اپلیکیشن بسیار مفید خواهد بود.

در این مقاله نگاهی اجمالی به 2.0 Identity در پروژه‌های مختلف فریم‌ورک و نحوه پیکربندی آنها را بررسی کردیم و با تغییرات و قابلیت‌های جدید به اختصار آشنا شدیم. در مقالات بعدی بررسی‌های عمیق‌تر خواهیم داشت و با نحوه استفاده و پیاده‌سازی قسمت‌های مختلف این فریم‌ورک آشنا خواهیم شد.

مطالعه بیشتر

[MSDN: Announcing RTM of ASP.NET Identity 2.0.0](#)

[Identity 2.0 on Codeplex](#)

نظرات خوانندگان

نویسنده: محمد زارع
تاریخ: ۱۳۹۳/۰۶/۰۷ ۱۹:۷

سلام. من از ASP.NET Identity توی پروژم استفاده کردم و همه چیز درست کار میکنه (در محیط توسعه). سوالی که دارم مربوط به بررسی نقش‌های کاربر هستش. میخواستم بدونم استفاده از `HttpContext.Current.User.IsInRole` توی `HttpContext.Current.User` مشکلی در استفاده ازش نمیبینم ولی چندجا توی اینترنت دیدم که گفته هم درست هست یا نه؟ من الان توی محیط توسعه هیچ مشکلی در استفاده ازش نمیبینم ولی چندجا توی اینترنت دیدم که گفته بودن استفاده از این روش قدیمی هستش و بعد از `Publish` کردن به مشکل میخوره و بهتر هستش که از `UserManager.IsInRole` استفاده بشه.

ممنون میشم اگر راهنمایی بفرمایید.
<https://aspnetidentity.codeplex.com/workitem/2189>
<http://codeverge.com/asp.net.security/user.isinrole-not-working-in-production/73992>

نویسنده: هومن
تاریخ: ۱۳۹۳/۰۶/۱۱ ۱۳:۲۴

سلام
نمونه پروژه‌ای وجود داره که با استفاده از 2.0 identity سیستم گروپ بیس رو پیاده سازی کرده باشه؟
نمونه 1.0 identity رو پیدا کردم ولی هر کاری کردم نتونستم تبدیلش کنم به نسخه دوم

نویسنده: ایلیا
تاریخ: ۱۳۹۳/۰۸/۱۹ ۱۳:۲۰

اگه در حالت EF Code First نباشیم و دیتابیس‌مون به صورت Database First باشه و دیتاکسیسمون ADO Entity Data Model همون edmx باشه. چطوری می‌تونیم از Identity استفاده کنیم؟ (البته از 5 MVC و 6 EF دارم استفاده می‌کنم و دوم اینکه جداول لازم رو خودمون باید بسازیم یا مثل Membership خودش می‌سازیم؟)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۱۹ ۱۴:۰

[Use ASP.NET Identity on existing DB-Model PART 1](#) -
[ASP.NET Identity 2.0 on existing DB-model PART 2](#) -

نویسنده: نرگس حقیقی
تاریخ: ۱۳۹۳/۰۹/۲۲ ۱۲:۳

سلام؛ آیا میشه در یه پروژه MVC که از EF6 برای ارتباط با SQL Server استفاده میکنه و بدلیل زیاد بودن تعداد جداول از class Identity Code Library جدآگاهه‌ای بهمنظور Migrate کردن بانک استفاده میکنه، از Identity هم استفاده کرد؛ با وجود اینکه به کاربران در جداول دیگر هم نیاز هست؟ اصلا این کار ممکنه؟

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۳/۰۹/۲۲ ۱۸:۵۳

سرویس خوبیه اما کمی کار باهاش در عین سادگی پیچیده اس.
مثلاً پیاده سازی اون و تفکیک تو لایه‌های سرویس و دامین و لایه نمایش واقعاً ادم گیج میکنه
هر چند زیاد هم نمی‌توان الگوی واحد کار باهاش به کار برد - البته نظر شخصی منه

نویسنده: وحید نصیری

« [اعمال تزریق وابستگی‌ها به مثال رسمی ASP.NET Identity](#) »

نویسنده: وحید نصیری
تاریخ: ۱۳:۴۳ ۱۳۹۳/۰۹/۲۹

بله. مراجعه کنید به این مثال: « [اعمال تزریق وابستگی‌ها به مثال رسمی ASP.NET Identity](#) »

نویسنده: آذرنیوا
تاریخ: ۷:۲۴ ۱۳۹۳/۱۱/۲۷

نحوه تعیین سطح دسترسی در `asp form` برای متدها و فرم‌ها مثل کاری که فیلتر `authorize` در `mvc` انجام می‌دهد

نویسنده: محسن خان
تاریخ: ۱۰:۱۹ ۱۳۹۳/۱۱/۲۷

[افزودن Web Forms به یک پروژه ASP.NET Identity](#)

یکی از روش‌های اطلاع رسانی به سایت‌های دیگر در مورد لینک دادن به آن‌ها در سایت خود است. برای مثال من لینکی از یکی از مطالب شما را در متن جاری خودم قرار می‌دهم. سپس به وسیله‌ی ارسال یک ping، در مورد انجام اینکار به شما اطلاع رسانی می‌کنم. حاصل آن عموماً قسمت معروف ping-backs سایت‌ها است. این مورد نیز یکی از روش‌های مؤثر SEO در گرفتن backlink است و تبلیغ محتوا.

کار کردن با پروتکل Ping-back آنچنان ساده نیست؛ از این جهت که تبادل ارتباطات آن با پروتکل XML-RPC انجام می‌شود. - XML-RPC نیز توسط PHP کارها بیشتر مورد استفاده قرار می‌گیرد؛ بجای استفاده از پروتکل‌های استاندارد وب سرویس‌ها مانند Soap و امثال آن. پیاده سازی‌های ابتدایی Pingback نیز مرتبط است به Wordpress که با PHP تهیه شده‌است. در ادامه نگاهی خواهیم داشت به جزئیات پیاده سازی ارسال ping-back توسط برنامه‌های ASP.NET.

یافتن آدرس وب سرویس سایت پذیرای Pingback

اولین قدم در پیاده سازی Pingback، یافتن آدرسی است که باید اطلاعات مورد نظر را به آن ارسال کرد. این آدرس عموماً به دو طریق ارائه می‌شود:

- الف) در هدری به نام pingback و یا x-pingback
- ب) در قسمتی از کدهای HTML صفحه به شکل

```
<link rel="pingback" href="pingback server">
```

برای مثال اگر به و بلاگ‌های MSDN دقت کنید، هدر x-pingback را می‌توانید در خروجی وب سرور آن‌ها مشاهده کنید:

The screenshot shows a browser page titled "Latest Developments in General Purpose GPU Programming with F#" from blogs.msdn.com. Below the page, the Fiddler interface displays a list of network requests. One request is selected: "GET latest-developments-in-gp". The "Headers" tab is active, showing various response headers. Two specific headers are circled in red: "Response Headers" and "X-Pingback". The "X-Pingback" header is set to "http://blogs.msdn.com/b/dsyme/pingback.aspx". Other visible headers include Cache-Control, Content-Encoding, Content-Length, Content-Type, Date, Expires, Pragma, Server, Set-Cookie, Vary, X-AspNet-Version, and X-Frame-Options.

همانطور که ملاحظه می‌کنید، نیاز است Response header را آنالیز کنیم.

```
private Uri findPingbackServiceUri()
{
    var request = (HttpWebRequest)WebRequest.Create(_targetUri);
    request.UserAgent = UserAgent;
    request.Timeout = Timeout;
    request.ReadWriteTimeout = Timeout;
    request.Method = WebRequestMethods.Http.Get;
    request.AutomaticDecompression = DecompressionMethods.GZip | DecompressionMethods.Deflate;
    using (var response = request.GetResponse() as HttpWebResponse)
    {
        if (response == null) return null;

        var url = extractPingbackServiceUriFromHeaders(response);
        if (url != null)
            return url;

        if (!isResponseHtml(response))
            return null;

        using (var reader = new StreamReader(response.GetResponseStream()))
        {
            return extractPingbackServiceUriFromPage(reader.ReadToEnd());
        }
    }
}

private static Uri extractPingbackServiceUriFromHeaders(WebResponse response)
{
    var pingUrl = response.Headers.AllKeys.FirstOrDefault(header =>
        header.Equals("x-pingback", StringComparison.OrdinalIgnoreCase) ||
        header.Equals("pingback", StringComparison.OrdinalIgnoreCase));
}
```

```

        return getValidAbsoluteUri(pingUrl);
    }

    private static Uri extractPingbackServiceUriFromPage(string content)
    {
        if (string.IsNullOrWhiteSpace(content)) return null;
        var regex = new Regex(@"(?s)<link\ srel=""pingback""\ href=""(.+?)""", RegexOptions.IgnoreCase);
        var match = regex.Match(content);
        return (!match.Success || match.Groups.Count < 2) ? null : getValidAbsoluteUri(match.Groups[1].Value);
    }

    private static Uri getValidAbsoluteUri(string url)
    {
        Uri absoluteUri;
        return string.IsNullOrWhiteSpace(url) || !Uri.TryCreate(url, UriKind.Absolute, out absoluteUri) ? null : absoluteUri;
    }

    private static bool isResponseHtml(WebResponse response)
    {
        var contentTypeKey = response.Headers.AllKeys.FirstOrDefault(header =>
            header.Equals("content-type", StringComparison.OrdinalIgnoreCase));
        return !string.IsNullOrWhiteSpace(contentTypeKey) &&
            response.Headers[contentTypeKey].StartsWith("text/html", StringComparison.OrdinalIgnoreCase);
    }
}

```

نحوه استخراج آدرس سرویس Pingback یک سایت را در کدهای فوق ملاحظه می‌کنید. آدرسی است از یک سایت دیگر که در سایت ما درج شده است. زمانیکه این صفحه را درخواست می‌کنیم، `_targetUri` حاصل می‌تواند حاوی کلید `x-pingback` باشد یا خیر. اگر بلی، همینجا کار پایان می‌یابد. فقط باید مطمئن شد که این آدرس مطلق است و نه نسبی. به همین جهت در متدهای `getValidAbsoluteUri`، `isResponseHtml`، `UriKind.Absolute` انجام شده است.

اگر هدر فاقد کلید `x-pingback` باشد، قسمت `b` را باید بررسی کرد. یعنی نیاز است محتوای `HTML` صفحه را برای یافتن `link` بررسی کنیم. همچنین باید دقت داشت که پیش از اینکار نیاز است حتماً بررسی `isResponseHtml` صورت گیرد. برای مثال در سایت شما لینکی به یک فایل 2 گیگابایتی `SQL Server` درج شده است. در این حالت نباید ابتدا 2 گیگابایت فایل دریافت شود و سپس بررسی کنیم که آیا محتوای آن حاوی `link rel=pingback` است یا خیر. اگر محتوای ارسالی از نوع `text/html` بود، آنگاه کار دریافت محتوای لینک انجام خواهد شد.

ارسال Ping به آدرس سرویس

اکنون که آدرس سرویس `pingback` یک سایت را یافته‌ایم، کافی است `ping` ای را به آن ارسال کنیم:

```

public void Send()
{
    var pingUrl = findPingbackServiceUri();
    if (pingUrl == null)
        throw new NotSupportedException(string.Format("{0} doesn't support pingback.", _targetUri.Host));

    sendPing(pingUrl);
}

private void sendPing(Uri pingUrl)
{
    var request = (HttpWebRequest)WebRequest.Create(pingUrl);
    request.UserAgent = UserAgent;
    request.Timeout = Timeout;
    request.ReadWriteTimeout = Timeout;
    request.Method = WebRequestMethods.Http.Post;
    request.ContentType = "text/xml";
    request.ProtocolVersion = HttpVersion.Version11;
    makeXmlRpcRequest(request);
    using (var response = (HttpWebResponse)request.GetResponse())
    {
        response.Close();
    }
}

```

```

        }

    private void makeXmlRpcRequest(WebRequest request)
    {
        var stream = request.GetRequestStream();
        using (var writer = new XmlTextWriter(stream, Encoding.ASCII))
        {
            writer.WriteStartDocument(true);
            writer.WriteStartElement("methodCall");
            writer.WriteElementString("methodName", "pingback.ping");
            writer.WriteStartElement("params");

            writer.WriteStartElement("param");
            writer.WriteStartElement("value");
            writer.WriteLineString("string", Uri.EscapeUriString(_sourceUri.ToString()));
            writer.WriteEndElement();
            writer.WriteEndElement();

            writer.WriteStartElement("param");
            writer.WriteStartElement("value");
            writer.WriteLineString("string", Uri.EscapeUriString(_targetUri.ToString()));
            writer.WriteEndElement();
            writer.WriteEndElement();

            writer.WriteEndElement();
            writer.WriteEndElement();
        }
    }
}

```

اینبار `HttpWebRequest` تشکیل شده از نوع `post` است و نه `get`. همچنین مقداری را که باید ارسال کنیم نیاز است مطابق پروتکل XML-RPC باشد. برای کار با XML-RPC در دات نت یا میتوان از کتابخانه [Cook Computing's XML-RPC.Net](#) استفاده کرد و یا مطابق کدهای فوق، دستورات آنرا توسط یک `XmlTextWriter` کنار هم قرار داد و نهایتاً در درخواست `Post` ارسالی درج کرد. در اینجا `sourceUri` آدرس صفحه‌ای در سایت ما است که `targetUri` ای (آدرسی از سایت دیگر) در آن درج شده‌است. در یک `pingback`، صرفاً این دو آدرس به سرویس دریافت کننده‌ی `pingback` ارسال می‌شوند. سپس سایت دریافت کننده‌ی `ping`، ابتدا `sourceUri` را دریافت می‌کند تا عنوان آن را استخراج کند و همچنین بررسی می‌کند که آیا `targetUri` در آن درج شده‌است یا خیر (آیا `spam` است یا خیر)؟ تا اینجا اگر این مراحل را کنار هم قرار دهیم به کلاس `Pingback` ذیل خواهیم رسید:

[Pingback.cs](#)

نحوه استفاده از کلاس `Pingback` تهیه شده

کار ارسال `Pingback` عموماً به این نحو است: هر زمانیکه مطلبی یا یکی از نظرات آن، ثبت یا ویرایش می‌شوند، نیاز است `Pingback`‌های آن ارسال شوند. بنابراین تنها کاری که باید انجام شود، استخراج لینک‌های خارجی یک صفحه و سپس فرآخوانی متدهای `Send` است. یافتن لینک‌های یک محتوا را نیز می‌توان مانند متد `extractPingbackServiceUriFormPage` فوق، توسط یک `Regex` انجام داد و یا حتی با استفاده از کتابخانه [HTML Agility Pack](#) :

```

var doc = new HtmlWeb().Load(url);
var linkTags = doc.DocumentNode.Descendants("link");
var linkedPages = doc.DocumentNode.Descendants("a")
    .Select(a => a.GetAttributeValue("href", null))
    .Where(u => !String.IsNullOrEmpty(u));

```

چندی قبل، معرفی ادیتور سبک وزن و مناسبی را تحت عنوان [RedActor](#)، در این سایت ملاحظه کردید. زمانیکه این کار انجام شد، این ادیتور هم رایگان بود و هم سورس آخرین نگارش آن به سادگی در دسترس. بعد از مدتی، هر دو ویژگی یاد شده‌ی RedActor حذف شدند. پس از آن ادیتور مدرن و بسیار مناسب دیگری به نام [Froala](#) منتشر شد که هرچند نگارش‌های تجاری هم دارد، اما سورس آخرین نگارش آن برای عموم قابل دریافت است. در ادامه مروری خواهیم داشت بر نحوه‌ی یکپارچه سازی آن با ASP.NET Web forms و همچنین MVC

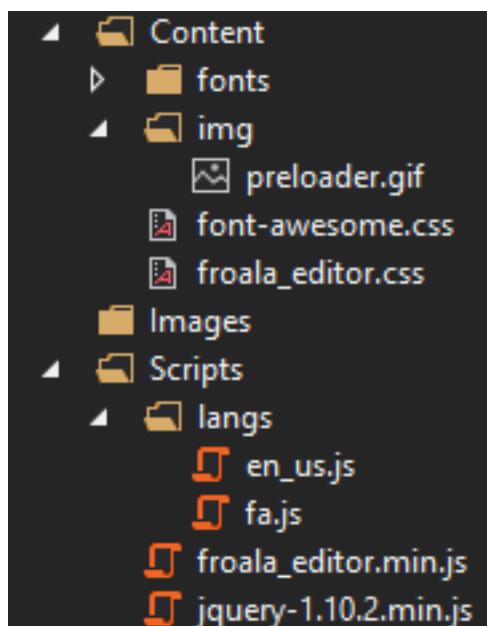
Froala WYSIWYG Editor
دریافت آخرین نگارش این ادیتور وب می‌توانید به سایت آن، قسمت دریافت فایل‌ها مراجعه نمایید.

<http://editor.froala.com/download>

و یا به این آدرس مراجعه کنید:

<https://github.com/froala/wysiwyg-editor/releases>

ساختار پروژه و نحوه کپی فایل‌های آن
در هر دو مثالی که فایل‌های آن را از انتهای بحث می‌توانید دریافت کنید، این ساختار رعایت شده است:



فایل‌های CSS و فونت‌های آن، در پوششی Content قرار گرفته‌اند.
فایل‌های اسکریپت و زبان آن (که دارای زبان فارسی هم هست) در پوششی Scripts کپی شده‌اند.

یک نکته
فایل CSS را نیاز است کمی اصلاح کنید. مسیر پوششی فونت‌های آن اکنون با fonts شروع می‌شود.

تنظیمات اولیه

تفاوتی نمی‌کند که از وب فرم‌ها استفاده می‌کنید یا MVC، نحوه‌ی تعریف و افزودن پیش‌نیازهای این ادیتور به نحو ذیل است:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>

    <link href="Content/font-awesome.css" rel="stylesheet" />
    <link href="Content/froala_editor.css" rel="stylesheet" />

    <script src="Scripts/jquery-1.10.2.min.js"></script>
    <script src="Scripts/froala_editor.min.js"></script>
    <script src="Scripts/langs/fa.js"></script>
</head>
<body>
    <form id="form1" runat="server">
        </form>
</body>
</html>
```

دو فایل CSS دارد (آیکن‌های آن و همچنین شیوه نامه‌ی اصلی ادیتور) به همراه سه فایل JS (جی‌کوئری، ادیتور و فایل زبان فارسی آن) که باید در فایل layout یا master سایت اضافه شوند.

استفاده از ASP.NET MVC در Froala WYSIWYG Editor

در ادامه نحوه‌ی فعال سازی ادیتور وب Froala View را در یک برگه‌ی برنامه‌های MVC ملاحظه می‌کنید:

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

<style type="text/css">
    /* تنظیم فونت پیش فرض ادیتور */
    .froala-element {
    }
</style>

@using (Html.BeginForm(actionName: "Index", controllerName: "Home"))
{
    @Html.TextArea(name: "Editor1")
    <input type="submit" value="ارسال" />
}

@section Scripts
{
    <script type="text/javascript">
        $(function () {
            $('#Editor1').editable({
                buttons: ["bold", "italic", "underline", "strikeThrough", "fontFamily",
                    "fontSize", "color", "formatBlock", "align", "insertOrderedList",
                    "insertUnorderedList", "outdent", "indent", "selectAll", "createLink",
                    "insertImage", "insertVideo", "undo", "redo", "html", "save",
                    "insertHorizontalRule"],
                inlineMode: false,
                inverseSkin: true,
                preloaderSrc: '@Url.Content("~/Content/img/preloader.gif")',
                allowedImageTypes: ["jpeg", "jpg", "png"],
                height: 300,
                language: "fa",
                direction: "rtl",
                fontList: ["Tahoma, Geneva", "Arial, Helvetica", "Impact, Charcoal"],
                autosave: true,
                autosaveInterval: 2500,
                saveURL: '@Url.Action("FroalaAutoSave", "Home")',
                saveParams: { postId: "123" },
                spellcheck: true,
                plainPaste: true,
                imageButtons: ["removeImage", "replaceImage", "linkImage"],
                borderColor: '#00008b',
                imageUploadURL: '@Url.Action("FroalaUploadImage", "Home")',
                imageParams: { postId: "123" },
                enableScript: false
            });
        });
    </script>
}
```

```
</script>
}
```

اگر می‌خواهید فونت پیش فرض آن را تنظیم کنید، باید مطابق کدهای ابتدای فایل، ویژگی‌های froala-element سپس این ادیتور را بر روی المان TextArea قرار گرفته در صفحه، فعال می‌کنیم.

در قسمت مقادیر buttons، تمام حالات ممکن پیش بینی شده‌اند. هر کدام را که نیاز ندارید، حذف کنید. نحوه‌ی تعریف زبان و راست به چپ بودن این ادیتور را با مقدار دهی پارامترهای direction و language ملاحظه می‌کنید.

پارامترهای autosave، saveURL و saveParams کار تنظیم ارسال خودکار محتوای ادیتور را جهت ذخیره‌ی آن در سرور به عهده دارند. بر اساس مقدار autosaveInterval می‌توان مشخص کرد که هر چند میلی ثانیه یکبار این کار باید انجام شود.

```
/// <summary>
    ذخیره سازی خودکار ///
  </summary>
  [HttpPost]
  [ValidateInput(false)]
  public ActionResult FroalaAutoSave(string body, int? postId)
  {
    //todo: save body ...
    return new EmptyResult();
  }
```

در قسمت سمت سرور هم می‌توان این مقادیر ارسالی را در اکشن متدهای ذخیره سازی ارسالی دریافت کرد. چون قرار است تگ‌های HTML به سرور ارسال شوند، ویژگی ValidateInput به false تنظیم شده‌است. آن، برای مقدار دهی پارامترهای اضافی است که نیاز می‌باشد تا به سرور ارسال شوند. مثلاً شماره مطلب جاری نیز به سرور ارسال گردد.

در اینجا نام پارامتری که ارسال می‌گردد، دقیقاً مساوی body است. بنابراین آن را تغییر ندهید.

پارامترهای imageParams و imageUploadURL برای فعال سازی ذخیره تصاویر آن در سرور کاربرد دارند. اکشن متدهای آن به نحو ذیل می‌تواند تعریف شود:

```
// todo: مسایل امنیتی آپلود را فراموش نکنید
  /// <summary>
    ذخیره سازی تصاویر ارسالی
  </summary>
  [HttpPost]
  public ActionResult FroalaUploadImage(HttpPostedFileBase file, int? postId)
  {
    تغییر ندهید
    var fileName = Path.GetFileName(file.FileName);
    var rootPath = Server.MapPath("~/images/");
    file.SaveAs(Path.Combine(rootPath, fileName));
    return Json(new { link = "images/" + fileName }, JsonRequestBehavior.AllowGet);
  }
```

در اینجا نام پارامتری که به سرور ارسال می‌گردد، دقیقاً معادل file است. بنابراین آن را تغییر ندهید. خروجی آن برای مشخص سازی محل ذخیره سازی تصویر در سرور باید یک خروجی JSON دارای خاصیت link باشد (این مسیر، یک مسیر نسبی است؛ نسبت به ریشه سایت). آن برای مقدار دهی پارامترهای اضافی است که نیاز می‌باشد تا به سرور ارسال شوند. مثلاً شماره مطلب جاری نیز به سرور ارسال گردد.

استفاده از Froala WYSIWYG Editor در ASP.NET Web forms

تمام نکاتی که در قسمت تنظیمات ASP.NET MVC در مورد ویژگی‌های سمت کلاینت این ادیتور ذکر شد، در مورد وب فرم‌ها نیز صادق است. فقط قسمت مدیریت سمت سرور آن اندکی تفاوت دارد.

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
```

```
ValidateRequest="false"
EnableEventValidation="false"
AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="FroalaWebFormsTest.Default" %>
<%--اعتبارسنجی ورودی غیرفعال شده چون باید تگ ارسال شود--%>
<%--همچنین در وب کانفیگ هم تنظیم دیگری نیاز دارد--%>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <%--حالت کلاینت آی دی بهتر است تنظیم شود در اینجا--%>
    <asp:TextBox ID="txtEditor" ClientIDMode="Static"
        runat="server" Height="199px" TextMode="MultiLine" Width="447px"></asp:TextBox>
    <br />
    <asp:Button ID="btnSave" runat="server" OnClick="btnSave_Click" Text="ارسال" />
    <style type="text/css">
        /*تنظیم فونت پیش فرض ادیتور*/
        .froala-element {
        }
    </style>
    <script type="text/javascript">
        $(function () {
            $('#txtEditor').editable({
                buttons: ["bold", "italic", "underline", "strikeThrough", "fontFamily",
                    "fontSize", "color", "formatBlock", "align", "insertOrderedList",
                    "insertUnorderedList", "outdent", "indent", "selectAll", "createLink",
                    "insertImage", "insertVideo", "undo", "redo", "html", "save",
                    "inserthorizontalrule"],
                inlineMode: false,
                inverseSkin: true,
                preloaderSrc: 'Content/img/preloader.gif',
                allowedImageTypes: ["jpeg", "jpg", "png"],
                height: 300,
                language: "fa",
                direction: "rtl",
                fontlist: ["Tahoma, Geneva", "Arial, Helvetica", "Impact, Charcoal"],
                autosave: true,
                autosaveInterval: 2500,
                saveURL: 'FroalaHandler.ashx',
                saveParams: { postId: "123" },
                spellcheck: true,
                plainPaste: true,
                imageButtons: ["removeImage", "replaceImage", "linkImage"],
                borderColor: '#00008b',
                imageUploadURL: 'FroalaHandler.ashx',
                imageParams: { postId: "123" },
                enableScript: false
            });
        });
    </script>
</asp:Content>
```

همانطور که ملاحظه می‌کنید، ValidateRequest صفحه به false تنظیم شده و همچنین در وب کانفیگ httpRuntime به نگارش 2 requestValidationMode تنظیم گردیده است تا بتوان توسط این ادیتور تگ‌های ارسالی را به سرور ارسال کرد. به علاوه ClientIDMode=Static نیز تنظیم شده است، تا بتوان از ID تکست باکس قرار گرفته در صفحه، به سادگی در کدهای سمت کاربر جی‌کوئری استفاده کرد.

اگر دقت کرده باشید، save url اینبار به فایل FroalaHandler.ashx اشاره می‌کند. محتوای این Genric handler را ذیل مشاهده می‌کنید:

```
using System.IO;
using System.Web;
using System.Web.Script.Serialization;

namespace FroalaWebFormsTest
{
    public class FroalaHandler : IHttpHandler
    {
        //todo: برای اینکارها بهتر است از ووب ای پی آی استفاده شود
        //todo: یا دو هندر مجزا یکی برای تصاویر و دیگری برای ذخیره سازی متن

        public void ProcessRequest(HttpContext context)
        {

```

```

var body = context.Request.Form["body"];
var postId = context.Request.Form["postId"];
if (!string.IsNullOrWhiteSpace(body) && !string.IsNullOrWhiteSpace(postId))
{
    //todo: save changes

    context.Response.ContentType = "text/plain";
    context.Response.Write("");
    context.Response.End();
}

var files = context.Request.Files;
if (files.Keys.Count > 0)
{
    foreach (string fileKey in files)
    {
        var file = context.Request.Files[fileKey];
        if (file == null || file.ContentLength == 0)
            continue;

        در اینجا مسایل امنیتی آپلود فراموش نشود
        var fileName = Path.GetFileName(file.FileName);
        var rootPath = context.Server.MapPath("~/images/");
        file.SaveAs(Path.Combine(rootPath, fileName));
    }
}

var json = new JavaScriptSerializer().Serialize(new { link = "images/" + fileName });
البته اینجا یک فایل بیشتر ارسال نمی‌شود
context.Response.ContentType = "text/plain";
context.Response.Write(json);
context.Response.End();
}

context.Response.ContentType = "text/plain";
context.Response.Write("");
context.Response.End();
}

public bool IsReusable
{
    get { return false; }
}
}
}

```

در اینجا نحوه مدیریت سمت سرور `auto save` و همچنین ارسال تصاویر ادیتور Froala، ذکر شده‌اند. با استفاده از `context.Request.Form` می‌توان به عناصر ارسالی به سرور دسترسی پیدا کرد. همچنین توسط `context.Request.Files`، اگر فایلی ارسال شده بود، ذخیره شده و نهایتا خروجی JSON مدنظر بازگشت داده می‌شود.

یک نکته‌ی امنیتی مهم

```

<location path="upload">
  <system.webServer>
    <handlers accessPolicy="Read" />
  </system.webServer>
</location>

```

تنظیم فوق را در `web.config` سایت، جهت `Read only` کردن پوشه‌ی ارسال تصاویر، حتماً مدنظر داشته باشد. در اینجا فرض شده‌است که پوشه‌ی `uploads` قرار است قابلیت اجرای فایل‌های پویا را نداشته باشد.

کدهای کامل این مطلب را در ادامه می‌توانید دریافت کنید

[Froala-Sample](#)

نظرات خوانندگان

نویسنده: علی رضایی
تاریخ: ۱۳۹۳/۰۲/۱۸ ۱۸:۱۱

سلام
ممnon برای این مطلب
فکر کنم این ادیتور با کروم یک مشکل دارد:
برای مثال کلمه «تسنّت» رو در اون تایپ کنید، سپس بعد از «س» یک حرف جدید مثل «ن» وارد کنید: نتیجه باید بشود «تسنّت»
اما میشود «تسنّت». البته من در کروم 34 تسنّت کردم. با IE 10 و Firefox 28 مشکلی نداشت.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۱۸ ۱۸:۵۷

مشکلات این ادیتور را لطفا در [issue tracker آنها](#) ارسال کنید.

نویسنده: محسن
تاریخ: ۱۳۹۳/۰۲/۲۰ ۸:۳۰

عالیه؛ فقط قسمت آپلود فایل را کم داره.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۲۰ ۸:۵۶

[به زودی](#).

نویسنده: میثم مهریانی
تاریخ: ۱۳۹۳/۰۲/۲۱ ۸:۵۴

برای Upload تصویر به صورتی که در Sample خود ادیتور با Progress انجام می‌شود چه باید کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۲۱ ۹:۹

سمت کلاینت آن به صورت پیش فرض فعال است و نیاز به تنظیم اضافه‌تری ندارد. فقط باید مدیریت سمت سرور آن را مطابق کدهایی که در مطلب جاری توضیح داده شدند، انجام دهید.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۳/۰۳/۰۹ ۱۱:۱۵

با سلام . برای ایجاد بلاک کد که در redactor موجود است، در این ادیتور چه باید کرد؟



نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۰۹ ۱۴:۳

امکان [افزونه نویسی](#) هم دارد. اگر همان کدهای افزونه‌ای را که برای redactor نوشتم، با این ادیتور تطابق دهیم، به دکمه‌ی سفارشی زیر خواهیم رسید:

```

buttons: [
    // ....,
    "insertHTML" //custom button
],

customButtons: {
    insertHTML: {
        title: 'Insert Code',
        icon: {
            type: 'font',
            value: 'fa fa-dollar' // Font Awesome icon class fa fa-*
        },
        callback: function (editor) {
            editor.saveSelection();

            var codeModal = $("<div>").addClass("froala-modal").appendTo("body");
            var wrapper = $("<div>").addClass("f-modal-wrapper").appendTo(codeModal);
            $("<h4>").append('<span data-text="true">Insert Code</span>')
                .append($('<i class="fa fa-times" title="Cancel">'))
                .click(function () {
                    codeModal.remove();
                })
                .appendTo(wrapper);

            var dialog = "<textarea id='code_area' style='height: 211px; width: 538px;'><br/><label>Language:</label><select id='code_lang'><option>CSharp</option><option>VB</option><option>JScript</option><option>Sql</option><option>XML</option><option>CSS</option><option>Java</option><option>Delphi</option></select> <input type='button' name='insert' id='insert_btn' value='Insert' /><br/>";
            $(dialog).appendTo(wrapper);

            $("#code_area").text(editor.text());

            if (!editor.selectionInEditor()) {
                editor.$element.focus();
            }

            $('#insert_btn').click(function () {
                var lang = $("#code_lang").val();
                var code = $("#code_area").val();
                code = code.replace(/\s+$/, ""); // rtrim
                code = $(<span>).text(code).html(); // encode

                var htmlCode = "<pre language='" + lang + "' name='code'>" + code +
                "</pre></div>";
                var codeBlock = "<div align='left' dir='ltr'>" + htmlCode +
                "</div><br/>";

                editor.restoreSelection();
                editor.insertHTML(codeBlock);
                editor.saveUndoStep();

                codeModal.remove();
            });
        }
    }
}

```

نوبنده: احسان میرسعیدی
تاریخ: ۱۱:۱۳ ۱۳۹۳/۰۳/۱۱

آقای نصیری با تشکر از معرفی این ویرایش گر و ارائه این تکه کد. من این کد رو در گیت هاب دیدم که به اشتراک گذاشتید. به نظرم مشکلی با SyntaxHighlighter داشت که در [اینجا](#) مطرح کردم. با تغییری که اعمال کردم در وبلاگ من کار کرد و فوق العاده بود. با تشکر از به اشتراک گذاری دانش تون

نوبنده: ایلیا اکبری فرد
تاریخ: ۱۱:۲۷ ۱۳۹۳/۰۴/۰۱

قطعه کد برای `fullscreen` این ادیتور:

```
fullscreen: {
    title: 'FullScreen',
    icon: {
        type: 'font',
        value: 'fa fa-external-link'
    },
    callback: function (editor) {
        if (editor.$box.hasClass("froala-editor-full-screen")) {
            editor.$box.removeClass("froala-editor-full-screen");
            editor.$box.appendTo(editor.editorParent);
        }
        else {
            if (editor.editorParent === undefined)
                editor.editorParent = editor.$box.parent();
            editor.$box.addClass("froala-editor-full-screen");
            editor.$box.appendTo("body");
        }
        editor.focus();
    },
},
```

و کلاس css مربوطه :

```
.froala-editor-full-screen {
    position: fixed !important;
    z-index: 10000;
    top: 0;
    bottom: 0;
    left: 0;
    right: 0;
    background-color: #fff;
}

.froala-editor-full-screen div.f-placeholder {
    height: 100% !important;
}
```

نویسنده: حمید رضا رحیمی
تاریخ: ۱۳۹۳/۰۴/۲۴

سلام و ممنون از انتشار مطالب مفیدتون
یک سوال : بنده (بنا به سفارش مشتری) احتیاج به ادیتوری دارم که در ثبت لینک امکان flow و noflow رو داشته باشه
تا به حال خودم این امکان رو ندیدم. شما موردی رو مشاهده کردید و یا میشه همین ادیتور رو دستکاری کرد ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۲۴

این قابلیت را به صورت توکار دارد. [اطلاعات بیشتر](#)

نویسنده: حمید رضا رحیمی
تاریخ: ۱۳۹۳/۰۴/۲۴

ممنون از پاسخ تان اما این برای کلیه لینک‌ها می‌باشد من می‌خوام موردی به یک لینک این امکان داده شده و به یکی دیگه داده
.Enables or disables adding the rel="nofollow" attribute on all links نشده

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۲۴

در قسمت مدیریت پروژه آن درخواست دهید تا یک checkbox مخصوص اینکار به صفحه دیالوگ استاندارد افزودن لینک آن
اضافه کنند.

نویسنده: رضایی

تاریخ: ۱۳۹۳/۰۵/۲۷ ۲:۳۰

با سلام؛ توسط خاصیت `imageUploadURL` مشخص میکنیم که تصاویر کجا ذخیره شوند. یک خاصیت دیگه هم در `imagesLoadURL` است که تصاویر کجا ذخیره شوند. خصوص این خاصیت میشه توضیح بدین.

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۵/۲۷ ۹:۲

برای فعال سازی `media manager` آن کاربرد دارد. [یک افزونه](#) است که باید جداگانه اضافه شود. مقدار `imagesLoadUrl` به مسیری در سرور اشاره میکند که قرار است آرایه‌ای از مسیرهای تصاویر قابل نمایش در آن را به فرمت JSON بازگشت دهد.

نویسنده: رضایی

تاریخ: ۱۳۹۳/۰۵/۲۷ ۲۲:۵۵

برای حذف یک تصویر از قسمت منیجر از خاصیت `imageDeleteParams` و برای ارسال پارامترها از `imageDeleteURL` میکنیم. خاصیت `imageDeleteParams` را باید به چه صورت مقداردهی کرد موقعي که از کد زير استفاده ميکنم اديتور لود نميشه

```
imageDeleteParams: { src: $img.attr('src') }
```

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۵/۲۸ ۰:۷

- نیازی به تنظیم `imageDeleteParams` نیست. از آن برای ارسال یک سری اطلاعات اضافی و دستی مانند شماره `id` کاربر و امثال آن استفاده میشود. آن استفاده `src` تصویری که قرار است حذف شود، به صورت خودکار به مسیر مشخص شده توسط `imageDeleteURL` در سمت سرور ارسال میشود.

- برای تشخیص دقیق نام پارامترهای ارسالی به سرور از برگه‌ی `network` [فایرباگ](#) استفاده کنید.
- روش دوم تشخیص پارامترها و مقادیر ارسالی به سرور استفاده از [FormCollection](#) است.

نویسنده: رضایی

تاریخ: ۱۳۹۳/۰۵/۲۸ ۱۵:۱۵

ممnon؛ پارامتر `src` که آدرس تصویر رو در خودش داره به سمت سرور ارسال میشه.

نویسنده: رضایی

تاریخ: ۱۳۹۳/۰۵/۲۸ ۲۲:۱۲

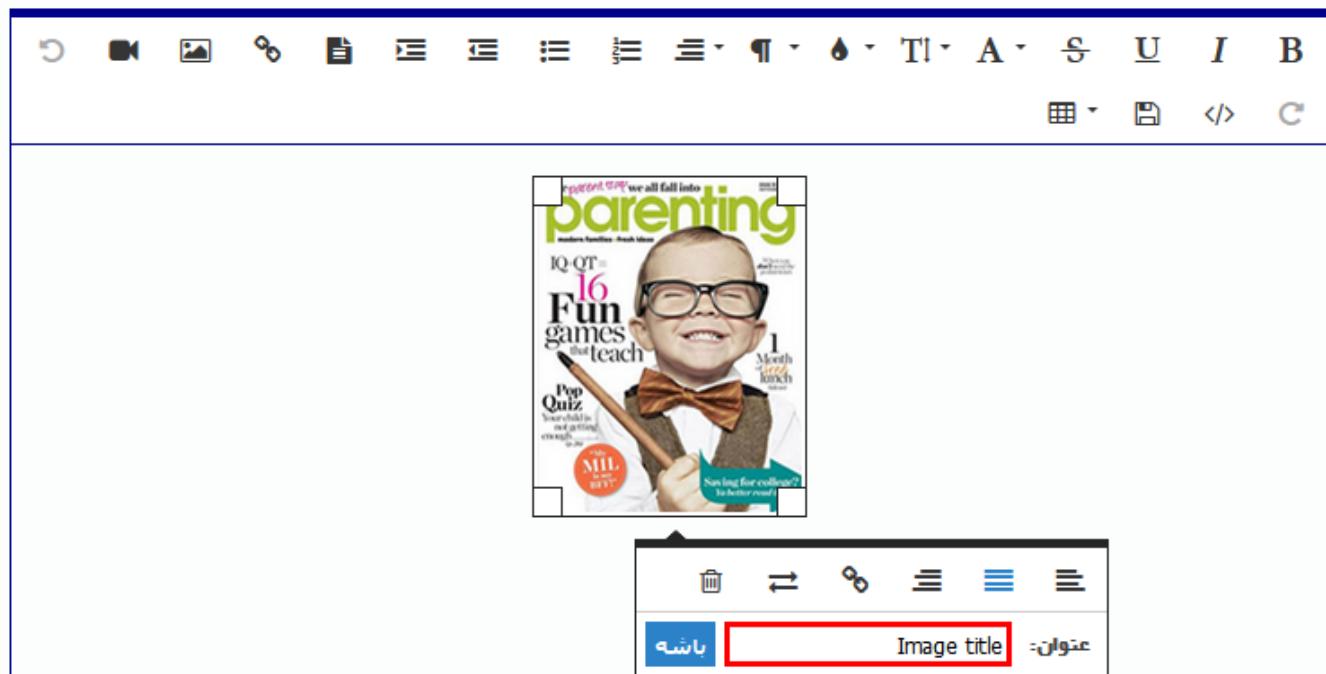
با سلام

موقعي که از اديتور توی یک `modal` استفاده ميکنم تکست باکس‌های اديتور غیر قابل فوکوس میشن `z-index` رو هم تغییر دادم اما تاثیری نداشت و موقع کلیک ، فوکوس به تکست باکس‌های دیگه که روی `modal` هستند منتقل میشه

عنوان پست

آنکار ۴

متن پست



نویسنده: وحید نصیری
تاریخ: ۲۲:۵۷ ۱۳۹۳/۰۵/۲۸

[اینجا گزارش شده](#). در حال بررسی است.

نویسنده: Mohammadreza AAA
تاریخ: ۱۲:۵ ۱۳۹۳/۰۶/۰۲

سلام.

من این ادیتورو گذاشتم. فوق العاده راحت و قشنگه. فقط با بخش ارسال عکس مشکل دارم. مسئله اینکه عکس من توی پوشه مورد نظرم داره ذخیره میشه ولی اشکالش اینکه توی متنه موقع نمایش هیچی نمیاید. همه خصوصیات مثلای ایتالیک و بولد به متون اعمال میشن ولی عکس نیست!

نویسنده: وحید نصیری
تاریخ: ۱۲:۱۱ ۱۳۹۳/۰۶/۰۲

[بررسی کنید چه خطای در فایرباگ گزارش شده](#)

نویسنده: M.Shakeri
تاریخ: ۸:۵۷ ۱۳۹۳/۰۶/۰۴

سلام

اگه داخل ادیتور متنه نوشته نشه مقدار پیش فرضش

"<p>
</p>"

هست که موقع اعتبار سنجی معتبره.
تو تنظیماتش هم گزینه‌ای برای خالی کردن ادیتور وجود نداره.

نویسنده: وحید نصیری
تاریخ: ۱۲:۵۰ ۱۳۹۳/۰۶/۰۵

«[اعتبار سنجی سمت کاربر wysiwyg-editor در ASP.NET MVC](#)

نویسنده: وحید نصیری
تاریخ: ۱۳:۲۴ ۱۳۹۳/۰۶/۱۷

امکان آپلود فایل هم به آن اضافه شده: [^](#) و [^](#)

نویسنده: هادی دانش پژوه
تاریخ: ۹:۴۳ ۱۳۹۳/۰۶/۱۸

با سلام
مثل اینکه این ادیتور دیگه رایگان نیست :)
می خواستم بدونم اگه از همان سورسی که برای دانلود گذاشتیند استفاده کنم مشکلی بوجود نمیاد؟

نویسنده: وحید نصیری
تاریخ: ۱۰:۱۶ ۱۳۹۳/۰۶/۱۸

این مسایل مشکل ما نیست، چون فلان ابزار رایگان هست یا نیست، دلیلی بر این نیست و نخواهد بود که ما در مورد آن مطلبی تهیه نکنیم یا کاری انجام ندهیم.

نویسنده: مهرداد
تاریخ: ۱۷:۱۹ ۱۳۹۳/۰۷/۰۸

سلام؛ چطوری می‌توان به این ویرایشگر قابلیت `Code` اضافه کرد آیا شما قبلاً تجربه همچین کاری را دارید و یا حتی جایی دیده‌اید؟

نویسنده: وحید نصیری
تاریخ: ۱۷:۲۹ ۱۳۹۳/۰۷/۰۸

[کمی بالاتر](#) توضیح دادم.

نویسنده: اصغر
تاریخ: ۱۱:۱۹ ۱۳۹۳/۰۷/۲۵

سلام. من از این قطعه کدی که شما گذاشتید برای ایجاد بلاک کد استفاده می‌کنم و درست هم کار می‌کنه منتها یک مشکلی که دارم این هستش که وقتی فرم رو پست می‌کنم تمام کد رو فقط در یک خط نشون میده. درحالی که اگر برم توی دیتابیسم و داخل تگ `pre` کدم رو پیست کنم درست نشون میده. اگر نکته‌ای داره ممنون می‌شم بهم بگید. تو اینترنت هم خیلی گشتم ولی چیز خاصی پیدا نکردم.

نویسنده: وحید نصیری
تاریخ: ۱۱:۴۳ ۱۳۹۳/۰۷/۲۵

مشکل CSS دارید. سطر:

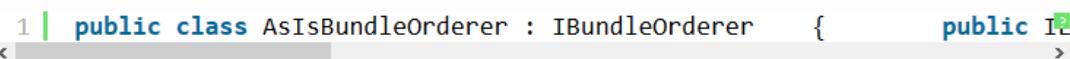
```
var htmlCode = "<pre language='" + lang + "' name='code'>" + code + "</pre></div>";
```

مطابق تنظیمات شخصی است. اینجا class را بر اساس تنظیمات CSS نمایشگر کدهای خود اضافه کنید (مثلًا 'brush' (csharp)).

```
var htmlCode = "<pre class='brush: " + lang + "' language='" + lang + "' name='code'>" + code + "</pre></div>;"
```

نویسنده: اصغر
تاریخ: ۱۲:۱۱ ۱۳۹۳/۰۷/۲۵

بله من قبلاً این کار رو کردم تا استایل نمایش کد درست شد ولی الان مشکلی که هست فقط اینه که چندین خط کدی که پیست میکنم همچنان توی یک خط نمایش داده میشه و اسکرول افقی میخوره. مثل شکل زیر:



نویسنده: وحید نصیری
تاریخ: ۱۲:۱۸ ۱۳۹۳/۰۷/۲۵

مشکل این ادیتور یا افزونه‌ی نوشته شده نیست. اگر از یک فشرده کننده HTML استفاده می‌کنید، ممکن است خطوط جدید را حذف کرده باشد. برای مثال این مورد در « بلاگر » گوگل مرسوم است. راه حل آن:

```
SyntaxHighlighter.config.bloggerMode = true;
SyntaxHighlighter.all();
```

[برای مطالعه بیشتر](#)

نویسنده: رضا ریاضی
تاریخ: ۱۹:۲۳ ۱۳۹۳/۱۰/۱۴

سلام؛ در زمان استفاده از Ajax.BeginForm در صورتی که از این ادیتور هم استفاده کنیم عمل POST انجام نمی‌شود. به چه دلیل؟ در شرایطی که با Html.BeginForm چنین مشکلی وجود ندارد.

نویسنده: وحید نصیری
تاریخ: ۲۱:۵۱ ۱۳۹۳/۱۰/۱۴

در مثالی که پیوست شد، از فایل jquery.unobtrusive-ajax.js استفاده نشده‌است. ابتدا این فایل را اضافه کنید:

```
PM> Install-Package Microsoft.jQuery.Unobtrusive.Ajax
```

بعد مدخل تعریف آن هم باید به فایل _Layout.cshtml اضافه شود:

```
<script src="~/Scripts/jquery-1.10.2.min.js"></script>
<script src="~/Scripts/jquery.validate.min.js"></script>
<script src="~/Scripts/jquery.unobtrusive-ajax.min.js"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>
```

به این ترتیب Ajax.BeginForm کار خواهد کرد (آزمایش شد).

نویسنده: رضا منصوری
تاریخ: ۱۳۹۳/۱۱/۲۳ ۱۶:۰

بر اساس [این](#) باید از this editor به جای استفاده کنیم.
من به این صورت استفاده کردم

```
insertHTML: {
    title: 'Insert Code',
    icon: {
        type: 'font',
        value: 'fa fa-dollar' // Font Awesome icon class fa fa-*
    },
    callback: function (editor) {
        this.saveSelection();
        var thisEditor = this;
        var codeModal = $("<div>").addClass("froala-modal").appendTo("body");
        var wrapper = $("<div>").addClass("f-modal-wrapper").appendTo(codeModal);
        $("<h4>").append('<span data-text="true">Insert Code</span>')
            .append('<i class="fa fa-times" title="Cancel">')
            .click(function () {
                codeModal.remove();
            })
            .appendTo(wrapper);

        var dialog = "<textarea id='code_area' style='height: 211px; width: 538px;' /><label>Language:</label><select id='code_lang'><option>CSharp</option><option>VB</option><option>JScript</option><option>Sql</option><option>XML</option><option>CSS</option><option>Java</option><option>Delphi</option></select> <input type='button' name='insert' id='insert_btn' value='Insert' />";
        $(dialog).appendTo(wrapper);

        $("#code_area").text(this.text());
        if (!this.selectionInEditor()) {
            this.$element.focus();
        }

        $('#insert_btn').click(function () {
            var lang = $("#code_lang").val();
            var code = $("#code_area").val();
            code = code.replace(/\s+$/ , ""); // rtrim
            code = $('<span/>').text(code).html(); // encode
            var htmlCode = "<pre class='brush: " + lang.toLowerCase() + " language=" + lang + "' name='code'>" + code + "</pre></div>"; // syntaxhighlighter
            //var htmlCode = "<pre language='" + lang + "' name='code'>" + code + "</pre>";
            var codeBlock = "<div align='left' dir='ltr'>" + htmlCode +
                "</div>";

            thisEditor.restoreSelection();
            thisEditor.insertHTML(codeBlock);
            thisEditor.saveUndoStep();

            codeModal.remove();
        });
    }
}
```

سازی: نویسنده: ۱۳۹۳/۱۲/۲۵ ۱۶:۳۰

اما این کدی که به عنوان custombutton گذاشته شده رو توی کدوم فایل و به چه صورت و به کجا فایلی که معرفی می فرماید باید وارد کرد؟
اجازه بفرمایید تصحیح کنم ،
این کد را مطابق دستور شما اضافه کردم اما زمانی کلیک میکنم هیچ اتفاقی نمیافته

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۲۵ ۱۶:۳۸

- سورس انتهای بحث را دریافت کنید. به خاصیت buttons آن (که در View برنامه موجود است)، مقدار insertHTML اضافه می‌شود. همچنین یک خاصیت جدید هم به نام customButtons با مقداری که مشخص شده، باید در کنار سایر خواص این افزونه، اضافه شود.
- سایر نظرات تکمیلی و نگارش‌های بهبود یافته را هم مد نظر داشته باشید.

نویسنده: سانی
تاریخ: ۱۳۹۳/۱۲/۲۵ ۱۷:۵۳

از راهنمایی شما بسیار متشکرم. دو مورد برای من پیش اومد؛ یکی اینکه با استفاده از راهنمایی‌های شما دکمه کد اضافه شد، اما اول اینکه زمانی روی آن کلیک می‌کنم هیچ اتفاقی نمیافته. با دو browser چک کردم و هیچ تغییری نکرد. مطلب دوم: در local آیکن‌ها نمایش داده می‌شوند اما روی هاست آیکن‌ها نمایش داده نمی‌شوند. در ابتدا از bundle استفاده می‌کردم اما با اینکه لینک مستقیم فایل‌ها را هم گذاشتیم باز همان مشکل را داشتم. البته فایل اصلی jquery را هم به ابتدای صفحه انتقال دادم، اما باز هم تغییری نکرد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۲۵ ۲۳:۱۴

- کدهای این مطلب به مخزن کد [Froala-Sample](#) منتقل شدند.
- نحوه‌ی تعریف دکمه‌ی کد [در اینجا](#)
- آیکن‌ها از [پوشه‌ی fonts](#) قرار گرفته در ریشه‌ی سایت خوانده می‌شوند. اگر سرور قادر به نمایش آن‌ها نیست به احتمال زیاد آن‌ها را تعریف نکرده‌است ([^](#) و [_](#) و [^](#)) .

```
<system.webServer>
  ...
<staticContent>
  <remove fileExtension=".woff" />
  <mimeMap fileExtension=".woff" mimeType="application/font-woff" />
  <mimeMap fileExtension=".eot" mimeType="application/vnd.ms-fontobject" />
  <mimeMap fileExtension=".otf" mimeType="font/otf" />
  <mimeMap fileExtension=".svg" mimeType="images/svg+xml" />
  <mimeMap fileExtension=".svgz" mimeType="images/svg+xml" />
</staticContent>
</system.webServer>
```

نویسنده: احمد نواصری
تاریخ: ۰۱/۱۳/۱۳۹۴ ۰:۳۷

سلام. من در حال نوشتمن یک سیستم CMS هستم و برای نوشتمن مطلب از این ادیتور استفاده می‌کنم. وقتی مطلب رو ارسال می‌کنم و می‌خواهم در قسمت مطالب نشون بدم، محتوای نوشته به صورت کد HTML نشون داده می‌شده، می‌خواستم بدونم چطور می‌تونم این مشکل رو برطرف کنم؟

نویسنده: وحید نصیری
تاریخ: ۰۱/۱۳/۱۳۹۴ ۱:۴۱

باید از متد [@Html.Raw](#) برای جلوگیری از encoding خودکار razor استفاده کنید.

نویسنده: احمد نواصری
تاریخ: ۰۱/۱۳/۱۳۹۴ ۱۱:۲۰

آیا استفاده از این متد، مشکلی از نظر امنیتی ایجاد نخواهد کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۱/۱۳ ۱۱:۴۲

ورودی کاربر را همیشه باید [پردازش و تمیز کنید](#) و یا از [هدرهای Content Security Policy](#) استفاده کنید.

نویسنده: احمد نواصری
تاریخ: ۱۳۹۴/۰۳/۰۲ ۱۷:۴۶

سلام؛ اگه کنترلر توی یک Area باشه چطوری میتونیم مسیر عکس‌های ذخیره شده تو ریشه سایت رو به فرمت JSON برگردانیم؟ مثلاً ما یک Area داریم به نام Admin و کنترلر Post هم تو این Area هست. من وقتی عکس رو آپلود میکنم چیزی نمایش داده نمیشه رفتم تو حالت Developer مرورگر دیدم همچین خطایی میده :

```
http://localhost:1455/Admin/Post/Content/Images/a_sunny_days_end-wallpaper-1440x900.jpg 404 (Not Found)
```

دلیلش هم اینه که تو متد Froala aUploadImage مقدار JSON برگشتی به صورت زیر هست :

```
return Json(new { link = "Content/Images/" + fileName }, JsonRequestBehavior.AllowGet);
```

که اشاره به مسیر بالا داره. من میخوام خروجی JSON به صورت زیر باشه

```
http://localhost:1455/Content/Images/a_sunny_days_end-wallpaper-1440x900.jpg
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۳/۰۲ ۲۰:۱۰

از این روش برای تولید url استفاده کنید:

```
new UrlHelper(this.Request.RequestContext).Content("~/content/images/file.png")
```

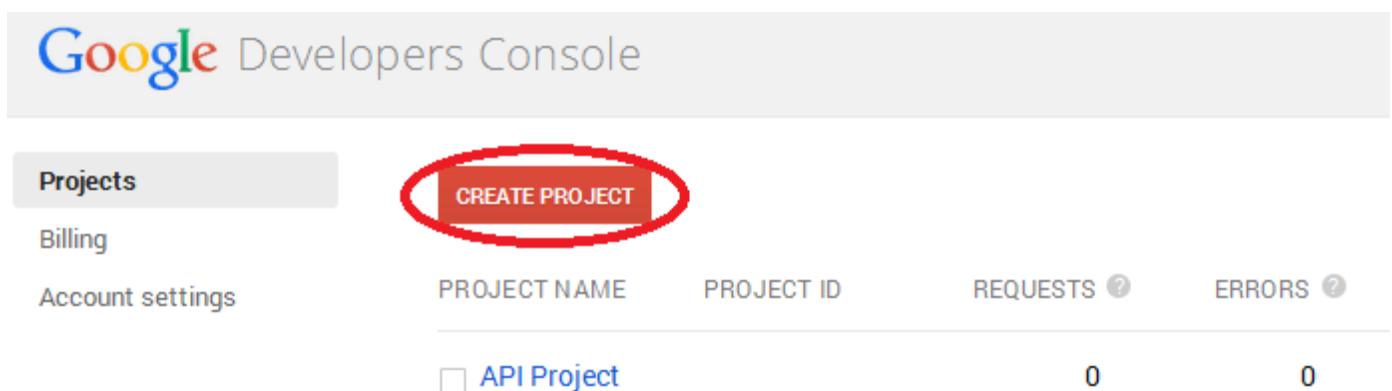
در زمان نگارش این مطلب، آخرین نگارش API مخصوص Google Analytics، نگارش سوم آن است و ... کار کردن با آن دارای مراحل خاصی است که حتما باید رعایت شوند. در غیر اینصورت عملکرد یک برنامه‌ی وب یا سرویس ویندوز قابل اجرا نخواهد بود. زیرا در حالت متدالوک کار با API مخصوص Google Analytics، ابتدا یک صفحه‌ی لاگین به Gmail باز می‌شود که باید به صورت اجباری، مراحل آن را انجام داد تا مشخصات تأیید شده‌ی اکانت در حال استفاده‌ی از API، در پوششی AppData ویندوز برای استفاده‌های بعدی ذخیره شود. این مورد برای یک برنامه‌ی دسکتاب معمولی مشکل ساز نیست؛ زیرا کاربر برنامه، به سادگی می‌تواند صفحه‌ی مرورگری را که باز شده‌است، دنبال کرده و به اکانت گوگل خود وارد شود. اما این مراحل را نمی‌توان در یک برنامه‌ی وب یا سرویس ویندوز پیگیری کرد، زیرا عموماً امکان لاگین از راه دور به سرور و مدیریت صفحه‌ی لاگین به Gmail وجود ندارد یا بهتر است عنوان شود، بی‌معنا است. برای حل این مشکل، گوگل راه حل دیگری را تحت عنوان اکانت‌های سرویس، ارائه داده است که پس از ایجاد آن، یک فایل Certificate X509 برای اعتبارسنجی سرویس، در اختیار برنامه نویس قرار می‌گیرد تا بدون نیاز به لاگین دستی به Gmail، بتواند از API گوگل استفاده کند. در ادامه نحوه‌ی فعال سازی این قابلیت و استفاده از آن را بررسی خواهیم کرد.

ثبت برنامه‌ی خود در گوگل و انجام تنظیمات آن

اولین کاری که برای استفاده از نگارش سوم Google Analytics API باید صورت گیرد، ثبت برنامه‌ی خود در Google Developer Console است. برای این منظور ابتدا به آدرس ذیل وارد شوید:

<https://console.developers.google.com>

سپس بر روی دکمه‌ی Create project کلیک کنید. نام دلخواهی را وارد کرده و در ادامه بر روی دکمه‌ی Create کلیک نمائید تا پروفایل این پروژه ایجاد شود.



تنها نکته‌ی مهم این قسمت، بخاطر سپردن نام پروژه است. زیرا از آن جهت اتصال به API گوگل استفاده خواهد شد. پس از ایجاد پروژه، به صفحه‌ی آن وارد شوید و از منوی سمت چپ صفحه، گزینه‌ی Credentials را انتخاب کنید. در ادامه در صفحه‌ی باز شده، بر روی دکمه‌ی Create new client id کلیک نمائید.

The screenshot shows the Google Developers Console interface. At the top, it says "Google Developers Console". Below that, there's a navigation bar with "My Project" and "OAuth". The main area has a sidebar with links: "Overview", "APIs & auth", "APIs", "Credentials" (which is highlighted with a red oval), "Consent screen", and "Push". To the right of the sidebar, under "OAuth", there's a description of OAuth 2.0 and a "Learn more" link. At the bottom right of this section is a red button labeled "CREATE NEW CLIENT ID" which is also circled in red.

در صفحه‌ی باز شده، گزینه‌ی Service account را انتخاب کنید. اگر سایر گزینه‌ها را انتخاب نماید، کاربری که قرار است از API استفاده کند، باید بتواند توسط مرورگر نصب شده‌ی بر روی کامپیوتر اجرا کننده‌ی برنامه، یکبار به گوگل لاگین نماید که این مورد مطلوب برنامه‌های وب و همچنین سرویس‌ها نیست.

Create Client ID

APPLICATION TYPE

Web application

Accessed by web browsers over a network.

Service account

Calls Google APIs on behalf of your application instead of an end-user. [Learn more](#)

Installed application

Runs on a desktop computer or handheld device (like Android or iPhone).

[Create Client ID](#)

[Cancel](#)

در اینجا ابتدا یک فایل مجوز JSON را به صورت خودکار دریافت خواهید کرد و همچنین پس از ایجاد Client ID، نیاز است، ایمیل آن را جایی یادداشت نماید:

OAuth

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private.
[Learn more](#)

[CREATE NEW CLIENT ID](#)

Compute Engine and App Engine [Learn more](#)

Client ID	129908.apps.googleusercontent.com
Email address	129908@developer.gserviceaccount.com
Download JSON	

Service Account

Client ID	107-908-	mq.apps.googleusercontent.com
	ti1vi55flpnkp3	
Email address	10-908-	mq@developer.gserviceaccount.com
	ti1vi55flpnkp3o	
Public key fingerprints	811e1d9bfd8ed	3b976cd516b55

[Generate new key](#)

[Download JSON](#)

[Delete](#)

از این ایمیل و همچنین فایل JSON ارائه شده، جهت لاینین به سرور استفاده خواهد شد.
همچنین نیاز است تا به برگه‌ی APIs پروژه‌ی ایجاد شده رجوع کرد و گزینه‌ی Analytics API آن را فعال نمود:

The screenshot shows the 'My Project' interface in the Google Cloud Platform. On the left, there's a sidebar with 'Overview', 'APIs & auth', 'APIs' (which is highlighted with a red circle), and 'Credentials'. The main area lists APIs with columns for 'NAME', 'QUOTA', and 'STATUS'. The 'Analytics API' is listed with '0%' quota and an 'ON' status button, which is also circled in red.

NAME	QUOTA	STATUS
Analytics API	0%	ON
BigQuery API	0%	ON
Google Cloud SQL		ON

تا اینجا کار ثبت و فعال سازی برنامه‌ی خود در گوگل به پایان می‌رسد.

دادن دسترسی به Client ID ثبت شده در برنامه‌ی Google Analytics

پس از اینکه Client ID سرویس خود را ثبت کردید، نیاز است به اکانت Google Analytics خود وارد شوید. سپس در منوی آن، گزینه‌ی Admin را پیدا کرده و به آن قسمت، وارد شوید:

The screenshot shows the Google Analytics Admin interface. The top navigation bar includes 'Home', 'Reporting', 'Customization', and 'Admin' (which is highlighted with a red circle). Below the navigation, there's a sidebar with 'ACCOUNT' set to 'www.dotnettips.info'. Under the account settings, there are links for 'Account Settings', 'User Management' (highlighted with a red circle), 'AdSense Linking', 'All Filters', and 'Change History'.

در ادامه به گزینه‌ی User management آن وارد شده و به ایمیل Client ID ایجاد شده در قسمت قبل، دسترسی خواندن و آنالیز را اعطاء کنید:

The screenshot shows the 'User Management' section of the Google Analytics Admin interface. On the left, there's a list of users: '1. 107 5flp vi5 kebn g4j3mq@developer.gserviceaccount.com' (with 'User Management' highlighted with a red circle) and '2.'. On the right, there's a table titled 'Account Permissions' with columns for 'Email' and 'Account Permissions'. The first user has a dropdown menu open, with 'Read & Analyze' selected (highlighted with a red circle). Below the table, there's a 'Manage Users, Edit, Collaborate, Read & Analyze' button and a 'Show row' link. At the bottom, there's a modal for adding new permissions, with fields for 'Add permissions for:' (an input field with placeholder 'User e-mail that is registered in Google accounts'), 'Read & Analyze' (a dropdown menu), 'Notify this user by email' (a checkbox), and 'Add' and 'Cancel' buttons (the 'Add' button is highlighted with a red circle).

در صورت عدم رعایت این مساله، کلاینت API، قادر به دسترسی به Google Analytics نخواهد بود.

استفاده از نگارش سوم Google Analytics API در دات نت

قسمت مهم کار، تنظیمات فوق است که در صورت عدم رعایت آن‌ها، شاید نصف روزی را مشغول به دیباگ برنامه شوید. در ادامه نیاز است پیشنازهای دسترسی به نگارش سوم Google Analytics API را نصب کنیم. برای این منظور، سه بسته‌ی نیوگت ذیل را توسط کنسول پاورشل نیوگت، به برنامه اضافه کنید:

```
PM> Install-Package Google.Apis
PM> Install-Package Google.Apis.auth
PM> Install-Package Google.Apis.Analytics.v3
```

پس از نصب، کلاس GoogleAnalyticsApiV3 زیر، جزئیات دسترسی به Google Analytics API را کپسوله می‌کند:

```
using System;
using System.Linq;
using System.Security.Cryptography.X509Certificates;
using Google.Apis.Analytics.v3;
using Google.Apis.Analytics.v3.Data;
using Google.Apis.Auth.OAuth2;
using Google.Apis.Services;

namespace GoogleAnalyticsAPIv3Tests
{
    public class AnalyticsQueryParameters
    {
        public DateTime Start { set; get; }
        public DateTime End { set; get; }
        public string Dimensions { set; get; }
        public string Filters { set; get; }
        public string Metrics { set; get; }
    }

    public class AnalyticsAuthentication
    {
        public Uri SiteUrl { set; get; }
        public string ApplicationName { set; get; }
        public string ServiceAccountEmail { set; get; }
        public string KeyFilePath { set; get; }
        public string KeyFilePathPassword { set; get; }

        public AnalyticsAuthentication()
        {
            KeyFilePathPassword = "notasecret";
        }
    }

    public class GoogleAnalyticsApiV3
    {
        public AnalyticsAuthentication Authentication { set; get; }
        public AnalyticsQueryParameters QueryParameters { set; get; }

        public GaData GetData()
        {
            var service = createAnalyticsService();
            var profile = getProfile(service);
            var query = service.Data.Ga.Get("ga:" + profile.Id,
                QueryParameters.Start.ToString("yyyy-MM-dd"),
                QueryParameters.End.ToString("yyyy-MM-dd"),
                QueryParameters.Metrics);
            query.Dimensions = QueryParameters.Dimensions;
            query.Filters = QueryParameters.Filters;
            query.SamplingLevel = DataResource.GaResource.GetRequest.SamplingLevelEnum.HIGHERPRECISION;
            return query.Execute();
        }

        private AnalyticsService createAnalyticsService()
        {
            var certificate = new X509Certificate2(Authentication.KeyFilePath,
```

```
Authentication.KeyFilePassword, X509KeyStorageFlags.Exportable);
    var credential = new ServiceAccountCredential(
        new ServiceAccountCredential.Initializer(Authentication.ServiceAccountEmail)
    {
        Scopes = new[] { AnalyticsService.Scope.AnalyticsReadonly }
    }.FromCertificate(certificate));

    return new AnalyticsService(new BaseClientService.Initializer
    {
        HttpClientInitializer = credential,
        ApplicationName = Authentication.ApplicationName
    });
}

private Profile getProfile(AnalyticsService service)
{
    var accountListRequest = service.Management.Accounts.List();
    var accountList = accountListRequest.Execute();
    var site = Authentication.SiteUrl.Host.ToLowerInvariant();
    var account = accountList.Items.FirstOrDefault(x =>
x.Name.ToLowerInvariant().Contains(site));
        var webPropertyListRequest = service.Management.Webproperties.List(account.Id);
        var webPropertyList = webPropertyListRequest.Execute();
        var sitePropertyList = webPropertyList.Items.FirstOrDefault(a =>
a.Name.ToLowerInvariant().Contains(site));
            var profileListRequest = service.Management.Profiles.List(account.Id, sitePropertyList.Id);
            var profileList = profileListRequest.Execute();
            return profileList.Items.FirstOrDefault(a => a.Name.ToLowerInvariant().Contains(site));
    }
}
```

در اینجا در ابتدا بر اساس فایل 12m ایی که از گوگل دریافت شد، یک Certificate2X509 ایجاد می‌شود. پسورد این فایل مساوی است با ثابت notasecret که در همان زمان تولید اکانت سرویس در گوگل، لحظه‌ای در صفحه نمایش داده خواهد شد. به کمک آن و همچنین ServiceAccountEmail ایمیلی که پیشتر به آن اشاره شد، می‌توان به AnalyticsService لاگین کرد. به این ترتیب به صورت خودکار می‌توان شماره پروفایل اکانت سایت خود را یافت و از آن در حین فراخوانی service.Data.Ga.Get استفاده کرد.

مثال، از نحوه استفاده از کلاس GoogleAnalyticsApiV3

در ادامه یک برنامه، کنسرسیوم GoogleAnalyticsApiV3 را ملاحظه می‌کنید که از کلاس استفاده می‌کند:

```
using System;
using System.Collections.Generic;
using System.Linq;
namespace GoogleAnalyticsAPIv3Tests
{
    class Program
    {
        static void Main(string[] args)
        {
            var statistics = new GoogleAnalyticsApiV3
            {
                Authentication = new AnalyticsAuthentication
                {
                    ApplicationName = "My Project",
                    KeyFilePath = "811e1d9976cd516b55-privatekey.p12",
                    ServiceAccountEmail = "10152bng4j3mq@developer.gserviceaccount.com",
                    SiteUrl = new Uri("http://www.dotnettips.info/")
                },
                QueryParameters = new AnalyticsQueryParameters
                {
                    Start = DateTime.Now.AddDays(-7),
                    End = DateTime.Now,
                    Dimensions = "ga:date",
                    Filters = null,
                    Metrics = "ga:users,ga:sessions,ga:pageviews"
                }
            }.GetData();
        }
    }
}
```

```
        foreach (var result in statistics.TotalsForAllResults)
    {
        Console.WriteLine(result.Key + " -> total:" + result.Value);
    }
    Console.WriteLine();

    foreach (var row in statistics.ColumnHeaders)
    {
        Console.Write(row.Name + "\t");
    }
    Console.WriteLine();

    foreach (var row in statistics.Rows)
    {
        var rowItems = (List<string>)row;
        Console.WriteLine(rowItems.Aggregate((s1, s2) => s1 + "\t" + s2));
    }

    Console.ReadLine();
}
}
```

جند نکته

همان نام پروژه‌ای است که ابتدای کار، در گوگل ایجاد کردیم. ApplicationName مسیر فایل مجوز ۱۲۰ آیی است که گوگل در حین ایجاد اکانت سرویس، در اختیار ما قرار می‌دهد. KeyFilePath آدرس ایمیل اکانت سرویس است که در قسمت ادمین Google Analytics به آن دسترسی دادیم. ServiceAccountEmail آدرس سایت شما است که هم اکنون در Google Analytics دارای یک اکانت و پروفایل ثبت شده است. SiteUrl توسط AnalyticsQueryParameters می‌توان نحوه‌ی کوئری گرفتن از Google Analytics را مشخص کرد. تاریخ شروع و پایان گزارش گیری در آن مشخص هستند. در مورد پارامترهایی مانند Dimensions و Metrics بهتر است به مرجع کامل آن در گوگل مراجعه نمائید:

Dimensions & Metrics Reference

برای نمونه در مثال فوق، تعداد کاربران، سیشن‌های آن و همچنین تعداد بار مشاهده‌ی صفحات، گزارشگیری می‌شود.

برای مطالعه بیشتر

Using Google APIs in Windows Store Apps

How To Use Google Analytics From C# With OAuth

#Google Analytic's API v3 with C

NET Library for Accessing and Querying Google Analytics V3 via Service Account.

#Google OAuth2 C

نظرات خوانندگان

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۳/۰۲/۲۴ ۱۰:۲۸

سلام و تشکر از مقاله خوبتان.
به نظر شما بهتر است که یک اکانت در آنالیتیکس تعریف کنیم و تمام وبسایت‌های خودمان را به این اکانت اضافه کنیم یا اینکه برای هر وبسایت یک اکانت مجزا بسازیم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۲۴ ۱۰:۳۶

- برای متده `getProfile` نوشته شده تفاوتی نمی‌کند. چون تمام اکانت‌های موجود را جهت یافتن آدرس سایت مدنظر جستجو می‌کند تا شماره پروفایل آن را برای کوئری گرفتن استخراج کند.
+ اکانت ID Client ساخته شده، برای تمام سرویس‌های متفاوت گوگل کاربرد دارد؛ و منحصر به Google Analytics نیست. فقط باید در قسمت APIs آن پروژه، سرویس موردنظر را فعال کرد.
- همچنین مزیت این روش نسبت به پروتکل‌های قدیمی گوگل که از نام کاربری و کلمه‌ی عبور Gmail استفاده می‌کردند، عدم نیاز به ذخیره سازی اطلاعات حساس اکانت گوگل خود در فایل کانفیگ برنامه است. بنابراین نیازی نیست تا اکانت‌های متفاوت گوگل را برای این کار خاص ایجاد کنید.

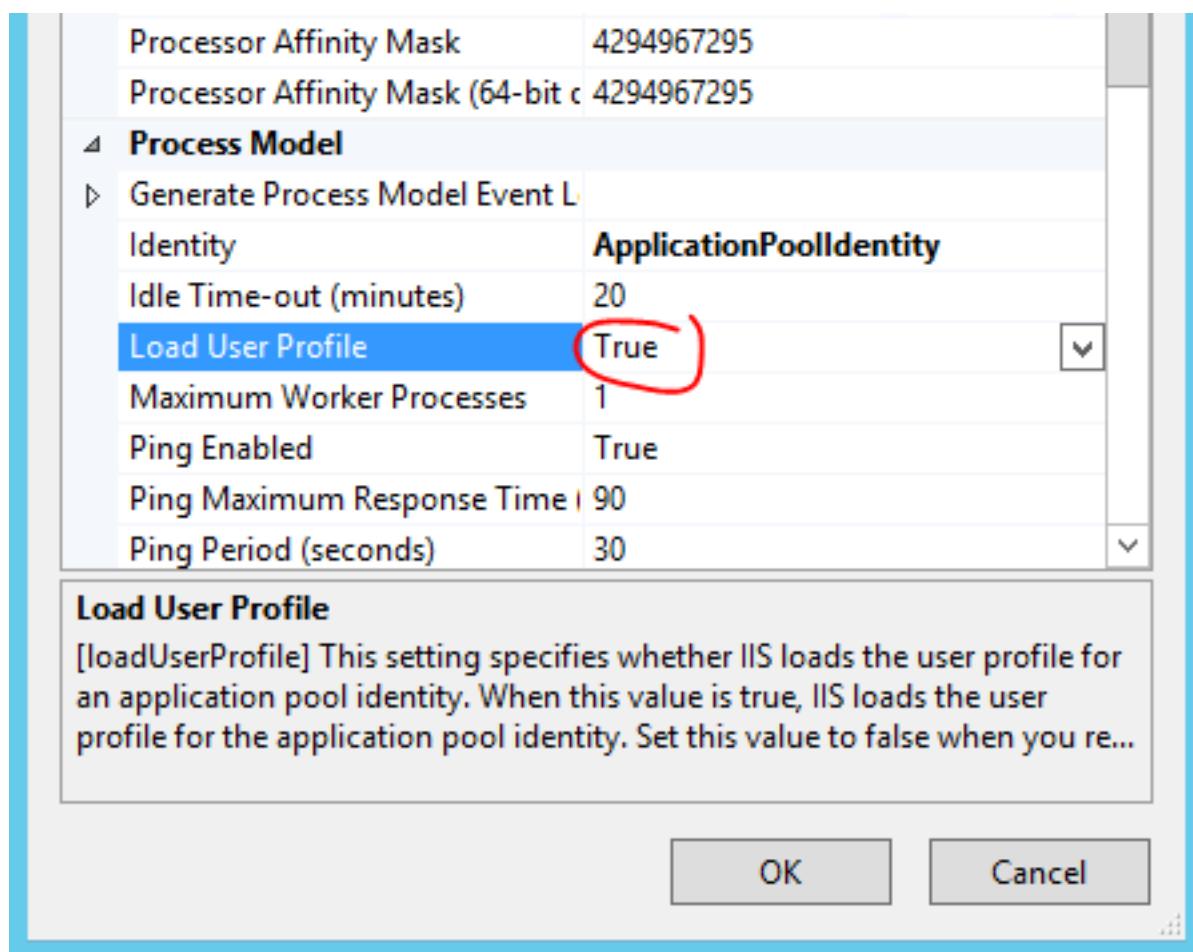
نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۳/۰۲/۲۴ ۱۲:۳۱

با سلام.
همه چیز بر روی کلاینت درست کار می‌کند ولی وقتی سایت را بر روی سرور آپلود می‌کنم خطای زیر روی می‌دهد:
`System.Security.Cryptography.CryptographicException: An internal error occurred`

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۲۴ ۱۳:۲۷

این سه مورد را بررسی کنید:
- آیا مسیر فایل `m12` را درست تنظیم کرده‌اید؟ در یک برنامه‌ی وب باید به این نحو باشد:
`Server.MapPath("~/folder/" + filename)`

- آیا در application pool برنامه‌ی شما `load user profile` کار نخواهد کرد؟



- تنظیم زیر را هم امتحان کنید:
بجای

```
var certificate = new X509Certificate2(Authentication.KeyFilePath, Authentication.KeyFilePassword,
X509KeyStorageFlags.Exportable);
```

بنویسید:

```
var certificate = new X509Certificate2(Authentication.KeyFilePath, Authentication.KeyFilePassword,
X509KeyStorageFlags.MachineKeySet |
X509KeyStorageFlags.PersistKeySet |
X509KeyStorageFlags.Exportable);
```

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۴:۲۰ ۱۳۹۳/۰۲/۲۴

حالت سوم پاسخ شما برای مورد من درست کار کرد. تشکر فراوان.

نویسنده: وحید نصیری
تاریخ: ۱۵:۲۸ ۱۳۹۳/۰۲/۲۹

یک نکته‌ی تکمیلی

اگر کتابخانه‌ی Google.Apis.Analytics.v3 را بر روی یک سیستم دات نت 4 اجرا کنید، احتمالا خطای ذیل را دریافت خواهد کرد:

```
Could not load type 'System.Net.HttpStatusCode' from assembly System.Net
```

علت اینجا است که دات نت 4 نیاز به وصله‌ی [KB2468871 portable libraries](#) را بارگذاری کند.

نویسنده: کامران
تاریخ: ۲:۵۲ ۱۳۹۳/۰۵/۲۱

از این ایراد میگیره میگه null هست، من اکانت ساختم پر اپرتی هم درست کردم

```
Line 77: var webPropertyListRequest = service.Management.Webproperties.List(account.Id);  
Object reference not set to an instance of an object.
```

نویسنده: وحید نصیری
تاریخ: ۱۰:۱۱ ۱۳۹۳/۰۵/۲۱

«من اکانت ساختم پر اپرتی هم درست کردم»
کافی نیست. تمام مراحل باید انجام شوند.

این خطاهای صرفاً به معنای غیرمعتبر بودن یکی از مراحل است که سبب شده‌اند اطلاعات آن مرحله خاص قابل دریافت نباشد.

نویسنده: کامران
تاریخ: ۱۰:۵۳ ۱۳۹۳/۰۵/۲۱

نه اینکه فقط اکانت و پر اپرتی درست کرده باشم، همه مراحل رو انجام دادم اما از خطی که خطا میده متوجه شدم که پر اپرتی‌های من رو پیدا نمیکنه یعنی خالی بر میگردونه، قبل از هم کار کرده بودم همین مشکل پیش می‌یومد.

نویسنده: وحید نصیری
تاریخ: ۱۱:۹ ۱۳۹۳/۰۵/۲۱

در متده‌ی `getProfile` :
- اگر `account` نال هست یعنی `SiteUrl` مشخص شده در ابتدای برنامه، در [اکانت آنالیتیکس](#) شما موجود نیست. روی سطر `break-point` قرار دهید تا لیست واقعی اکانت‌های ثبت شده را مشاهده کنید.
- اگر به مرحله‌ی اجرای `ServiceAccountEmail` نمی‌رسید، یعنی `accountList` معتبری وارد نشده‌است.

نویسنده: کامران
تاریخ: ۱۲:۳۲ ۱۳۹۳/۰۵/۲۱

متوجه اشکال شدم.
من کدهای شما رو کپی کردم و نخوندم چه کردید، کدها رو که بررسی کردم متوجه شدم شما توی لیست اکانتها که واکنشی میکنید اکانتی که میخوایم باید همانم یا شبیه آدرس سایت(`SiteUrl`) باشه و طبیعتاً من وقتی توی گوگل آنالیز اکانت و پروفایل‌هام رو می‌سازم نام دلخواه میدم، شما بجای استفاده از آئی دی از اسم استفاده کردید که از روی اسم آئی دی رو در بیارید برای همین چون اسم اکانت با آدرس سایتم یکی نبود چیزی پیدا نمیکرد :

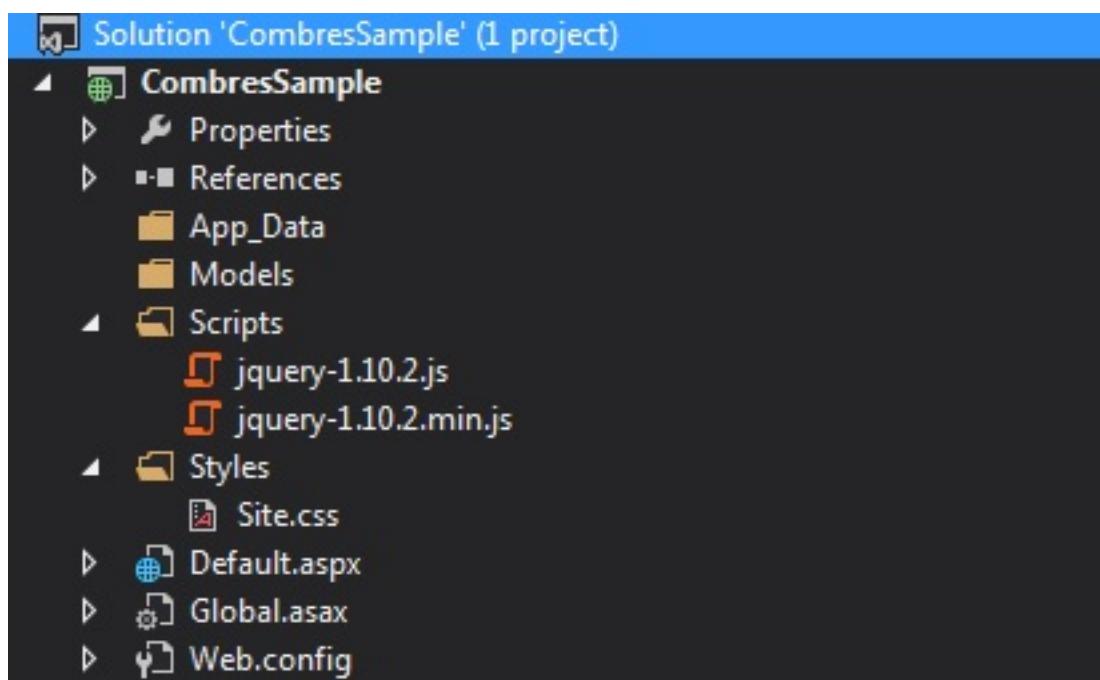
```
var account = accountList.Items.FirstOrDefault(x => x.Name.ToLowerInvariant().Contains(site));
```

برای پر اپرتی هم به همین صورت.

نویسنده: س محمد رضا برنتی
تاریخ: ۱۹:۲۰ ۱۳۹۳/۱۱/۲۵

با توجه به تحریم بودن این سرویس در ایران شما چه سرویس جایگزینی را پیشنهاد می‌کنید؟

یکی از مواردی که در توسعه وب نقش مهمی دارد، بهینه سازی فایل‌های js و css است که با فشرده سازی و کش کردن آنها می‌توان سرعت بارگذاری را تا حد چشمگیری افزایش داد. برای درک بهتر، به مثال زیر توجه کنید.
یک پروژه ساده را ایجاد می‌کنیم و فایل‌های CSS و js را مانند شکل زیر، به آن اضافه می‌کنیم:



طبق تصویر فایل‌ها را به صفحه‌ای که ساختیم اضافه می‌کنیم:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>combress</title>
    <link href="Styles/Site.css" rel="stylesheet" />
    <script src="Scripts/jquery-1.10.2.js"></script>
    <script src="Scripts/jquery-1.10.2.min.js"></script>
</head>
<body>
    <form id="form1" runat="server">
        <div>
        </div>
    </form>
</body>
</html>
```

پروژه را اجرا کرده و توسط افزونه [firebug](#) درخواست هایی را که از سرور شده است، بررسی می کنیم. مشاهده خواهید کرد که به ازای هر فایل، یک درخواست به سرور ارسال شده و هیچ کدام از فایل ها توسط وب سرور فشرده سازی نشده اند و اطلاعاتی در مورد کش، به هدر آنها اضافه نشده است.

+ GET Site.css	200 OK
+ GET jquery-1.10.2.js	200 OK
+ GET jquery-1.10.2.min.js	200 OK

برای رفع این موارد، روش های گوناگونی وجود دارد که امروز قصد داریم این کار را توسط کتابخانه Combress انجام دهیم ! نصب کتابخانه [Combres](#)

شما می توانید با استفاده از [nuget](#) این کتابخانه را به پروژه خود اضافه کنید. ایجاد فایل تنظیمات `app_data/combres.xml` در فolder `app_data` ساخته می شود که تمامی فعالیت های کتابخانه کتابخانه براساس آن انجام می شود و ساختار آن بصورت زیر است:

```
<?xml version="1.0" encoding="utf-8" ?>
<combres xmlns='urn:combres'>
    <filters>
        <filter type="Combres.Filters.FixUrlsInCssFilter, Combres" />
    </filters>
    <resourceSets url="~/combres.axd">
        defaultDuration="30"
        defaultVersion="auto"
        defaultDebugEnabled="false"
        defaultIgnorePipelineWhenDebug="true"
        localChangeMonitorInterval="30"
        remoteChangeMonitorInterval="60">
            <resourceSet name="siteCss" type="css">
                <resource path="~/content/Site.css" />
                <resource path="~/content/anotherCss.css" />
                <resource path="~/scripts/yetAnotherCss.css" />
            </resourceSet>
            <resourceSet name="siteJs" type="js">
                <resource path="~/scripts/jquery-1.4.4.js" />
            </resourceSet>
        </resourceSets>
    </combres>
```

```
<resource path="~/scripts/anotherJs.js" />
<resource path="~/scripts/yetAnotherJs.js" />
</resourceSet>
</resourceSets>
</combres>
```

با استفاده از هر **ResourceSet** می توانید آن مجموعه فایل را در یک درخواست از سرور دریافت کنید.
پارامتر **url** : برای تولید لینک فایل ها از آن استفاده می کند.

پارامتر **defaultDuration** : این عدد به تعداد روزهای پیشفرض برای کش کردن فایلها اشاره می کند.

پارامتر **defaultVersion** : در صورتی که مقدار آن **auto** باشد به ازای هر تغییر، آدرس جدیدی برای فایل موردنظر ایجاد می شود.

پارامتر **defaultDebugEnabled** : با استفاده از آن میتوانید **debug mode** را مشخص کنید. در صورتیکه مقدار آن **auto** باشد، این مقدار مستقیما از وب کانفیگ خوانده می شود.

مقدادر پیش فرض برای تمامی **ResourceSet** ها استفاده می شود و در صورت نیاز میتوان این مقادیر را برای هر **ResourceSet** بازنویسی کرد. فیلترها برای اعمال تغییراتی در فایل js و CSS استفاده می شوند که باید به این شکل معروف شوند. در قسمت بعد با

فیلترها بیشتر آشنا می شویم.

فایل **combres.xml** را به منظور استفاده در پروژه به صورت زیر تغییر می دهیم.

```
<?xml version="1.0" encoding="utf-8" ?>
<combres xmlns='urn:combres'>
  <filters>
    <filter type="Combres.Filters.FixUrlsInCssFilter, Combres" />
  </filters>
  <resourceSets url="~/combres.axd"
    defaultDuration="30"
    defaultVersion="auto"
    defaultDebugEnabled="false"
    defaultIgnorePipelineWhenDebug="true"
    localChangeMonitorInterval="30"
    remoteChangeMonitorInterval="60">
    <resourceSet name="siteCss" type="css">
      <resource path="~/Styles/Site.css" />
    </resourceSet>
    <resourceSet name="siteJs" type="js">
      <resource path="~/Scripts/jquery-1.10.2.js" />
      <resource path="~/Scripts/jquery-1.10.2.min.js" />
    </resourceSet>
  </resourceSets>
</combres>
```

اگر از نیوگت برای نصب کتابخانه استفاده کرده باشید تغییرات مورد نیاز در فایل وب کانفیگ به صورت خودکار اعمال می شود:
در غیر اینصورت باید قسمتهای زیر را به آن اضافه کنید.

```
<configuration>
  <configSections>
    <section name="combres" type="Combres.ConfigSectionSetting, Combres, Version=2.2, Culture=neutral,
PublicKeyToken=1ca6b37997dd7536" />
  </configSections>
  <system.web>
    <pages>
      <namespaces>
        <add namespace="Combres" />
      </namespaces>
    </pages>
  </system.web>
  <combres definitionUrl="~/App_Data/combres.xml" />
  <appSettings>
    <add key="CombresSectionName" value="combres" />
  </appSettings>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="AjaxMin" publicKeyToken="21ef50ce11b5d80f" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-4.84.4790.14405" newVersion="4.84.4790.14405" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>
```

حال باید RouteCombres را به RouteTable اضافه کنیم. در صورتیکه از نیوگت استفاده کرده باشید، کلاسی به فولدر اضافه شده است که با استفاده از [WebActivator](#) اینکار را در Application_Start انجام میدهد؛ در غیر اینصورت باید به صورت زیر کار را انجام دهیم.

```
protected void Application_Start(object sender, EventArgs e)
{
    RouteTable.Routes.AddCombresRoute("Combres Route");
}
```

بعد از انجام مراحل قبل، نوبت به آن رسیده است که از لینک‌های combres در صفحات خود استفاده کنیم. شیوه استفاده از آن در وب فرم به این صورت است:

```
<%@ Import Namespace="Combres" %>
<head runat="server">
    <%= WebExtensions.CombresLink("siteCss") %>
    <%= WebExtensions.CombresLink("siteJs") %>
</head>
```

و در MVC به صورت زیر می‌باشد:

```
<%= Url.CombresLink("siteCss") %>
<%= Url.CombresLink("siteJs") %>
```

در هر دو مورد نام ResourceSet برای تولید لینک به متده استفاده از CombresLink ارسال می‌شود. پس از اجرای مجدد برنامه و با استفاده از firebug خواهیم دید که به ازای هر Resource Set، یک درخواست به سرور ارسال شده است و حجم فایلها به صورت چشمگیری کاهش یافته است و اطلاعات مربوط به کش هم به آن اضافه شده است.

+ GET /combres.axd/siteCss/13518664:	200 OK
+ GET /combres.axd/siteJs/119039392:	200 OK

در ادامه می‌توانید فایل site.css قبلی و فعلی را مقایسه کنید!

GET /combres.axd/siteCss/13518664	200 OK	localhost:55439	220 B
Headers Response Cache			
body{padding-top:50px;padding-bottom:20px}.body-content{padding-left:15px;padding-right:15px;textarea{max-width:280px}.field-validation-error{color:#b94a48}.field-validation-valid{color:#4f81bd}.input.input-validation-error{border:1px solid #b94a48}input[type="checkbox"].input-validation-valid{border:0 none}.validation-summary-errors{color:#b94a48}.validation-summary-valid{color:#4f81bd}			
+ GET /combres.axd/siteJs/119039392	200 OK	localhost:55439	67.5 KB
- GET site.css	200 OK	localhost:55439	463 B
Headers Response Cache			
<pre>body { padding-top: 50px; padding-bottom: 20px; } /* Set padding to keep content from hitting the edges */ .body-content { padding-left: 15px; padding-right: 15px; } /* Set width on the form input elements since they're 100% wide by default */ input, select, textarea { max-width: 280px; }</pre>			

در قسمت بعد با سازوکار combres و روش استفاده از فیلترها، بیشتر آشنا میشویم.

[CombresProject-67bd3b7299f24c5484edc35537fa990b.zip](#)

نظرات خوانندگان

نویسنده: آرایه
تاریخ: ۹:۴۰ ۱۳۹۳/۰۲/۲۹

ضمن تشکر به خاطر توضیحات خوب، مزیت استفاده از Combres نسبت به امکان minification و bundle خود چیست؟

نویسنده: سامان هاشمی
تاریخ: ۱۳:۴۴ ۱۳۹۳/۰۲/۳۰

- شاید عدمه مزیت های Combres این چهار مورد زیر باشد:
- ۱- از کتابخانه های مختلف برای بهینه سازی پشتیبانی میکند مانند YUI Compressor for .NET Microsoft Ajax Minifier و Google Closure Compiler.
 - ۲- تعریف جدالگانه اطلاعات کش به ازای هر فایل
 - ۳- فیلترها
 - ۴- لاغ کردن پردازش های داخلی (به صورت پیشفرض توسط Log4Net انجام میگیرد)

نویسنده: rezaei
تاریخ: ۱۷:۳۶ ۱۳۹۳/۰۲/۳۰

با سلام؛ من تمامی مراحلی که فرمودید را انجام دادم. اما موقع اجرا خطای زیر را می‌گیرم

```
System.ArgumentException: Illegal characters in path
```

```
@Url.CombresLink("jquery")
```

نویسنده: سامان هاشمی
تاریخ: ۷:۵۷ ۱۳۹۳/۰۲/۳۱

احتمالاً نوع آدرس دهی یکی از ResourceSet های فایل xml یک مشکلی وجود داره! (بهتر بود نمونه کد یا پروژه اینجا قرار میدادید تا بهتر بتونم کمکتون کنم)

نویسنده: afsharsalar
تاریخ: ۱۱:۵۶ ۱۳۹۳/۰۳/۱۳

برای MVC از [Combres.Mvc](#) استفاده کنید و سپس از دستور

```
@Html.Raw(WebExtensions.CombresLink("siteCss"))
```

نویسنده: rezaei
تاریخ: ۲۳:۱۳ ۱۳۹۳/۰۶/۱۱

با سلام
موقع استفاده یکی از فایل های جاوا اسکریپت توی cache قرار میگیره و تغییرات روش اعمال نمیشه
اما وقتی به صورت عادی بدون combres فایل جاوا اسکریپت رو استفاده میکنم این مشکل وجود نداره
ولی در مورد فایل های CSS همچین مشکلی وجود نداره

ممnon میشم راهنمایی کنید

دات نت 4.5.2 قابلیت توکاری را به نام در صفت قرار دادن یک کار پس زمینه، اضافه کرده است که در ادامه خلاصه‌ای از آن را مرور خواهیم کرد.

روش متد ایجاد کارهای پس زمینه

ساده‌ترین روش انجام کارهای پس زمینه در برنامه‌های دات نتی، استفاده از متدهایی هستند که یک ترد جدید را ایجاد می‌کنند مانند Task.Run, Task.Factory.StartNew, Delegate.BeginInvoke, ThreadPool.QueueUserWorkItem و امثال آن. اما ... این روش‌ها در برنامه‌های ASP.NET ایده‌ی خوبی نیستند! از این جهت که موتور ASP.NET در این حالات اصلاً نمی‌داند که شما کار پس زمینه‌ای را ایجاد کرده‌اید. به همین جهت اگر پروسه‌ی برنامه [پس از مدتی recycle شود](#)، تمام کارهای پس زمینه‌ی موجود نیز از بین خواهد رفت.

HostingEnvironment.QueueBackgroundWorkItem معرفی

متدهای `HostingEnvironment.QueueBackgroundWorkItem` به دات نت 4.5.2 اضافه شده‌است تا بتوان توسط آن یک کار پس زمینه را توسط موتور ASP.NET شروع کرد و نه مانند قبل، بدون اطلاع آن. البته باید دقت داشت که این کارهای پس زمینه مستقل از هر نوع درخواستی اجرا می‌شوند.

در این حالت چون موتور ASP.NET از وجود کار پس زمینه‌ی آغاز شده مطلع است، در صورت فرا رسیدن زمان `recycle` شدن برنامه، کل `AppDomain` را به یکباره نابود خواهد کرد. البته این مورد فقط به این معنا است که در صورت فرا رسیدن زمان `recycle` شدن پروسه، با تنظیم یک `CancellationToken`، اطلاع رسانی خواهد کرد. در این حالت حداقل 30 ثانیه فرصت خواهد داشت تا کارهای پس زمینه را بدون مشکل خاتمه دهید. اگر کار پس زمینه در این مدت به پایان نرسد، همانند قبل، کل `AppDomain` نابود خواهد شد.

این متدهای `overload` دارند و در هر دو حالت، تنظیم خودکار پارامتر `CancellationToken` توسط ASP.NET، بیانگر آغاز زمان خاتمه‌ی کل برنامه است:

```
public static void QueueBackgroundWorkItem(Action<CancellationToken> workItem);
public static void QueueBackgroundWorkItem(Func<CancellationToken, Task> workItem);
```

در متدهای اول، یک متدهای `void` قابل پردازش است. در متدهای دوم، می‌توان متدهای `Task` دار را که قرار است کارهای `async` را پردازش کنند، معرفی نمود. علت استفاده از `Action` و `Func` در اینجا، امکان تعریف خلاصه و `inline` یک متدهای ارسال پارامتری به آن از طرف برنامه است، بجای تعریف یک اینترفیس جدید، نیاز به پیاده سازی آن اینترفیس و بعد برای مثال ارسال یک مقدار از طرف برنامه به متدهای `Stop` آن (بجای تعریف یک اینترفیس تک متدهای `Action` و `Func` [نیز می‌توان استفاده کرد](#)).

نمونه‌ای از نحوه‌ی فراخوانی این دو `overload` را در ذیل مشاهده می‌کنید:

```
HostingEnvironment.QueueBackgroundWorkItem(cancellationToken =>
{
    //todo: ...
});

HostingEnvironment.QueueBackgroundWorkItem(async cancellationToken =>
{
    //todo: ...
    await Task.Delay(20000, cancellationToken);
});
```

پشت صحنه‌ی `HostingEnvironment.QueueBackgroundWorkItem`

روش استاندارد ثبت و معرفی یک کار پس زمینه در ASP.NET، توسط پیاده سازی اینترفیسی به نام `IRegisteredObject` انجام می‌شود. سپس توسط متدهای `HostingEnvironment.RegisterObject` و `HostingEnvironment.UnregisterObject` معرفی کرد. در این حالت زمانیکه `AppDomain` قرار است خاتمه یابد، متدهای `Stop` و `Dispose` اینترفیس `IRegisteredObject` کار اطلاع رسانی را انجام می‌دهد. توسط `ThreadPool.QueueUserWorkItem` دقیقاً از همین روش به همراه فراخوانی `QueueBackgroundWorkItem` جهت اجرای متدهای `Run` و `Cancel` معرفی شده‌ی به آن استفاده می‌شود.

از مکانیزم `IRegisteredObject` در [DNT Scheduler](#) نیز استفاده شده‌است.

پیشنبازها

ابتدا نیاز است به خواص پروژه مراجعه کرده و `Target framework` را بر روی 5.4.2 قرار داد. اگر به [روز رسانی دوم VS 2013](#) نیز وجود دارد:

[NET Framework 4.5.2.](#)
[Developer pack](#)

محدودیت‌های `QueueBackgroundWorkItem`

- از آن در خارج از یک برنامه‌ی وب ASP.NET نمی‌توان استفاده کرد.
- توسط آن، خاتمه‌ی یک `AppDomain` تنها به مدت 30 ثانیه به تاخیر می‌افتد؛ تا فرصت داشته باشد کارهای در حال اجرا را با حداقل خسارت به پایان برسانید.
- یک `work item`، اطلاعاتی را از فراخوان خود دریافت نمی‌کند. به این معنا که مستقل از زمینه‌ی یک درخواست اجرا می‌شود.
- استفاده‌ی از آن الزاماً به این نیست که کار درخواستی شما حتماً اجرا خواهد شد. زمانیکه که کار خاتمه‌ی `AppDomain` آغاز می‌شود، فراخوانی‌های `QueueBackgroundWorkItem` دیگر پردازش نخواهد شد.
- اگر برنامه به مقدار `CancellationToken` تنظیم شده توسط `ASP.NET` دقت نکند، جهت پایان یافتن کار در حال اجرا، صبر نخواهد شد.

نظرات خوانندگان

نویسنده: سعید شیرعلی
تاریخ: ۱۳۹۳/۰۹/۰۸ ۱۵:۱۲

با سلام؛ من می‌خواهم یک کار در پس زمینه انجام بدم به صورتی که یک تابع را با چند آرگمن آنها در یک تابع در یک تسویه جدید اجرا بشود، به طوری که اگر من این تابع را هر چند بار که صدا بز نم همه آنها در یک تسویه جدید و مجزا از هم اجرا بشود. من چطوری می‌تونم این کار را انجام بدهم؟ با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۰۸ ۱۵:۱۸

- مطلب فوق در همین مورد است. **Queue BackgroundWorkItem** به معنای در صف قرار دادن این کارها برای پردازش مجزا از هم است.
- از آن استفاده کردید؟ به چه مشکلی برخوردید؟

نویسنده: محمد سعید
تاریخ: ۱۳۹۳/۱۱/۰۸ ۱۶:۱۹

با سلام
من می‌خواهم یک کار در پس زمینه انجام بدم که طول زمان انجامش بیشتر از خیلی بیشتر از سی ثانیه است. از اونجا که برای انجامش فقط سی ثانیه زمان دارم(طبق مطالب متن)، آیا هنوز می‌تونم از این روش استفاده کنم؟ اگه آره، چطوری؟ پیش‌آپیش از توجهتون تشکر می‌کنم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۱/۰۸ ۱۷:۴۷

«برای انجامش فقط سی ثانیه زمان دارم» خیر؛ اینطور نیست. این مورد فقط مرتبط است به زمانیکه قرار است سرور یا برنامه [ری‌استارت](#) شوند (خاتمه‌ی AppDomain). در غیر اینصورت محدودیت زمانی ندارید.

چرا JSON.NET

یک کتابخانه سورس باز کار با اشیاء JSON در دات نت است. تاریخچه آن به ۸ سال قبل بر می‌گردد و توسط یک برنامه نویس نیوزیلندی به نام James Newton King تهیه شده است. اولین نگارش آن در سال 2006 ارائه شد؛ مقارن با زمانی که اولین استاندارد JSON نیز ارائه گردید.

این کتابخانه از آن زمان تا کنون، 6 میلیون بار دانلود شده است و به علت کیفیت بالای آن، این روزها پایه اصلی بسیاری از کتابخانه‌ها و فریم ورک‌های دات نتی می‌باشد؛ مانند RavenDB تا ASP.NET Web API و SignalR مايكروسافت و همچنین گوگل نیز از آن جهت تدارک کلاینت‌های کار با API خود استفاده می‌کنند. هرچند دات نت برای نمونه در نگارش سوم آن جهت مصارف WCF کلاسی را به نام `DataContractJsonSerializer` ارائه کرد، اما کار کردن با آن محدود است به فرمت خاص WCF به همراه عدم انعطاف پذیری و سادگی کار با آن. به علاوه باید درنظر داشت که `JSON` از دات نت 2 به بعد تا مونو، Win8 و ویندوز فون را نیز پشتیبانی می‌کند.

برای نصب آن نیز کافی است دستور ذیل را در کنسول پاورشل نیوگت اجرا کنید:

```
PM> install-package Newtonsoft.Json
```

معماری JSON.NET

کتابخانه JSON.NET از سه قسمت عمده تشکیل شده است:

- الف) `JsonSerializer`
- ب) `LINQ to JSON`
- ج) `JSON Schema`

الف) `JsonSerializer`

کار `JsonSerializer` تبدیل اشیاء دات نتی به JSON و بر عکس است. مزیت مهم آن امکانات قابل توجه تنظیم عملکرد و خروجی آن می‌باشد که این تنظیمات را به شکل ویژگی‌های خواص نیز می‌توان اعمال نمود. به علاوه امکان سفارشی سازی هر کدام نیز توسط کلاسی به نام `JsonConverter`، پیش‌بینی شده است. یک مثال:

```
var roles = new List<string>
{
  "Admin",
  "User"
};
string json = JsonConvert.SerializeObject(roles, Formatting.Indented);
```

در اینجا نحوه استفاده از JSON.NET را جهت تبدیل یک شیء دات نتی، به معادل JSON آن مشاهده می‌کنید. اعمال تنظیم `Formatting.Indented` سبب خواهد شد تا خروجی آن دارای `Indentation` باشد. برای نمونه اگر در برنامه‌ی خود قصد دارید فرمت JSON تو در تويی را به نحو زیبا و خوانایی نمایش دهید یا چاپ کنید، همین تنظیم ساده کافی خواهد بود. و یا در مثال ذیل استفاده از یک `anonymous object` را مشاهده می‌کنید:

```
var jsonString = JsonConvert.SerializeObject(new
{
  Id = 1,
  Name = "Test"
}, Formatting.Indented);
```

به صورت پیش فرض تنها خواص عمومی کلاس‌ها توسط JSON.NET تبدیل خواهند شد.

تنظیمات پیشرفته‌تر JSON.NET

مزیت مهم JSON.NET بر سایر کتابخانه‌های موجود مشابه، قابلیت‌های سفارشی سازی قابل توجه آن است. در مثال ذیل نحوه معرفی می‌نمایید:

```
var jsonData = JsonConvert.SerializeObject(new
{
    Id = 1,
    Name = "Test",
    DateTime = DateTime.Now
}, new JsonSerializerSettings
{
    Formatting = Formatting.Indented,
    Converters =
    {
        new JavaScriptDateTimeConverter()
    }
});
```

در اینجا با استفاده از تنظیم `JavaScriptDateTimeConverter` استانداردی را به مصرف کنندگان جاوا اسکریپتی سمت کاربر ارائه داد؛ با خروجی ذیل:

```
{
    "Id": 1,
    "Name": "Test",
    "DateTime": new Date(1409821985245)
}
```

نوشتن خروجی JSON در یک استریم

خروجی متدهای `JsonConvert.SerializeObject` یک رشته‌است که در صورت نیاز به سادگی توسط متدهای `File.WriteAllText` در یک فایل قابل ذخیره می‌باشد. اما برای رسیدن به حداکثر کارآیی و سرعت می‌توان از استریم‌ها نیز استفاده کرد:

```
using (var stream = File.CreateText(@"c:\output.json"))
{
    var jsonSerializer = new JsonSerializer
    {
        Formatting = Formatting.Indented
    };
    jsonSerializer.Serialize(stream, new
    {
        Id = 1,
        Name = "Test",
        DateTime = DateTime.Now
    });
}
```

کلاس `response.Output` و متدهای `Serialize` آن یک استریم را نیز جهت نوشتند. برای مثال برنامه‌های وب نیز یک استریم است و در اینجا نوشتن مستقیم در استریم بسیار سریعتر است از تبدیل شیء به رشته و سپس ارائه خروجی آن؛ زیرا سربار تهیه رشته JSON از آن حذف می‌گردد و نهایتاً GC کار کمتری را باید انجام دهد.

تبدیل JSON رشته‌ای به اشیاء دات نت

اگر رشته‌ی `jsonData` ابی را که پیشتر تولید کردیم، بخواهیم تبدیل به نمونه‌ای از شیء `User` ذیل کنیم:

```
public class User
{
```

```
public int Id { set; get; }
public string Name { set; get; }
public DateTime DateTime { set; get; }
}
```

خواهیم داشت:

```
var user = JsonConvert.DeserializeObject<User>(jsonData);
```

در اینجا از متدهای `DeserializeObject` به همراه مشخص سازی صریح نوع شیء نهایی استفاده شده است. البته در اینجا با توجه به استفاده از `JavaScriptDateTimeConverter` برای تولید `json`، نیاز است چنین تنظیمی را نیز در حالت `DeserializeObject` مشخص کنیم:

```
var user = JsonConvert.DeserializeObject<User>(jsonData, new JsonSerializerSettings
{
    Converters = { new JavaScriptDateTimeConverter() }
});
```

مقدار دهی یک نمونه یا وله‌ی از پیش موجود

متدهای `JsonConvert.DeserializeObject` یک شیء جدید را ایجاد می‌کند. اگر قصد دارید صرفاً تعدادی از خواص یک وله‌ی موجود، توسط `JSON.NET` مقدار دهی شوند از متدهای `PopulateObject` استفاده کنید:

```
JsonConvert.PopulateObject(jsonData, user);
```

کاهش حجم JSON تولیدی

زمانیکه از متدهای `JsonConvert.SerializeObject` استفاده می‌کنیم، تمام خواص عمومی تبدیل به معادل JSON آنها خواهند شد؛ حتی خواصی که مقدار ندارند. این خواص در خروجی JSON، با مقدار `null` مشخص می‌شوند. برای حذف این خواص از خروجی JSON نهایی تنها کافی است در تنظیمات `JsonSerializerSettings` مقدار `NullValueHandling = NullValueHandling.Ignore` را بفرماییم.

```
var jsonData = JsonConvert.SerializeObject(object, new JsonSerializerSettings
{
    NullValueHandling = NullValueHandling.Ignore,
    Formatting = Formatting.Indented
});
```

مورد دیگری که سبب کاهش حجم خروجی نهایی خواهد شد، تنظیم `DefaultValueHandling = DefaultValueHandling.Ignore` است. در این حالت کلیه خواصی که دارای مقدار پیش فرض خودشان هستند، در خروجی JSON ظاهر نخواهند شد. مثلاً مقدار پیش فرض خاصیت `int` مساوی صفر است. در این حالت کلیه خواص از نوع `int` که دارای مقدار صفر می‌باشند، در خروجی قرار نمی‌گیرند.

به علاوه حذف `Formatting = Formatting.Indented` نیز توصیه می‌گردد. در این حالت فشرده‌ترین خروجی ممکن حاصل خواهد شد.

مدیریت ارث بری توسط JSON.NET

در مثال ذیل کلاس کارمند و کلاس مدیر را که خود نیز در اصل یک کارمند می‌باشد، ملاحظه می‌کنید:

```
public class Employee
{
    public string Name { set; get; }
}

public class Manager : Employee
{
    public IList<Employee> Reports { set; get; }
}
```

در اینجا هر مدیر لیست کارمندانی را که به او گزارش می‌دهند نیز به همراه دارد. در ادامه نمونه‌ای از مقدار دهی این اشیاء ذکر شده‌اند:

```
var employee = new Employee { Name = "User1" };
var manager1 = new Manager { Name = "User2" };
var manager2 = new Manager { Name = "User3" };
manager1.Reports = new[] { employee, manager2 };
manager2.Reports = new[] { employee };
```

با فراخوانی

```
var list = JsonConvert.SerializeObject(manager1, Formatting.Indented);
```

یک چنین خروجی JSON ای حاصل می‌شود:

```
{
  "Reports": [
    {
      "Name": "User1"
    },
    {
      "Reports": [
        {
          "Name": "User1"
        }
      ],
      "Name": "User3"
    }
  ],
  "Name": "User2"
}
```

این خروجی JSON جهت تبدیل به نمونه‌ی معادل دات نتی خود، برای مثال جهت رسیدن به manager1 در کدهای فوق، چندین مشکل را به همراه دارد:

- در اینجا مشخص نیست که این اشیاء، کارمند هستند یا مدیر. برای مثال مشخص نیست User2 چه نوعی دارد و باید به کدام شیء نگاشت شود.
- مشکل دوم در مورد کاربر User1 است که در دو قسمت تکرار شده‌است. این شیء JSON اگر به نمونه‌ی معادل دات نتی خود نگاشت شود، به دو وله‌ی از User1 خواهیم رسید و نه یک وله‌ی اصلی که سبب تولید این خروجی JSON شده‌است.

برای حل این دو مشکل، تغییرات ذیل را می‌توان به JSON.NET اعمال کرد:

```
var list = JsonConvert.SerializeObject(manager1, new JsonSerializerSettings
{
    Formatting = Formatting.Indented,
    TypeNameHandling = TypeNameHandling.Objects,
    PreserveReferencesHandling = PreserveReferencesHandling.Objects
});
```

با این خروجی:

```
{
  "$id": "1",
  "$type": "JsonNetTests.Manager, JsonNetTests",
  "Reports": [
    {
      "$id": "2",
      "$type": "JsonNetTests.Employee, JsonNetTests",
      "Name": "User1"
    },
    {
      "$id": "3",
      "$type": "JsonNetTests.Manager, JsonNetTests",
      "Reports": [
        {
          "$ref": "2"
        }
      ],
      "Name": "User3"
    }
  ],
  "Name": "User2"
}
```

- با تنظیم TypeNameHandling = TypeNameHandling.Objects سبب خواهیم شد تا خاصیت اضافه‌ای به نام type\$ به خروجی JSON اضافه شود. این نوع، در حین فراخوانی متدهای JsonConvert.DeserializeObject جهت تشخیص صحیح نگاشت اشیاء بکار گرفته خواهد شد و اینبار مشخص است که کدام شیء، کارمند است و کدامیک مدیر.
- با تنظیم PreserveReferencesHandling = PreserveReferencesHandling.Objects شماره Id خودکاری نیز به خروجی JSON اضافه می‌گردد. اینبار اگر به گزارش دهنده‌ها با دقت نگاه کنیم، مقدار ref=2 را خواهیم دید. این مورد سبب می‌شود تا در حین نگاشت نهایی، دو وله متفاوت از شیء با Id=2 تولید نشود.

باید دقت داشت که در حین استفاده از JsonConvert.DeserializeObject یاد شده، تنظیم JsonSerializerSettings نیز باید باشد.

ویژگی‌های قابل تنظیم در JSON.NET

- علاوه بر JSON که از آن صحبت شد، در JSON.NET امکان تنظیم یک سری از ویژگی‌ها به ازای خواص مختلف نیز وجود دارد.
- برای نمونه ویژگی JsonIgnore معروفترین آن‌ها است:

```
public class User
{
  public int Id { set; get; }

  [JsonIgnore]
  public string Name { set; get; }

  public DateTime DateTime { set; get; }
}
```

سبب می‌شود تا خاصیتی در خروجی نهایی JSON تولیدی حضور نداشته باشد و از آن صرفنظر شود.

- با استفاده از ویژگی JsonProperty اغلب مواردی را که پیشتر بحث کردیم مانند NullValueHandling، TypeNameHandling و غیره، می‌توان تنظیم نمود. همچنین گاهی از اوقات کتابخانه‌های جاوا اسکریپتی سمت کاربر، از اسامی خاصی که از روش‌های نامگذاری دات نتی پیروی نمی‌کنند، در طراحی خود استفاده می‌کنند. در اینجا می‌توان نام خاصیت نهایی را که قرار است رندر شود نیز صریحاً مشخص کرد. برای مثال:

```
[JsonProperty(PropertyName = "m_name", NullValueHandling = NullValueHandling.Ignore)]
public string Name { set; get; }
```

همچنین در اینجا امکان تنظیم Order نیز وجود دارد. برای مثال مشخص کنیم که خاصیت x در ابتدا قرار گیرد و پس از آن خاصیت ۷ رندر شود.

- استفاده از ویژگی JsonProperty به همراه مقدار OptIn آن به این معنا است که از کلیه خواصی که دارای ویژگی نیستند، صرفنظر شود. حالت پیش فرض آن OptOut است؛ یعنی تمام خواص عمومی در خروجی JSON حضور خواهد داشت منهای مواردی که با `JsonIgnore` مزین شوند.

```
[JsonObject(MemberSerialization.OptIn)]
public class User
{
    public int Id { set; get; }

    [JsonProperty]
    public string Name { set; get; }

    public DateTime DateTime { set; get; }
}
```

- با استفاده از ویژگی JsonConverter می‌توان نحوه‌ی رندر شدن مقدار خاصیت را سفارشی سازی کرد. برای مثال:

```
[JsonConverter(typeof(JavaScriptDateTimeConverter))]
public DateTime DateTime { set; get; }
```

تهیه یک JsonConverter سفارشی

با استفاده از `JsonConvert`ها می‌توان کنترل کاملی را بر روی اعمال serialization و deserialization مقادیر خواص اعمال کرد. مثال زیر را در نظر بگیرید:

```
public class HtmlColor
{
    public int Red { set; get; }
    public int Green { set; get; }
    public int Blue { set; get; }
}

var colorJson = JsonConvert.SerializeObject(new HtmlColor
{
    Red = 255,
    Green = 0,
    Blue = 0
}, Formatting.Indented);
```

در اینجا علاوه‌نماییم، در حین عملیات serialization، بجای اینکه مقادیر اجزای رنگ تهیه شده به صورت int نمایش داده شوند، کل رنگ با فرمت hex رندر شوند. برای اینکار نیاز است یک JsonConverter سفارشی را تدارک دید:

```
public class HtmlColorConverter : JsonConverter
{
    public override bool CanConvert(Type objectType)
    {
        return objectType == typeof(HtmlColor);
    }

    public override object ReadJson(JsonReader reader, Type objectType,
                                   object existingValue, JsonSerializer serializer)
    {
        throw new NotSupportedException();
    }

    public override void WriteJson(JsonWriter writer, object value, JsonSerializer serializer)
    {
        var color = value as HtmlColor;
        if (color == null)
```

```

        return;

        writer.WriteValue("#" + color.Red.ToString("X2")
                      + color.Green.ToString("X2") + color.Blue.ToString("X2")));
    }
}

```

کار با ارث بری از کلاس پایه `JsonConverter` شروع می‌شود. سپس باید تعدادی از متدهای این کلاس پایه را بازنویسی کرد. در متد `CanConvert` اعلام می‌کنیم که تنها اشیایی از نوع کلاس `HtmlColor` را قرار است پردازش کنیم. سپس در متد `WriteJson` منطق سفارشی خود را می‌توان پیاده سازی کرد.

از آنجائیکه این تبدیلگر صرفاً قرار است برای حالت `serialization` استفاده شود، قسمت `ReadJson` آن پیاده سازی نشده است.

در آخر برای استفاده از آن خواهیم داشت:

```

var colorJson = JsonConvert.SerializeObject(new HtmlColor
{
    Red = 255,
    Green = 0,
    Blue = 0
}, new JsonSerializerSettings
{
    Formatting = Formatting.Indented,
    Converters = { new HtmlColorConverter() }
});

```

نظرات خوانندگان

نوبت‌دهنده: افتخاری
تاریخ: ۱۳۹۳/۰۶/۱۳ ۲۱:۵۴

سلام؛ من مطالب مربوطه رو خوندم فقط اینکه تویی یه صفحه rozar در mvc من به چه نحو می‌تونم از آن استفاده کنم، حتی تویی سایت خودش هم رفته و sample‌ها رو دیدم فقط می‌خواهم در یک پروژه به چه نحو ازش استفاده کنم و کجا کارش ببیرم؟

نوبت‌دهنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۱۳ ۲۱:۵۹

در یک اکشن متده باجای return Json پیش فرض و توکار، می‌شود نوشت:

```
return Content(JsonConvert.SerializeObject(obj));
```

البته این ساده‌ترین روش استفاده از آن است؛ برای مقاصد Ajax ای. و یا برای ذکر Content type می‌توان به صورت زیر عمل کرد:

```
return new ContentResult
{
    Content = JsonConvert.SerializeObject(obj),
    ContentType = "application/json"
};
```

نوبت‌دهنده: رحمت الله رضائي
تاریخ: ۱۳۹۳/۰۶/۱۴ ۱۴:۳

"ASP.NET Web API" و SignalR از این کتابخانه استفاده می‌کنند". دلیلی دارد هنوز ASP.NET MVC از این کتابخانه استفاده نکرده است؟

نوبت‌دهنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۱۴ ۱۴:۱۹

- تا ۵ از ASP.NET MVC از JavaScriptSerializer در JsonResult استفاده می‌شود.
- در نگارش بعدی ASP.NET MVC که با Web API یکی شده (یعنی در یک کنترلر هم می‌توانید ActionResult داشته باشید و هم خروجی‌های متد اول Web API را با هم) اینبار تامین کننده‌ی JsonResult از طریق تزریق وابستگی‌ها تامین می‌شود و می‌تواند هر کتابخانه‌ای که صلاح می‌دانید باشد. البته یک مقدار پیش فرض هم دارد که دقیقاً از JSON.NET استفاده می‌کند.

نوبت‌دهنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۱۸ ۱۲:۳۵

یک نکته‌ی تکمیلی
استفاده از استریم‌ها برای کار با فایل‌ها در JSON.NET

```
public static T DeserializeFromFile<T>(string filePath, JsonSerializerSettings settings = null)
{
    if (!File.Exists(filePath))
        return default(T);

    using (var fileStream = File.OpenRead(filePath))
    {
        using (var streamReader = new StreamReader(fileStream))
        {
            using (var reader = new JsonTextReader(streamReader))
            {
                var serializer = settings == null ? JsonSerializer.Create() :
JsonSerializer.Create(settings);
            }
        }
    }
}
```

```
        return serializer.Deserialize<T>(reader);
    }
}

public static void SerializeToFile(string filePath, object data, JsonSerializerSettings
settings = null)
{
    using (var fileStream = new FileStream(filePath, FileMode.Create))
    {
        using (var streamReader = new StreamWriter(fileStream))
        {
            using (var reader = new JsonTextWriter(streamReader))
            {
                var serializer = settings == null ? JsonSerializer.Create() :
JsonSerializer.Create(settings);
                serializer.Serialize(reader, data);
            }
        }
    }
}
```

پس از بررسی مقدماتی امکانات کتابخانه‌ی JSON.NET، در ادامه به تعدادی از تنظیمات کاربردی آن با ذکر مثال‌های خواهیم پرداخت.

گرفتن خروجی CamelCase از JSON.NET

یک سری از کتابخانه‌های جاوا اسکریپتی سمت کلاینت، به نام‌های خواص [CamelCase](#) نیاز دارند و حالت پیش فرض اصول نامگذاری خواص در دات نت عکس آن است. برای مثال بجای `userName` به `UserName` نیاز دارند تا بتوانند صحیح کار کنند. روش اول حل این مشکل، استفاده از ویژگی `JsonProperty` بر روی تک خواص و مشخص کردن نام‌های مورد نیاز کتابخانه‌ی جاوا اسکریپتی به صورت صحیح است.

روش دوم، استفاده از تنظیمات `ContractResolver` می‌باشد که با تنظیم آن به `CamelCasePropertyNamesContractResolver` صورت خودکار به تمامی خواص به صورت یکسانی اعمال می‌گردد:

```
var json = JsonConvert.SerializeObject(obj, new JsonSerializerSettings
{
    ContractResolver = new CamelCasePropertyNamesContractResolver()
});
```

درج نام‌های المان‌های یک `Enum` در خروجی JSON

اگر یکی از عناصر در حال تبدیل به JSON، از نوع `enum` باشد، به صورت پیش فرض مقدار عددی آن در JSON نهایی درج می‌گردد:

```
using Newtonsoft.Json;
namespace JsonNetTests
{
    public enum Color
    {
        Red,
        Green,
        Blue,
        White
    }

    public class Item
    {
        public string Name { set; get; }
        public Color Color { set; get; }
    }

    public class EnumTests
    {
        public string GetJson()
        {
            var item = new Item
            {
                Name = "Item 1",
                Color = Color.Blue
            };

            return JsonConvert.SerializeObject(item, Formatting.Indented);
        }
    }
}
```

با این خروجی:

```
{
    "Name": "Item 1",
    "Color": 2
}
```

اگر علاقمند هستید که بجای عدد 2، دقیقا مقدار Blue در خروجی JSON درج گردد، می‌توان به یکی از دو روش ذیل عمل کرد:

(الف) مزین کردن خاصیت از نوع enum به ویژگی JsonConverter از نوع :StringEnumConverter

```
[JsonConverter(typeof(StringEnumConverter))]
public Color Color { set; get; }
```

(ب) و یا اگر می‌خواهید این تنظیم به تمام خواص از نوع enum به صورت یکسانی اعمال شود، می‌توان نوشت:

```
return JsonConvert.SerializeObject(item, new JsonSerializerSettings
{
    Formatting = Formatting.Indented,
    Converters = { new StringEnumConverter() }
});
```

تهیه خروجی JSON از مدل‌های مرتبط، بدون Stack overflow

دو کلاس گروه‌های محصولات و محصولات ذیل را درنظر بگیرید:

```
public class Category
{
    public int Id { get; set; }
    public string Name { get; set; }

    public virtual ICollection<Product> Products { get; set; }

    public Category()
    {
        Products = new List<Product>();
    }
}

public class Product
{
    public int Id { get; set; }
    public string Name { get; set; }

    public virtual Category Category { get; set; }
}
```

این نوع طراحی در Entity framework بسیار مرسوم است. در اینجا طرفهای دیگر یک رابطه، توسط خاصیتی virtual معرفی می‌شوند که به آن‌ها خواص راهبری یا navigation properties هم می‌گویند.

با توجه به این دو کلاس، سعی کنید مثال ذیل را اجرا کرده و از آن، خروجی JSON تهیه کنید:

```
using System.Collections.Generic;
using Newtonsoft.Json;
using Newtonsoft.Json.Converters;

namespace JsonNetTests
{
    public class SelfReferencingLoops
    {
        public string GetJson()
        {
            var category = new Category
            {
                Id = 1,
                Name = "Category 1"
            };
            var product = new Product
```

```
        {
            Id = 1,
            Name = "Product 1"
        };

        category.Products.Add(product);
        product.Category = category;

        return JsonConvert.SerializeObject(category, new JsonSerializerSettings
    {
        Formatting = Formatting.Indented,
        Converters = { new StringEnumConverter() }
    });
}

}
```

برنامه یا این استثناء متوقف می‌شود:

```
An unhandled exception of type 'Newtonsoft.Json.JsonSerializationException' occurred in  
Newtonsoft.Json.dll  
Additional information: Self referencing loop detected for property 'Category' with type  
'JsonNetTests.Category'. Path 'Products[0]'.
```

عملیات Serialization سبب بروز یک حلقه‌ی بازگشته، به نهایت شده و در آخر، بر نامه با خطای stack overflow خاتمه می‌یابد.

راه حل اول:

به تنظیمات JSON.NET، مقدار ReferenceLoopHandling = ReferenceLoopHandling.Ignore را اضافه کنید تا از حلقه‌ی بازگشتی در بین این حلقه‌ها جلوگیری شود:

```
return JsonConvert.SerializeObject(category, new JsonSerializerSettings
{
    Formatting = Formatting.Indented,
    ReferenceLoopHandling = ReferenceLoopHandling.Ignore,
    Converters = { new StringEnumConverter() }
});
```

راہ حل دوں:

به تنظیمات JSON.NET، مقدار `PreserveReferencesHandling = PreserveReferencesHandling.Objects` را اضافه کنید تا مدیریت ارهاگات اشیاء توسط خود JSON.NET انجام شود:

```
return JsonConvert.SerializeObject(category, new JsonSerializerSettings
{
    Formatting = Formatting.Indented,
    PreserveReferencesHandling = PreserveReferencesHandling.Objects,
    Converters = { new StringEnumConverter() }
});
```

خروجی حالت دوم به این شکل است:

```
{  
  "$id": "1",  
  "Id": 1,  
  "Name": "Category 1",  
  "Products": [  
    {  
      "$id": "2",  
      "Id": 1,  
      "Name": "Product 1",  
      "Category": {  
        "$ref": "1"  
      }  
    }  
  ]  
}
```

}]

همانطور که ملاحظه می‌کنید، دو خاصیت `id$` و `ref$` توسط JSON.NET به خروجی JSON اضافه شده است تا توسط آن بتواند ارجاعات و نمونه‌های اشیاء را تشخیص دهد.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۵:۱۷ ۱۳۹۳/۰۶/۲۳

گرفتن خروجی مرتب شده بر اساس نام خواص (جهت مقاصد نمایشی):

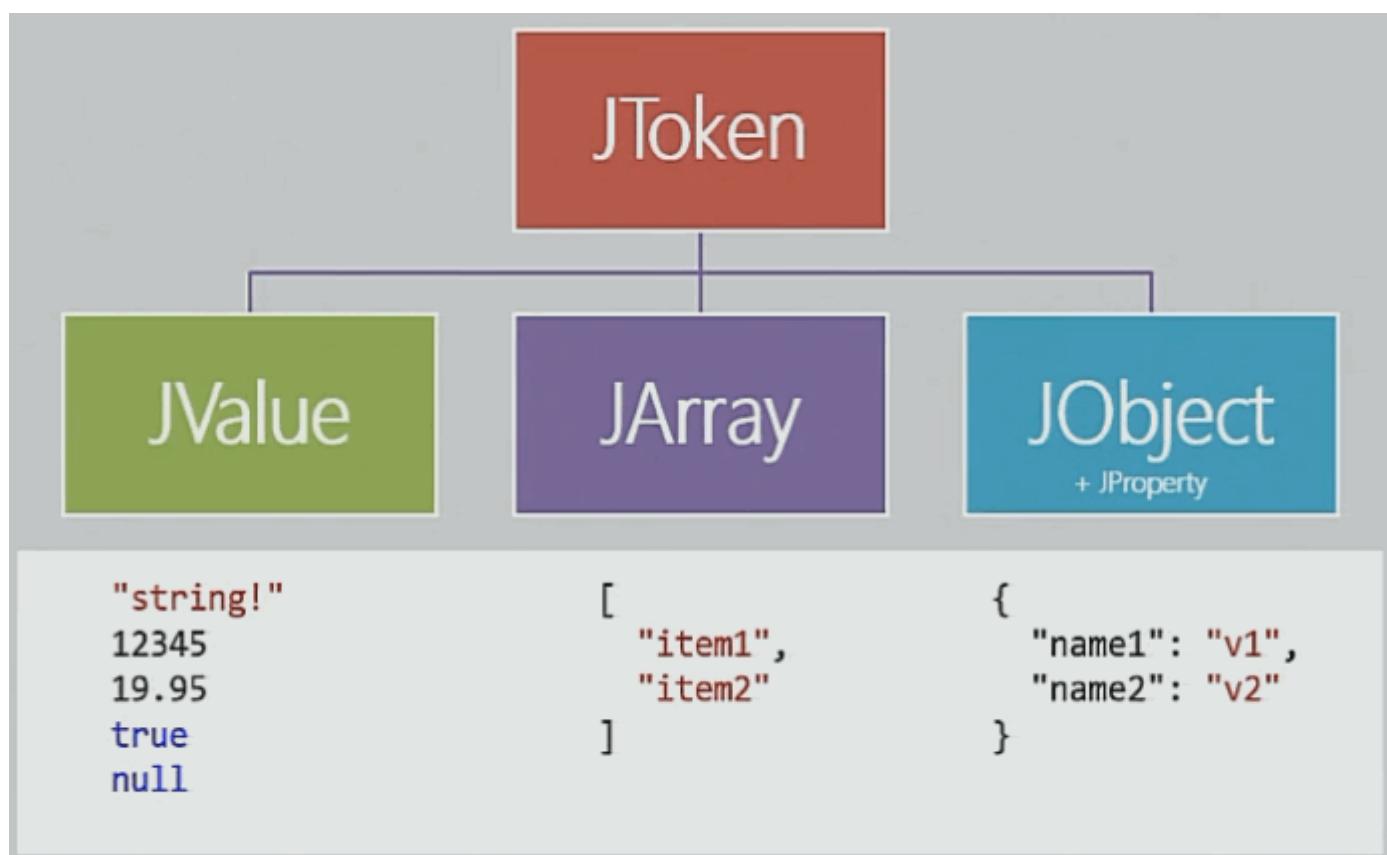
: DefaultContractResolver

```
public class OrderedContractResolver : DefaultContractResolver
{
    protected override IList<JsonProperty> CreateProperties(
        System.Type type, MemberSerialization memberSerialization)
    {
        return base.CreateProperties(type, memberSerialization).OrderBy(p =>
p.PropertyName).ToList();
    }
}
```

و بعد معرفی آن به نحو ذیل:

```
return JsonConvert.SerializeObject(data, new JsonSerializerSettings
{
    ContractResolver = new OrderedContractResolver()
});
```

عموماً از امکانات LINQ to JSON کتابخانه‌ی JSON.NET زمانی استفاده می‌شود که ورودی JSON تو در توى حجیمی را دریافت کرده‌اید اما قصد ندارید به ازای تمام موجودیت‌های آن یک کلاس معادل را جهت نگاشت به آن‌ها تهیه کنید و صرفاً یک یا چند مقدار تو در توى آن جهت عملیات استخراج نهایی مدنظر است. به علاوه در اینجا LINQ to JSON واژه‌ی کلیدی dynamic را نیز پشتیبانی می‌کند.



<code>"string!"</code> <code>12345</code> <code>19.95</code> <code>true</code> <code>null</code>	<code>[</code> <code> "item1",</code> <code> "item2"</code> <code>]</code>	<code>{</code> <code> "name1": "v1",</code> <code> "name2": "v2"</code> <code>}</code>
--	---	---

همانطور که در تصویر مشخص است، خروجی‌های JSON عموماً ترکیبی هستند از مقادیر، آرایه‌ها و اشیاء. هر کدام از این‌ها در LINQ to JSON به اشیاء JObject و JValue، JArray نگاشت می‌شوند. البته در حالت JObject هر عضو به یک JProperty تجزیه خواهد شد.
برای مثال آرایه [1,2] تشکیل شده‌است از یک JArray به همراه دو JValue که مقادیر آن را تشکیل می‌دهند. اگر مستقیماً بخواهیم یک JArray را تشکیل دهیم می‌توان از شیء JArray استفاده کرد:

```
var array = new JArray(1, 2, 3);
var arrayToJson = array.ToString();
```

و اگر یک JSON رشته‌ای دریافتی را داریم می‌توان از متده‌ی Parse مربوط به JArray کمک گرفت:

```
var json = "[1,2,3]";
var jArray= JArray.Parse(json);
var val = (int)jArray[0];
```

خروجی `JArray` یک لیست از `JToken`‌ها است و با آن می‌توان مانند لیست‌های معمولی کار کرد.

در حالت کار با اشیاء، شیء `JObject` امکان تهیه اشیاء JSON ای ب دارا است که می‌تواند مجموعه‌ای از `JProperty`‌ها باشد:

```
var jobject = new JObject(
    new JProperty("prop1", "value1"),
    new JProperty("prop2", "value2")
);
var jobjectToJson = jobject.ToString();
```

با `dynamic` به صورت `JObject` نیز می‌توان کار کرد:

```
dynamic jobject = new JObject();
jObj.Prop1 = "value1";
jObj.Prop2 = "value2";
jObj.Roles = new[] {"Admin", "User"};
```

این روش بسیار شبیه است به حالت که با اشیاء جاوا اسکریپتی در سمت کلاینت می‌توان کار کرد.
و حالت عکس آن توسط متد `JObject.Parse` قابل انجام است:

```
var json = "{ 'prop1': 'value1', 'prop2': 'value2'}";
var jObj = JObject.Parse(json);
var val1 = (string)jObj["prop1"];
```

اکنون که با اجزای تشکیل دهنده‌ی LINQ to JSON آشنا شدیم، مثال ذیل را درنظر بگیرید:

```
var array = @"[
{
    'prop1': 'value1',
    'prop2': 'value2'
},
{
    'prop1': 'test1',
    'prop2': 'test2'
}]";
var objects = JArray.Parse(array);
var obj1 = objects.FirstOrDefault(token => (string) token["prop1"] == "value1");
```

خروجی `JArray` یا `IEnumerable<JObject>` است و بر روی آن‌ها می‌توان کلیه متدهای LINQ را فراخوانی کرد. برای مثال در اینجا اولین شیءی که مقدار خاصیت `prop1` آن مساوی `value1` است، یافت می‌شود و یا می‌توان اشیاء را بر اساس مقدار خاصیتی مرتب کرده و سپس آن‌ها را بازگشت داد:

```
var values = objects.OrderBy(token => (string) token["prop1"])
    .Select(token => new {Value = (string) token["prop2"]})
    .ToList();
```

امکان انجام `sub queries` نیز در اینجا پیش بینی شده است:

```
var array = @"[
{
    'prop1': 'value1',
    'prop2': [1,2]
},
{
    'prop1': 'test1',
    'prop2': [1,2,3]
}]";
var objects = JArray.Parse(array);
var objectContaining3 = objects.Where(token => token["prop2"].Any(v => (int)v == 3)).ToList();
```

در این مثال، خواص prop2 از نوع آرایه‌ای از اعداد صحیح هستند. با کوئری نوشته شده، اشیایی که خاصیت prop2 آن‌ها دارای عضو 3 است، یافت می‌شوند.

حذف همزمان چندین رکورد GridView با استفاده از CheckBox در ASP.NET

عنوان:

عثمان رحیمی

نویسنده:

۱۳:۵ ۱۳۹۳/۰۷/۰۳

تاریخ:

www.dotnettips.info

آدرس:

ADO.NET, ASP.Net, GridView, CheckBox

گروه‌ها:

همانطور که می‌دانید GridView جزء جدایپذیر از اکثر پروژه‌های برنامه نویسان ASP.NET Web forms می‌باشد. اکثراً روشی که در میان برنامه نویسان بیشتر استفاده می‌شود، قرار دادن یک دکمه/لینک در هر ردیف از GridView برای حذف رکورد مورد نظر می‌باشد. در این مقاله قصد دارم روشی را ارائه کنم تا کاربر قادر باشد هر تعداد رکورد را که مدنظر دارد، انتخاب کرده و با فشردن دکمه "حذف" رکوردهای انتخاب شده را حذف کند.

برای درک بهتر، ابتدا جدولی به اسم "Employee" را در SQL Server با مشخصات زیر ساخته:

```
CREATE TABLE [dbo].[Employee] (
    [EmpId] INT NOT NULL,
    [FirstName] VARCHAR (20) NOT NULL,
    [LastName] VARCHAR (20) NOT NULL,
    [City] VARCHAR (20) NOT NULL,
    PRIMARY KEY CLUSTERED ([EmpId] ASC)
);
```

1- یک GridView به صفحه افزوده و خاصیت AutoGenerateColumns آن را برابر False قرار دهید . 2- فیلد‌هایی را که قصد نمایش آنها در GridView را دارید به صورت زیر به GridView بیفزایید :

```
<asp:BoundField DataField="FirstName" HeaderText="First Name" />
```

3- برای قرار دادن کنترل‌های Asp.net که در اینجا منظور CheckBox می‌باشد می‌بایست از TemplateField و قرار دادن تگ درون آن، به صورت زیر استفاده نمایید :

```
<asp:TemplateField>
    <ItemTemplate>
        <asp:CheckBox ID="chkDel" runat="server" />
    </ItemTemplate>
</asp:TemplateField>
```

و بعد از تگ GridView دکمه‌ای را برای حذف موارد انتخابی در فرم قرار دهید :

```
<asp:Button ID="btnDeleteRecord" runat="server" OnClick="btnDeleteRecord_Click" Text="Delete" />
```

برای نمایش یک پیغام به کاربر به منظور Confirm کردن دستور حذف در سمت کلاینت، قطعه کد Javascript زیر را قرار می‌دهیم:

```
function DeleteConfirm()
{
    var Ans = confirm("Do you want to Delete Selected Employee Record?");
    if (Ans)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

و در رویداد Page_Load کدهای زیر را جهت نمایش مقادیر در GridView و افزودن تابع فوق به دکمه، قبل از حذف رکوردها می‌افزاییم :

```
protected void Page_Load(object sender, EventArgs e)
```

حذف همزمان چندین رکورد GridView با استفاده از CheckBox در ASP.NET

```
{  
    if(!IsPostBack)  
    {  
        //Displaying the Data  
        showData();  
        //Adding an Attribute to Server Control(i.e. btnDeleteRecord)  
        btnDeleteRecord.Attributes.Add("onclick", "javascript:return DeleteConfirm()");  
    }  
}  
  
//Method for Displaying Data  
protected void showData()  
{  
    DataTable dt = new DataTable();  
    SqlConnection con = new SqlConnection(cs);  
    SqlDataAdapter adapt = new SqlDataAdapter("select * from Employee",con);  
    con.Open();  
    adapt.Fill(dt);  
    con.Close();  
    GridView1.DataSource = dt;  
    GridView1.DataBind();  
}
```

ابتدا تابع DeleteRecord را به صورت زیر پیاده سازی میکنیم :
که یک پارمرتر را از ورودی دریافت میکند که ID رکورد انتخاب شده میباشد و با استفاده از ID، رکورد مورد نظر را حذف میکنیم :

```
protected void DeleteRecord(int empid)  
{  
    SqlConnection con = new SqlConnection(cs);  
    SqlCommand com = new SqlCommand("delete from Employee where EmpId=@ID",con);  
    com.Parameters.AddWithValue("@ID",empid);  
    con.Open();  
    com.ExecuteNonQuery();  
    con.Close();  
}
```

و اما بخش مهم مربوط به رویداد دکمه میباشد. در هنگام کلیک بر روی دکمه باید تمامی رکوردهای GridView را چک و تمامی رکوردهایی را که آنها تیک خورده است گرفته و ID رکورد مورد نظر را به تابع DeleteRecord فرستاد و در پایان برای اعمال تغییرات، متده ShowDate را فراخوانی و GridView را مجددا Bind میکنیم.

```
protected void btnDeleteRecord_Click(object sender, EventArgs e)  
{  
    foreach (GridViewRow grow in GridView1.Rows)  
    {  
        //Searching CheckBox("chkDel") in an individual row of Grid  
        CheckBox chkdel = (CheckBox)grow.FindControl("chkDel");  
        //If CheckBox is checked than delete the record with particular empid  
        if(chkdel.Checked)  
        {  
            int empid = Convert.ToInt32(grow.Cells[1].Text);  
            DeleteRecord(empid);  
        }  
    }  
    //Displaying the Data in GridView  
    showData();  
}
```

نظرات خوانندگان

نویسنده: محمد حسین فخر اوری
تاریخ: ۱۸:۱۹ ۱۳۹۳/۰۷/۰۳

شرط بر اساس

```
where id in (.....)
```

در (chkdel.Checked) شما ID بگیرید

نویسنده: عثمان رحیمی
تاریخ: ۲۱:۴۱ ۱۳۹۳/۰۷/۰۳

بله حق با شمامست می توان کدهای فوق را نسبت به چیزی که گذاشته ام بهینه تر نوشت . ممنون از شما

تا قبل از آمدن html5 امکان آپلود چندین فایل در Asp.net web forms امکان پذیر نبود و کاربران می‌باشند فایل‌های مورد نظر خود را یکی یکی انتخاب و آپلود می‌کردند که تا حد زیادی سخت و حوصله زیادی هم می‌خواست. اما با معرفی html5 یک attribute به تگ مربوط به آپلود فایل به اسم AllowMultiple افزودن شد که مقادیر قابل قبول این attribute مقادیر بولی true, false می‌باشند. اگر این attribute به تگ‌های مربوط به آپلود فایل اضافه نشود، به صورت پیش فرض قادر خواهد بود که هر بار فقط یک فایل را برای آپلود انتخاب کنید. با مقدار دهی این Attribute true با مقدار این اجازه به شما داده می‌شود که در هر بار بتوانید چندین فایل را به صورت همزمان آپلود نمایید.

کد زیر نمایشی از چگونگی بکاربردن این attribute در تگ input می‌باشد:

```
<input type="file" multiple="multiple" name="FileUpload1" id="FileUpload1" />
```

این امکان از نسخه ASP.NET 4.5 Framework به Asp.net Web forms اضافه شده و برنامه نویسان می‌توانند در صورت نیاز از این امکان استفاده کنند. کد زیر، نحوه افزودن این attribute به تگ FileUpload در ASP.Net Web Forms می‌باشد:

```
<asp:FileUpload runat="server" ID="FileUploadMultiple" AllowMultiple="true" />
```

و اما به چه روشی می‌توانیم فایل یا فایل‌های انتخاب شده را در این حالت آپلود کرد؟ در صورتیکه از این روش در پروژه‌هایتان استفاده کنید فقط کافیست با یک حلقه تمامی کنترل‌های مورد نظر را پیمایش و هر کدام از فایل‌ها را آپلود و ذخیره نمایید.

برای درک بهتر مطلب، پروژه جدیدی در Asp.net web Forms ایجاد کرده و کنترل‌های زیر را به آن اضافه کنید:

```
<asp:FileUpload runat="server" ID="FileUploadMultiple" AllowMultiple="true" />
<asp:Button runat="server" ID="btnUplad" Text="Upload" OnClick="btnUplad_Click" />
<asp:Label runat="server" ID="lblMessage"></asp:Label>
```

و در رویداد مربوط به دکمه آپلود، قطعه کد زیر را قرار دهید :

```
int Count = 0;
foreach (var item in FileUploadMultiple.PostedFiles)
{
    string Extension = Path.GetExtension(item.FileName);
    string FileName = new Random().Next(1, 50).ToString() + Extension;
    item.SaveAs(Server.MapPath("~/") + "/File/" + FileName);
    Count++;
}
if (Count == FileUploadMultiple.PostedFiles.Count)
    lblMessage.Text = string.Format("آپلود شدند");
else
    lblMessage.Text = string.Format("از {0} فایل با موفقیت آپلود شد {1}, Count,
FileUploadMultiple.PostedFiles.Count);
```

کدهای پروژه [Upload-multiple-.rar](#) ;

در پژوهش های بزرگ نرم افزاری، از قدیم بحث تامین امنیت پژوهه، یکی از چالش های مهم بوده است. از دیدگاه شخصی بند، یک مدیر نرم افزار یا حتی یک توسعه دهنده برنامه های تحت وب، لازم است علاوه بر صرف وقت مطالعاتی و آشنایی و تسلط بر مباحث طراحی معماری سیستم های تحت وب، که از اهمیت بالا و مقیاس بزرگی برخوردارند آشنایی لازم را با چالش های امنیتی در پیاده سازی این گونه سیستم ها داشته باشد. امنیت در یک سیستم بزرگ و ارائه دهنده خدمات، باعث می شود تا کاربر علاوه بر یک تجربه کاربری (user experience) خوب از سیستم که حاصل پیاده سازی صحیح سیستم می باشد، اعتماد ویژه ای به سیستم مذکور داشته باشد. گاهها کاربران به علت بی اعتمادی به شرایط امنیتی حاکم بر یک سیستم، از تجربه کاربری خوب یک سیستم چشم پوشی می کنند. اهمیت این مسئله تا جاییست که غول های تکنولوژی دنیا همچون Google درگیر این چالش می باشند و همیشه سعی بر تامین امنیت کاربران علاوه بر ایجاد تجربه کاربری خوب دارند. پس عدم توجه به این موضوع میتواند خسارات وارد جبران ناپذیری را به یک سیستم از جهت های مختلف وارد کند.

در این سری از مقالات، بند سعی دارم تا حد توان در رابطه با چالش های امنیتی موجود در زمینه توسعه برنامه های تحت وب، مطالبی را منتشر کنم. از این رو امیدوارم تا این سری از مقالات برای دوستان مفید واقع گردد.

در این سری از مقالات چالش های امنیتی زیر مورد بحث و بررسی واقع خواهد گردید

XSS , LDAPi , RFI , LFI , SQLi , RFD , LFD , SOF , BSQLI , DNN , BOF , CRLF , CSRF , SSI , PCI , SCD , AFD , RCE

در بخش اول از این سری مقالات، به بررسی آسیب پذیری **Cross-site scripting** می پردازیم.

واژه XSS مخفف Cross-site scripting، نوعی از آسیب پذیریست که در برنامه های تحت وب نمود پیدا می کند. به طور کلی و خلاصه، این آسیب پذیری به فرد نفوذ کننده اجازه تزریق اسکریپت هایی را به صفحات وب، می دهد که در سمت کاربر اجرا می شوند (Client Side scripts) . در نهایت این اسکریپت ها توسط سایر افرادی که از صفحات مورد هدف قرار گرفته بازدید می کنند اجرا خواهد شد.

هدف از این نوع حمله :

بدست آوردن اطلاعات کوکی ها و سشن های کاربران (مرتبط با آدرسی که صفحه آلوده شده در آن قرار دارد) است. سپس فرد نفوذ کننده متناسب با اطلاعات بدست آمده می تواند به اکانت شخصی کاربران مورد هدف قرار گرفته، نفوذ کرده و از اطلاعات شخصی آن ها سوء استفاده کند.

به صورت کلی دو طبقه بندی برای انواع حملات Cross-site scripting وجود دارند.

حملات XSS ذخیره سازی شده (Stored XSS Attacks) :

در این نوع، کدهای مخرب تزریق شده، در سرور سایت قربانی ذخیره می شوند. محل ذخیره سازی می تواند دیتابیس سایت یا هر جای دیگری که داده ها توسط سایت یا برنامه تحت وب بازیابی می شوند و نمایش داده می شوند باشد. اما اینکه چگونه کدهای مخرب در منابع یاد شده ذخیره می شوند؟

فرض کنید در سایت جاری آسیب پذیری مذکور وجود دارد. راه های ارسال داده ها به این سایت چیست؟ نویسنده گان میتوانند مطلب ارسال کنند و کاربران میتوانند نظر دهند. حال اگر در یکی از این دو بخش بررسی های لازم جهت مقابله با این آسیب پذیری

وجود نداشته باشد و نوشته‌های کاربران که می‌تواند شامل کدهای مخرب باشد مستقیماً در دیتابیس ذخیره شده و بدون هیچ اعتبار سنجی نمایش داده شود چه اتفاقی رخ خواهد داد؟ مسلماً با بازدید صفحه آلوده شده، کدهای مخرب بر روی مرورگر شما اجرا و کوکی‌های سایت جاری که متعلق به شما هستند برای هکر ارسال می‌شود و ...

حملات XSS منعکس شده (Reflected XSS Attacks)

در این نوع از حمله، هیچ نوع کد مخربی در منابع ذخیره سازی و سایت یا اپلیکیشن تحت وب توسط فرد مهاجم ذخیره نمی‌شود! بلکه از ضعف امنیتی بخش‌هایی همچون بخش جستجو و ب سایت، بخش‌های نمایش پیغام خطأ و ... استفاده می‌شود ... اما به چه صورت؟

در بسیاری از سایتها، انجمن‌ها و سیستم‌های سازمانی تحت وب، مشاهده می‌شود که مثلاً در بخش جستجو، یک فیلد برای وارد کردن عبارت جستجو وجود دارد. پس از وارد کردن عبارت جستجو و submit فرم، علاوه بر نمایش نتایج جستجو، عبارت جستجو شده نیز به نمایش گذاشته می‌شود و بعضاً در بسیاری از سیستم‌ها این عبارت قبل از نمایش اعتبار سنجی نمی‌شود که آیا شامل کدهای مخرب می‌باشد یا خیر. همین امر سبب می‌شود تا اگر عبارت جستجو شامل کدهای مخرب باشد، آن‌ها به همراه نتیجه‌ی جستجو اجرا شوند.

اما این موضوع چگونه مورد سوء استفاده قرار خواهد گرفت؟ مگر نه اینکه این عبارت ذخیره نمی‌شود پس با توضیحات فوق، کد فقط بر روی سیستم مهاجم که کد جستجو را ایجاد می‌کند اجرا می‌شود، درست است؟ بله درست است ولی به نقطه ضعف زیر توجه کنید؟

www.test.com/search?q=PHNjcm1wdD5hbGVydChkb2N1bwVudC5jb29raWUpOzwvc2NyaXB0Pg==

این آدرس حاصل submit شدن فرم جستجو و سایت test (نام و سایت واقعی نیست و برای مثال است) و ارجاع به صفحه نتایج جستجو می‌باشد. در واقع این لینک برای جستجوی یک کلمه یا عبارت توسط این و سایت تولید شده و از هر کجا به این لینک مراجعه کنید عبارت مورد نظر مورد جستجو واقع خواهد شد. در واقع عبارت جستجو به صورت Base64 به عنوان یک query String به و سایت ارسال می‌شود؛ علاوه بر نمایش نتایج، عبارت جستجو شده نیز به کاربر نشان داده شده و اگر آسیب پذیری مورد بحث وجود داشته باشد و عبارت شامل کدهای مخرب باشد، کدهای مخرب بر روی مرورگر فردی که این لینک را باز کرده اجرا خواهد شد!

در این صورت کافیست فرد مهاجم لینک مخرب را به هر شکلی به فرد مورد هدف بدهد (مثلاً ایمیل و ...). حال در صورتیکه فرد لینک را باز کند (با توجه به اینکه لینک مربوط به یک سایت معروف است و عدم آگاهی کاربر از آسیب پذیری موجود در لینک، باعث باز کردن لینک توسط کاربر می‌شود)، کدها بر روی مرورگر اجرا شده و کوکی‌های سایت مذکور برای مهاجم ارسال خواهد شد ... به این نوع حمله XSS، نوع انعکاسی می‌گویند که کاملاً از توضیحات فوق الذکر، دلیل این نامگذاری مشخص می‌باشد.

اهمیت مقابله با این حمله:

برای نمونه این نوع باگ حتی تا سال گذشته در سرویس ایمیل یاهو وجود داشت. به شکلی که یکی از افراد انجمن hackforums به صورت Private این باگ را به عنوان Yahoo 0-Day XSS Exploit در محیط زیر زمینی و بازار سیاه هکرهای به مبلغ چند صد هزار دلار به فروش می‌رساند. کاربران مورد هدف کافی بود تا فقط یک ایمیل دریافتی از هکر را باز کنند تا کوکی‌های سایت یاهو برای هکر ارسال شده و دسترسی ایمیل‌های فرد قربانی برای هکر فراهم شود ... (در حال حاضر این باگ در یاهو وجود ندارد).

چگونگی جلوگیری از این آسیب پذیری

در این سری از مقالات کدهای پیرامون سرفصل‌ها و مثال‌ها با ASP.net تحت فریم ورک MVC و به زبان C# خواهند بود. هر چند کلیات مقابله با آسیب پذیری‌هایی از این دست در تمامی زبان‌ها و تکنولوژی‌های تحت وب یکسان می‌باشد.

خوب شنیدن کتابخانه ای قدرتمند برای مقابله با حمله مورد بحث وجود دارد با نام AntiXSS که میتوانید آخرین نسخه آن را با فرمان زیر از طریق nugget به پروژه خود اضافه کنید. البته ذکر این نکته حائز اهمیت است که Asp.net و فریم ورک MVC به صورت توکار تا حدودی از بروز این حملات جلوگیری می کند. برای مثال به این صورت که در View ها شما تا زمانی که از MvcHtmlString استفاده نکنید تمامی محتوای مورد نظر برای نمایش به صورت Encode شده رندر می شوند. این داستان برای Url ها هم که به صورت پیش فرض encode می شوند صدق می کند. ولی گاهها وقتی شما برای ورود اطلاعات مثلا از یک ادیتور WYSIWYG استفاده می کنید و نیاز دارید داده ها را بدون encoding رندر کنید. آنگاه به ناچار مجاب بر اعمال یک سری سیاست های خاص تر بر روی داده مورد نظر برای رندر می شوید و نمی توانید از encoding توکار فوق الذکر استفاده کنید. آنگاه این کتابخانه در اعمال سیاست های جلوگیری از بروز این آسیب پذیری می تواند برای شما مفید واقع شود.

PM> Install-Package AntiXSS

این کتابخانه مجموعه ای از توابع کد کردن عبارات است که از مواردی همچون Html, XML, Url, Form, LDAP, CSS, JScript and VBScript پشتیبانی می کند. استفاده از آن بسیار ساده می باشد. کافیست ارجاعات لازم را به پروژه خود افزوده و به شکل زیر از توابع ارائه شده توسط این کتابخانه استفاده کنید:

```
...  
var reviewContent = model.UserReview;  
reviewContent = Microsoft.Security.Application.Encoder.HtmlEncode(review);  
...
```

امیدوارم در اولین بخش از این سری مقالات، به صورت خلاصه مطالب مهمی که باعث ایجاد فهم کلی در رابطه با حملات XSS وجود دارد، برای دوستان روشن شده و پیش زمینه فکری برای مقابله با این دست از حملات برایتان به وجود آمده باشد.

نظرات خوانندگان

نویسنده: مجید و مسعود منظوری
تاریخ: ۱۲:۱۳ ۱۳۹۳/۱۰/۰۴

سلام

جایی دیدیم که نوعی از حملات هست که هکر تو یکی از پوشه های سایت (مثلاً پوشه Images) یک اسکریپت یا یک صفحه asp قرار میده و کاربر رو به سایت خودش هدایت میکنه، یا اینکه یک عکس نشون میده ظاهراً این حملات بیشتر برای استفاده از سایت هدف برای تبلیغات به کار میره راه مقابله با این حملات چی میتونه باشه؟
این حملات به نام حملات CRLF شناخته میشن گویا

نویسنده: وحید نصیری
تاریخ: ۱۲:۵۴ ۱۳۹۳/۱۰/۰۴

دسترسی اجرایی را از این پوشه ها با تنظیم ذیل در وب کانفیگ سایت، بگیرید:

```
<location path="upload">
  <system.webServer>
    <handlers accessPolicy="Read" />
  </system.webServer>
</location>
```

اطلاعات بیشتر: [^](#) و [_](#)

نویسنده: سید مهران موسوی
تاریخ: ۱۳:۲۳ ۱۳۹۳/۱۰/۰۴

در واقع آپلود فایل های مخرب نوعی حفره امنیتی در توسعه اپلیکیشن هست که موجب سوء استفاده میتوانه واقع بشه . این حفره امنیتی به [Unrestricted File Upload](#) معروفه که با نکته ای که آقای نصیری ذکر کردن قابل حل هست در asp net (لینک های ارجاعی رو مطالعه بفرمایید)

در رابطه با ارجاع کاربران به سایت هدف نوع حمله همون CRLF هست که شامل xss هم میشه .
مقابله با این نوع حملات ساده و استوار بر دو اصل اساسی هست :
1: همیشه بر این قانون که به ورودی داده های کاربر اعتماد نکنید استوار باشد
2: تمامی ورودی های کاربران را که قرار است مورد مشاهده در عموم و خصوص کاربران در مرورگر باشد تا حد ممکن پاکسازی کنید
مطالعه بیشتر [^](#)

استانداردی است بر روی پروتکل HTTP که Response های مدیریت یک فایل را بر روی سرویس دهنده وب، [WebDAV](#) تشریح می‌کند.

برای درک چرایی وجود این استاندارد بهتر است ذهن خود را معطوف به نحوه عملکرد سیستم فایل در OS کنیم که شامل API های خاص برای دسترسی نرم افزارهای گوناگون به فایل‌های روی یک سیستم است. حال فکر کنید یک سرور Cloud راه اندازی نموده‌اید که قرار است مدیریت فایل‌ها و پرونده‌های Office را بر عهده داشته باشد و چون امکان ویرایش اسناد Office بر روی وب را ندارید، نیاز است تا اجازه دهید نرم افزارهای Office مستقیماً فایل‌ها را از روی سرور شما باز کنند و بعد از تغییرات، به جای ذخیره در سیستم local، محتوا را به فایل روی سرور ارسال کنند. در مفهوم web عملاً این کار غیر استاندارد و نادرست است. همه درخواست‌ها و جواب‌ها باید بر روی پروتکل Http باشند. خوب حال تصور کنید نرم افزارهای Office قابلیت آن را داشته باشند که به جای تحویل محتوا به سیستم عامل برای ذخیره‌ی آن بر روی سیستم local، محتوا را به یک آدرس ارسال نمایند و پشت آن آدرس، متده باشد که بتواند به درخواست رسیده، به درستی پاسخ دهد.

این یعنی باید سمت سرور متده با قابلیت ارسال پاسخ‌های درست و در سمت کلاینت نرم افزاری با قابلیت ارسال درخواست‌های مناسب وجود داشته باشد.

WebDAV استاندارد تشریح محتوا درخواست‌ها و پاسخ‌های مربوط به مدیریت فایل‌ها است. خوشبختانه نرم افزارهای Office و بسیاری از نرم افزارهای دیگر، استاندارد WebDAV را پشتیبانی می‌کنند و فقط لازم است برای سرورتان متده با قابلیت پشتیبانی از درخواست‌های WebDAV پیاده سازی نمایید و البته متناسبانه کتابخانه‌های سورس باز چندانی برای WebDAV در سرور دات نت وجود ندارد. من مژولی را برای کار با WebDAV نوشتم و سورسش را در [Git](#) قرار دادم. برای این مثال هم از [همین کتابخانه](#) استفاده می‌کنم.

ابتدا یک پروژه‌ی وب MVC ایجاد نمایید و پکیج xDav را از nugget نصب کنید.

```
PM> Install-Package xDav
```

اگر به web.config نگاهی بیاندازیم می‌بینیم یک module به نام xDav به وب سرور اضافه شده که بررسی درخواست‌های WebDAV را به عهده دارد.

```
<system.webServer>
  <modules>
    <add name="xDav" type="xDav.XDavModule, XDav" />
  </modules>
</system.webServer>
```

همچنین یک config جدید هم به config برای پیکربندی xDav اضافه شده است.

```
<XDavConfig Name="xdav">
  <FileLocation URL="xdav" PathType="Local"></FileLocation>
</XDavConfig>
```

خاصیت Name برای xDav نشانگر درخواست‌هایی است که باید توسط این مژول اجرا شوند. در اینجا یعنی درخواست‌هایی که آدرس آن‌ها شامل "/xdav" باشد، توسط این مژول Handle می‌شوند. عبارت بعد از مقدار Name در URL هم طبیعتاً نام فایل مورد نظر شماست.

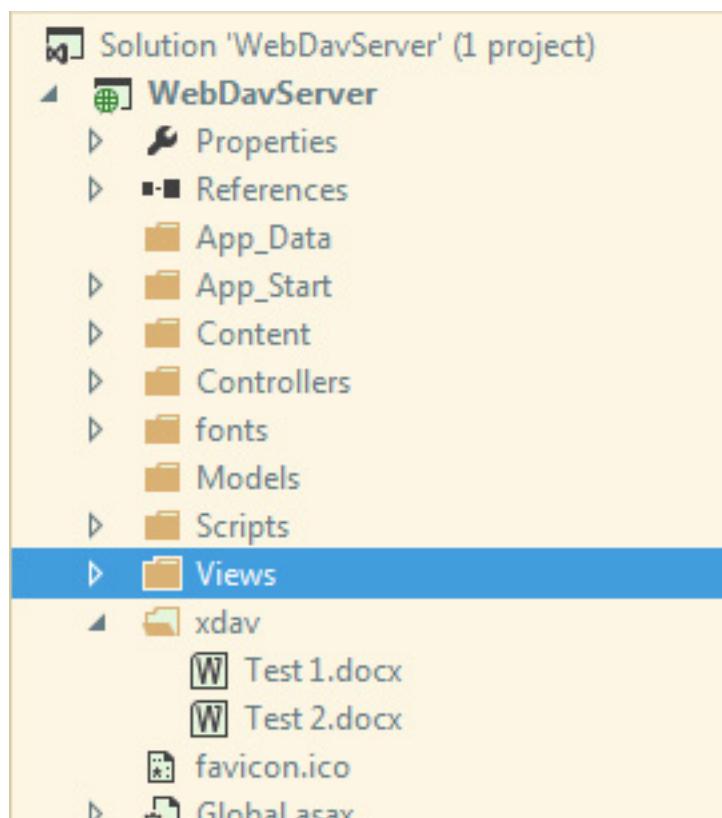
آدرس پوشه‌ای است که فایل‌ها در آن ذخیره و یا بازخوانی می‌شوند. اگر FileType با مقدار Local تنظیم شود،

آشنایی با WebDav و نحوه استفاده از آن

یعنی باید یک پوشه به نام خاصیت URL که در اینجا x dav است در پوششی اصلی وب شما وجود داشته باشد و اگر با Server مقدار دهی شود URL باید یک آدرس فیزیکی بر روی سرور داشته باشد . مثل "URL=c:\webdav"

ما در این مثال مقادیر را به صورت پیشفرض نگه می داریم. یعنی باید در پوششی وب، یک Folder با نام x dav ایجاد کنیم.

در ادامه چند فایل word را برای تست در این پوشه کپی می کنیم.



می خواهیم در صفحه Index، لیستی از فایل های درون این پوشه را نمایش دهیم طوری که در صورت کلیک بر روی هر کدام از آنها، آدرس WebDav فایل مورد نظر را به Word ارسال کنیم.

بعد از نصب Office، در registry چند نوع Url تعریف می شود که معرف اپلیکیشنی است که آدرس به آن فرستاده شود. این دقیقا همان چیزیست که ما به آن نیاز داریم. کافیست آدرس WebDav فایل را بعد از عبارت "ms-word:ofe|u|" در یک لینک قرار دهیم تا آدرس به نرم افزار Word ارسال شود. یعنی آدرس URL باید این شکلی باشد:

```
ms-word:ofe|u|http://Webaddress/xdav/filename
```

آدرس Webaddress آدرس وبسایت و filename نام فایل مورد نظرمان است. عبارت /xdav/ هم که نشان می دهد درخواست هایی که این آدرس را دارند باید توسط ماژول xDav پردازش شوند.

کلاسی با نام DavFile در پوششی Model ایجاد می کنیم:

```
public class DavFile
{
    public string Name { get; set; }
```

آشنایی با WebDav و نحوه استفاده از آن

```
public string Href(string webAddress)
{
    return string.Format("ms-word:ofe|u|http://{0}/xdav/{1}", webAddress, Name);
}
```

اکشن متد Index را در Home Controller، مانند زیر تغییر دهید:

```
var dir = new DirectoryInfo(XDav.Config.ConfigManager.DavPath);
        var model = dir.GetFiles().ToList()
                            .Select(f =>
                                new DavFile() {
                                    Name = f.Name
                                });
return View(model);
```

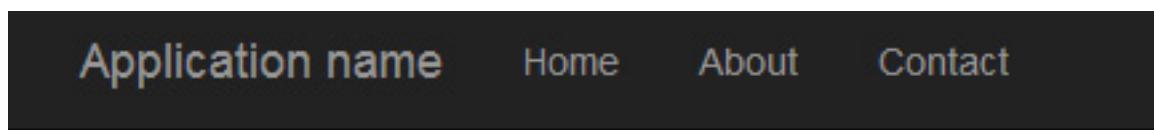
یک لیست از فایل هایی که در پوششی webDav قرار دارند تهیه می کنیم و به View ارسال می کنیم. View را هم مثل زیر بازنویسی می کنیم.

```
@model IEnumerable<WebDavServer.Models.DavFile>
<h1>
    File List
</h1>

<ul>
    @foreach (var item in Model)
    {
        <li> <a href="@Html.Raw(item.Href(ViewContext.HttpContext.Request.Url.Authority))">
@Html.Raw(item.Name) </a></li>
    }
</ul>
```

قرار است به ازای هر فایل، لینکی نمایش داده شود که با کلیک بر روی آن، آدرس فایل به word ارسال می شود. بعد از ثبت تغییرات، word محتوا را به همان آدرس ارسال می کند و ماژول xDav محتوا را در فایل فیزیکی سرور ذخیره خواهد کرد.

برنامه را اجرا کنید و بر روی فایل ها کلیک نمایید. اگر نرم افزار Office روی کامپیوترتان باز باشد با کلیک بر روی هر کدام از فایل ها، فایل word باز شده و می توانید محتوا را تغییر داده و ذخیره نمایید.



File List

- [Test1.docx](#)
- [Test2.docx](#)

© 2014 - My ASP.NET Application

نرم افزار کلاینت (word) درخواست هایی با verb های مشخص که در استاندارد WebDav ذکر شده به آدرس مورد نظر می فرستد. سرور WebDav درخواست را بر اساس Verb آن Request پردازش کرده و Response استاندارد را ایجاد می کند.

نرم افزار word پس از دریافت یک URL، به جای فرمت فیزیکی فایل، درخواست هایی را با تایپ های Option, Head, lock, get, post و unlock ارسال می کند. محتوای درخواست و پاسخ هر کدام از تایپ ها در استاندارد webDav تعريف شده و مازول xDav را پیاده سازی نموده است.

[دریافت پروژه مثال](#)

نظرات خوانندگان

نویسنده: نمو
تاریخ: ۱۳۹۳/۰۹/۱۵ ۱۱:۴۹

سلام؛ ممنون.

آیا از مرورگر خاصی باید استفاده شود؟ پروژه نمونه هم کار نمی کند و وقتی روی لینک های کلیک می کنم هیچ اتفاقی نمی افتد.

نویسنده: رضا بازرگان
تاریخ: ۱۳۹۳/۰۹/۱۵ ۱۳:۴۶

با سلام.

طمئن شوید که Office را سیستم شما نصب است. من Package ها را از پروژه نمونه حذف کردم. لطفاً مجدداً آن را نصب کنید.

فرقی در استفاده از مرورگر هم نیست. می توانید از پروژه ای که روی [Git](#) گذاشتم هم استفاده کنید که کاملتر است.

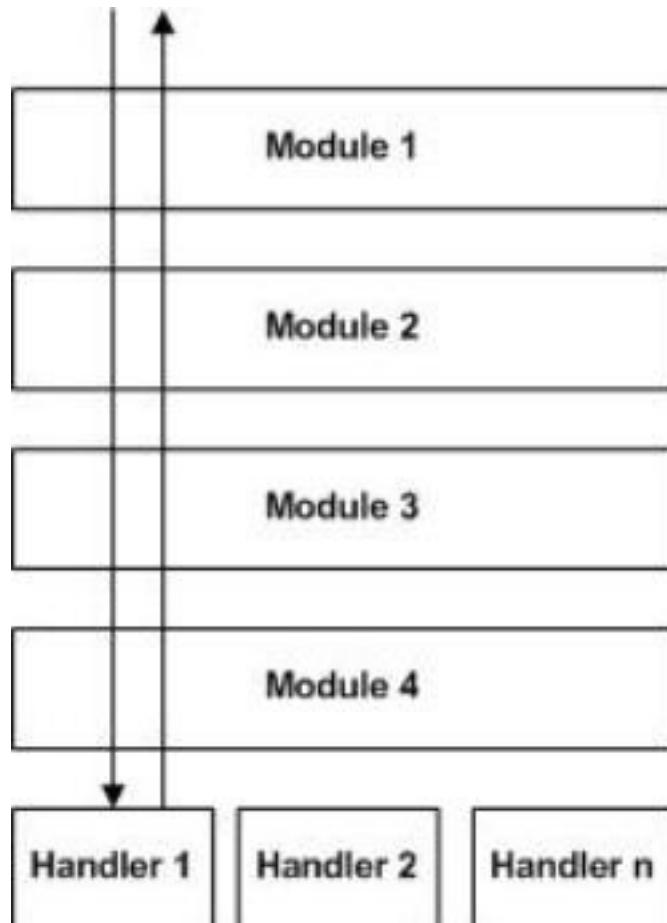
قبل از اینکه به httpmodule ها پردازیم، اجازه بدید کمی در در مورد **httphandler** اطلاعات کسب کنیم. **httphandler** ویژگی است که از asp.net به بعد ایجاد شد و در asp کلاسیک خبری از آن نیست.

یک httpmodule کامپوننتی است که از طریق اینترفیس System.Web.IHttpHandler پیاده سازی میشود و به پردازش درخواست های رسیده از httprequest رسیدگی می کند.

فرض کنید کاربری درخواست صفحه default.aspx را کرده است و سرور هم پاسخ آن را می دهد. در واقع پردازش اینکه چه پاسخی باید به کاربر یا کلاینت ارسال شود بر عهده این کامپوننت می باشد. برای وب سرویس هم موضوع به همین صورت است: هر نوع درخواست HTTP از این طریق انجام می شود.

حال به سراغ httpmodule می رویم. httpmodule ها اسمبلی یا ماژول هایی هستند که بر سر راه هر درخواست کاربر از سرور قرار گرفته و قبل از اینکه درخواست شما به httphandler برسد، اول از فیلتر اینها رد می شود. در واقع موقعی که شما درخواست صفحه default.aspx را می کنید، درخواست شما به موتور asp.net ارسال می شود و از میان فیلتر هایی رد می شود تا به دست httpmodule برای پردازش خروجی برسد. برای همین اگر گاهی به جای گفتن asp.net engine عبارت asp.net pipeline هم میگویند همین هست: چون درخواست شما از بین بخش های زیادی می گذرد تا به httpmodule برسد که یکی از آن بخش هاست. با هر درخواستی که سرور ارسال می شود، httpmodule ها صدای زده می شوند و به برنامه نویس امکان بررسی اطلاعات درخواستی و پردازش درخواست ها را در ورودی و خروجی، می دهد و شما میتوانید هر عملی را که نیاز دارید انجام دهید. تعدادی از این ماژول های آماده، همان state و authentication می باشند.

تصویر زیر نحوه ارسال و بازگشت یک درخواست را به سمت httphandler نشان می دهد



برنامه نویس هم میتواند با استفاده از اینترفیس‌های IHttpHandler و IHttpModule در درخواست‌ها دخالت نماید. برای شروع یک کلاس ایجاد کنید که اینترفیس IHttpModule را پیاده سازی می‌کند. شما دو متد را باید در این کلاس بنویسید؛ یکی Init و دیگر Dispose. همانطور که مطلع هستید، اولی در ابتدای ایجاد شیء و دیگر موقع از دست رفتن شی صدا زده می‌شود. متد Init یک آرگومان از نوع httpapplication دارد که مانند رسم نامگذاری متغیرها، بیشتر به اسم context یا app نام گذاری می‌شوند:

```
public void Init(HttpApplication app)
{
    app.BeginRequest += new EventHandler(OnBeginRequest);
}

public void Dispose() { }
```

همانطور که می‌بینید این شیء یک رویداد دارد که ما این رویداد را به تابعی به نام OnBeginRequest متصل کردیم. سایر رویدادهای موجود در httpapplication به شرح زیر می‌باشند:

این رویداد اولین رویدادی است که اجرا می‌شود، هر نوع عملی که میخواهید در ابتدای ارسال درخواست انجام دهید، باید در این قسمت قرار بگیرد؛ مثلاً قرار دادن یک بنر بالای صفحه	BeginRequest
خود دانت از یک سیستم امنیتی توکار بهره مند است و اگر می‌خواهید در مورد آن خصوصی سازی انجام بدھید، این رویداد می‌تواند کمکتان کند	AuthenticateRequest
بعد از رویداد بالا، این رویداد برای شناسایی انجام می‌شود. مثلاً دسترسی‌ها؛ دسترسی به قسمت‌هایی خاصی از منابع به او داده شود و قسمت‌هایی بعضی از منابع از او گرفته شود.	AuthorizeRequest
این رویداد برای کش کردن اطلاعات استفاده می‌شود. خود دانت تمامی این رویدادها را به صورت توکار فراهم آورده است؛ ولی اگر باز خصوصی سازی خاصی مد نظر شماست می‌توانید در این قسمت‌ها، تغییراتی را اعمال کنید. مثلاً ایجاد ... memory cache به جای file caching	ResolveRequestCache
این قسمت برای مدیریت state می‌باشد مثلاً مدیریت session	AcquireRequestState
این رویداد قبل از httpandler اجرا می‌شود.	PreRequestHandlerExecute
این رویداد بعد از httpandler اجرا می‌شود.	PostRequestHandlerExecute
این رویداد برای این صدا زده می‌شود که به شما بگوید عملیات درخواست پایان یافته است و باید state‌های ایجاد شده را release یا رها کنید.	ReleaseRequestState
برای خصوصی سازی output cache بکار می‌رود.	UpdateRequestCache
عملیات درخواست پایان یافته است. در صورتیکه قصد نوشتند دیباگری در طی تمامی عملیات دارید، میتواند به شما کمک کند.	EndRequest
این رویداد قبل از ارسال اطلاعات هدر هست. اگر قصد اضافه کردن اطلاعاتی به هدر دارید، این رویداد را به کار ببرید.	PreSendRequestHeaders
این رویداد موقعی صدا زده می‌شود که متد response.flush فراخوانی شود.. اگر می‌خواهید به محتوا چیزی اضافه کنید، از اینجا کمک بگیرید.	PreSendRequestContent

این رویداد اولین رویدادی است که اجرا می‌شود، هر نوع عملی که میخواهد در ابتدای ارسال درخواست انجام دهد، باید در این قسمت قرار بگیرد؛ مثلاً قرار دادن یک بنر بالای صفحه	BeginRequest
این رویداد موقعی رخ می‌دهد که یک استثنای مدیریت نشده رخ بدهد. برای نوشتن سیستم خطایابی خصوصی از این قسمت عمل کنید.	Error
این رویداد موقعی صدا زده میشود که درخواست، بنا به هر دلیلی پایان یافته است. برای عملیات پاکسازی و .. می‌شود از آن استفاده کرد. مثلاً یک جور rollback برای کارهای انجام گرفته.	Disposed

کد زیر را در نظر بگیرید:

کد زیر یک رویداد را تعریف کرده و سپس خود `httpapplication` را به عنوان `sender` استفاده می‌کند.

در اینجا قصد داریم یکی از صفحات را در خروجی تغییر دهیم. آدرس تایپ شده همان باشد ولی صفحه‌ی درخواست شده، صفحه‌ی دیگری است. این کار موقعی بیشتر کاربردی است که آدرس یک صفحه تغییر کرده و کاربر آدرس قبلی را وارد می‌کند. حالا یا از طریق بوک مارک یا از طریق یک لینک، در یک جای دیگر و شما میخواهید او را به صفحه‌ای جدید انتقال دهید، ولی در نوار آدرس، همان آدرس قبلی باقی بماند. همچنین کار دیگری که قرار است انجام بگیرد محاسبه مدت زمان رسیدگی به درخواست را محاسبه کند، برای همین در رویداد `BeginRequest` زمان شروع درخواست را ذخیره و در رویداد `EndRequest` با به دست آوردن اختلاف زمان فعلی با زمان شروع به مدت زمان مربوطه پی خواهیم برد.

با استفاده از `app.Context.Request.RawUrl` آدرس اصلی و درخواست شده را یافته و در صورتی که شامل نام صفحه مربوطه بود، با نام صفحه‌ی جدید جایجا می‌کنیم تا اطلاعات به صفحه‌ی جدید پاس شوند ولی در نوار آدرس، هنوز آدرس قبلی یا درخواست شده، قابل مشاهده است.

در خط `[app.Context.Items["start"]]` که یک کلاس ارث بری شده از اینترفیس `IDictionary` است، بر اساس کلید، داده شما را ذخیره و در موقع لزوم در هر رویداد به شما باز می‌گرداند.

```
public class UrlPath : IHttpModule
{
    public void Init(HttpApplication app)
    {
        app.BeginRequest+=new EventHandler(_BeginRequest);
        app.EndRequest+=new EventHandler(_EndRequest);
    }

    public void Dispose()
    {

    }

    void _BeginRequest(object sender, EventArgs e)
    {
        HttpApplication app = (HttpApplication) sender;
        app.Context.Items["start"] = DateTime.Now;

        if (app.Context.Request.RawUrl.ToLower().Contains("tours_list.aspx"))
        {
            app.Context.RewritePath(app.Context.Request.RawUrl.ToLower().Replace("tours_list.aspx", "tours_cat.aspx"));
        }
    }

    void _EndRequest(object sender, EventArgs e)
    {
        HttpApplication app = (HttpApplication)sender;
        string log = (DateTime.Now -
        DateTime.Parse(app.Context.Items["start"].ToString())).ToString();
        Debugger.Log(0,"duration","request took " + log+Environment.NewLine);
    }
}
```

}

حالا باید کلاس نوشته شده را به عنوان یک httpmodule به سیستم معرفی کنیم. به همین منظور وارد web.config شوید و کلاس جدید را معرفی کنید:

```
<httpModules>
  <add name="UrlPath" type="UrlPath"/>
</httpModules>
```

اگر کلاس شما داخل یک namespace.className قرار دارد، در قسمت type حتما قبل از نام کلاس، آن را تعریف کنید حالا دیگر کلاس UrlPath به عنوان یک httpmodule به سیستم معرفی شده است. تگ httpmodule را بین تگ <system.web> قرار داده ایم.

در ادامه پروژه را start بزنید تا نتیجه کار را ببینید:

اگر IIS شما، هم نسخه‌ی IIS من باشد، باید تفاوتی مشاهده کنید و می‌بینید که درخواست‌ها هیچ تغییری نکردند؛ چرا که اصلاً اجرا نشده است. در واقع در نسخه‌های قدیمی IIS یعنی 6 به قبل، این تعریف درست است ولی از نسخه‌ی 7 به بعد IIS، روش دیگری برای تعریف را قبول دارد و باید تگ <system.webserver>، بین دو تگ <httpModule> قرار بگیرد و نام تگ module به httpmodule تغییر پیدا کند.

پس کد فوق به این صورت تغییر می‌کند:

```
<system.webServer>
  <modules>
    <add name="UrlRewrite" type="UrlRewrite"/>
  </modules>
</system.webServer>
```

حالا اگر قصدا دارید که پروژه‌ی شما در هر دو IIS مورد حمایت قرار گیرد، باید این مازول را در هر دو جا معرفی کرده و در تگ system.webserver نیاز است تگ زیر تعریف شود که به طوری پیش فرض در webconfig می‌باشد:

```
<validation validateIntegratedModeConfiguration="false"/>
```

در غیر این صورت خطای زیر را دریافت می‌کنید:

در کل استفاده از این مازول به شما کمک می‌کند تمامی اطلاعات ارسالی به HTTP Error 500.22 - Internal Server Error سیستم را قبل از رسیدن به قسمت پردازش بررسی نمایید و هر نوع تغییری را که می‌خواهید اعمال کنید و لازم نیست این تغییرات را روی هر بخش، جداگانه انجام دهید یا یک کلاس بنویسید که هر بار در یک جا صدا بزنید و خیلی از موارد دیگر

HttpModule و Global.asax

اگر با global.asax کار کرده باشید حتما می‌پرسید که الان چه تفاوتی با httpmodule دارد؟ در فایل global هم همین‌ها را دارید و دقیقاً همین مزایا مهیا است؛ در واقع global.asax یک پیاده سازی از httpapplication هست. کلاس‌های httpmodule نام دیگری هم دارند به اسم Portable global.asax به معنی یک فایل قابل حمل یا پرتابل. دلیل این نام گذاری این هست که شما موقعی که یک کد را در فایل global می‌نویسید، برای همیشه آن کد متعلق به همان پروژه هست و قابل انتقال به یک پروژه دیگر نیست ولی شما میتوانید httpmodule‌ها را در قالب یک پروژه به هر پروژه‌ای که دوست دارید رفرنس کنید و کد شما قابلیت استفاده مجدد و Reuse پیدا می‌کند و هم اینکه در صورت نیاز می‌توانید آن‌ها را در قالب یک dll منتشر کنید.

نظرات خوانندگان

نویسنده: عباس حجتی
تاریخ: ۱۳۹۳/۰۹/۲۹ ۱۰:۲۴

با سلام؛ ممنون بابت مطلب خوبی که ارسال کردید.
من می‌خواهم از httpModule برای کنترل دسترسی کاربران (Authorization) استفاده کنم و به سشن کاربر نیاز دارم. مشکلی که هست موقع استفاده به این شکل `HttpContext.Current.Session` پیغام زیر را میده:
`Object not reference to instance of an object`
آیا برای استفاده از سشن راه خاصی هست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۲۹ ۱۴:۲۲

بله. در جدولی که تهیه کردم این مورد دقیقاً ذکر شده:
«AcquireRequestState» : این قسمت برای مدیریت `state` می‌باشد مثلاً مدیریت `session` به این معنا که سشن در تمام رویداد‌گردن‌های آن مهیا نیست. فقط تعدادی از آن‌ها دسترسی به سشن دارند. برای مثال:

```
public class SimpleModule : IHttpModule
{
    void IHttpModule.Init(HttpApplication application)
    {
        application.BeginRequest += new System.EventHandler(BeginRequest);
        application.AcquireRequestState += new EventHandler(application_AcquireRequestState);
    }

    public void BeginRequest(object sender, EventArgs e)
    {
        // no session here
    }

    void application_AcquireRequestState(object sender, EventArgs e)
    {
        HttpApplication app = sender as HttpApplication;
        app.Session.Add("Message", "hello module");
    }

    public void Dispose()
    {
    }
}
```

نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۳/۰۹/۲۹ ۱۴:۵۰

از رویداد `AcquireRequestState` استفاده کنید، همانطور که گفتم این رویداد برای مدیریت `state` هاست، عموماً در رویدادهای `ابتدایی` `session` هنوز ایجاد نشده است. کد زیر نمونه‌ای از دسترسی به `session` هاست :

```
public void Init(HttpApplication app)
{
    app.AcquireRequestState+=new EventHandler(_AUTH);
}

private void _AUTH(object sender, EventArgs e)
{
    HttpApplication app = (HttpApplication) sender;
    HttpContext context = app.Context;
    HttpSessionState session = context.Session;

    if(session!=null)
    {
        string text = "session is not exist";
    }
}
```

```
if (session["userid"] != null)
{
    text = "session is exist";
}
context.Response.Write(text);
}
```

توجه داشته باشید که خط if(session!=null) بسیار مهم هست و در صورت نبودن خط شرطی بعدی دچار خطایی که شما گرفتید می‌شود

نویسنده: عباس حجتی
تاریخ: ۱۳۹۳/۰۹/۲۹ ۱۷:۱۰

ممnon، درست شد.

یک مشکل دیگه اینکه درخواست‌های AJAX دیگه سمت سرور نمیره، می‌توانه به HttpModule ارتباطی داشته باشه؟

ممnon، این مشکل هم با فیلتر کردن درخواست‌های AJAX برطرف شد.

```
if (HttpContext.Current.Request.Headers["X-Requested-With"] != "XMLHttpRequest")
{ }
```

در [مقاله قبل](#) توضیح دادیم که وظیفه **httpHandler** رندر و پردازش خروجی یک درخواست هست؛ حالا در این مقاله قصد داریم **httpHandler** را بیشتر بررسی کنیم.

HttpHandler

برای تهیه‌ی یک **httpHandler**، باید کلاسی را بر اساس اینترفیس **IHttpHandler** پیاده سازی کنیم و بعدا آن را در **web.config** برنامه معرفی کنیم. برای پیاده سازی این اینترفیس، به یک متده به اسم **ProcessRequest** با یک پارامتر از نوع **HttpContext** و یک پرآپرتی به اسم **IsReusable** نیاز داریم که مقدار برگشتی این پرآپرتی را **false** بگذارید؛ بعدا خواهم گفت چرا اینکار را می‌کنیم. نحوه‌ی پیاده‌سازی یک **httpHandler** به شکل زیر است:

```
public class MyHttpHandler : IHttpHandler
{
  public void ProcessRequest(HttpContext context)
  {
  }

  public bool IsReusable
  {
    get { return false; }
  }
}
```

با استفاده از شیء **context** می‌توان به دو شیء **httprequest** و **httpresponse** دسترسی داشت. تکه کد زیر مثالی است در مورد نحوه‌ی تغییر در محتوای سایت:

```
public class MyHttpHandler : IHttpHandler
{
  public void ProcessRequest(HttpContext context)
  {
    HttpResponse response = context.Response;
    HttpRequest request = context.Request;

    response.Write("Every Page has a some text like this");
  }

  public bool IsReusable
  {
    get { return false; }
  }
}
```

بگذارید همین کد ساده را در وب کانفیگ معرفی کنیم:

```
<system.web>
  <httpHandlers>
    <add verb="*" path="*.aspx" type="MyHttpHandler"/>
  </httpHandlers>
</system.web>
```

اگر نسخه IIS شما همانند نسخه‌ی من باشد، نباید هیچ تغییری مشاهده کنید؛ زیرا کد بالا فقط در مورد نسخه‌ی IIS6 صدق می‌کند و برای نسخه‌های 7 IIS به بعد باید به شیوه زیر عمل کنید:

```
<configuration>
  <system.web>
    <httpHandlers>
      <add name="myhttpHandler" verb="*" path="*.aspx" type="MyHttpHandler"/>
    </httpHandlers>
  </system.web>
</configuration>
```

```
</httpHandlers>
</system.web>
</configuration>
```

خروجی نهایی باید تنها این متن باشد: Every Page has a some text like this

گزینه Type که نام کلاس می‌باشد و اگر کلاس داخل یک فضای نام قرار گرفته باشد، باید اینطور نوشت : namespace.ClassName
گزینه verb شامل مقادیری چون Delete,Get,Post,Head,Put را فقط برای این نوع درخواست‌ها اجرا می‌کند و در صورتیکه بخواهید چندتا از آن‌ها را استفاده کنید، با ، از هم جدا می‌شوند. مثلا Get,post و در صورتیکه همه‌ی گزینه‌ها را بخواهید علامت * را میتوان استفاده کرد.

گزینه‌ی path این امکان را به شما می‌دهد که مسیر و نوع فایل‌های را که قصد دارید روی آن‌ها فقط اجرا شود، مشخص کنید و ما در قطعه کد بالا گفته‌ایم که تنها روی فایل‌هایی با پسوند aspx اجرا شود و چون مسیری هم ذکر نکردیم برای همه‌ی مسیرها قابل اجراءست. یکی از مزیت‌های دادن پسوند این است که می‌توانید پسوندهای اختصاصی داشته باشید. مثلا پسوند RSS برای فیدهای وب سایتتان بسیاری از برنامه نویسان به جای استفاده از صفحات ashx از صفحات aspx استفاده می‌کنند که به مراتب سبک‌تر از aspx هست و شامل بخش url نمی‌شود و نتیجه خروجی آن بر اساس کدی که می‌نویسید مشخص می‌شود که می‌تواند صفحه متنی یا عکس یا xml یا ... باشد. در [اینجا](#) در مورد ساخت صفحات ashx توضیح داده شده است.

IHttpHandlerFactory

کار این اینترفیس پیاده سازی یک کلاس است که خروجی آن یک کلاس از نوع IHttpHandler هست. اگر دقت کنید در مثال‌های قبلی ما برای معرفی یک هندر در وب کافیگ یک سری path را به آن میدادیم و برای نمونه *.aspx را معرفی می‌کردیم؛ یعنی این هندر را بر روی همه‌ی فایل‌های aspx اجرا کن و اگر دو یا چند هندر در وب کافیگ معرفی کنیم و برای همه مسیر aspx را بدهیم، یعنی همه این هندرها باید روی صفحات aspx اجرا گردد و لی در IHttpHandlerFactory، ما چند هندر داریم و می‌خواهیم فقط یکی از آن‌ها بر روی صفحات aspx انجام بگیرد، پس ما یک هندر فکتوری را برای صفحات aspx معرفی می‌کنیم و در حین اجرا تصمیم می‌گیریم که کدام هندر را ارسال کنیم.

اجازه بدھید نوشتمن این نوع کلاس را آغاز کنیم، ابتدا دو هندر به نام‌های httpHandler1 و httpHandler2 می‌نویسیم :

```
public class MyHttpHandler1 : IHttpHandler
{
    public void ProcessRequest(HttpContext context)
    {
        HttpResponse response = context.Response;
        response.Write("this is httpHandler1");
    }

    public bool IsReusable
    {
        get { return false; }
    }
}

public class MyHttpHandler2 : IHttpHandler
{
    public void ProcessRequest(HttpContext context)
    {
        HttpResponse response = context.Response;
        response.Write("this is httpHandler2");
    }

    public bool IsReusable
    {
        get { return false; }
    }
}
```

سپس کلاس MyFactory را بر اساس اینترفیس IHttpFactory پیاده سازی می‌کنیم و باید دو متدهای آن صدابزنیم؛ یکی که هندر انتخابی را بر می‌گرداند و دیگری هم برای رها کردن یا آزادسازی یک هندر هست که در این مقاله کاری با آن نداریم. عموما

GC دات نت در این زمینه کارآیی خوبی دارد. در قسمت هندرهای غیرهمزمان به طور مختصر خواهیم گفت که GC چطور آن‌ها را مدیریت می‌کند. کد زیر نمونه کلاسی است که توسط IHttpFactory پیاده سازی شده است:

```
public class MyFactory : IHttpHandlerFactory
{
    public IHttpHandler GetHandler(HttpContext context, string requestType, string url, string
pathTrasnlated)
    {
    }

    public void ReleaseHandler(IHttpHandler handler)
    {
    }
}
```

در متدهای GetHandler و وجود دارند که به ترتیب برای موارد زیر به کار می‌روند:

Context
یک شی از کلاس HttpContext که دسترسی ما را برای اشیاء سروری چون response, request, session و ... فراهم می‌کند.
مشخص می‌کند که درخواست صفحه به چه صورتی است. این گزینه برای مواردی است که verb بیش از یک مورد را حمایت می‌کند. برای مثال دوست دارید یک هندر را برای درخواست‌های Get ارسال کنید و هندر دیگر را برای درخواست‌های نوع Post
مسیر مجازی virtual Path صفحه صدای زده شده
مسیر فیزیکی صفحه درخواست کننده را ارسال می‌کند.

متدهای GetHandler را بدین شکل می‌نویسیم و میخواهیم همه صفحات aspx هندر شماره یک را انتخاب کنند و صفحات aspx نامشان با t شروع می‌شوند، هندر شماره دو را انتخاب کند:

```
public IHttpHandler GetHandler(HttpContext context, string requestType, string url, string
pathTrasnlated)
{
    string handlername = "MyHttpHandler1";
    if(url.Substring(url.LastIndexOf("/") + 1).StartsWith("t"))
    {
        handlername = "MyHttpHandler2";
    }

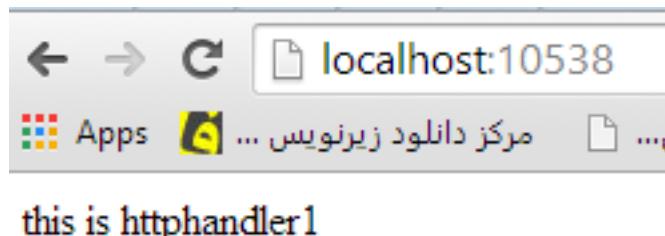
    try
    {
        return (IHttpHandler) Activator.CreateInstance(Type.GetType(handlername));
    }
    catch (Exception e)
    {
        throw new HttpException("Error: " + handlername, e);
    }
}

public void ReleaseHandler(IHttpHandler handler)
{}
```

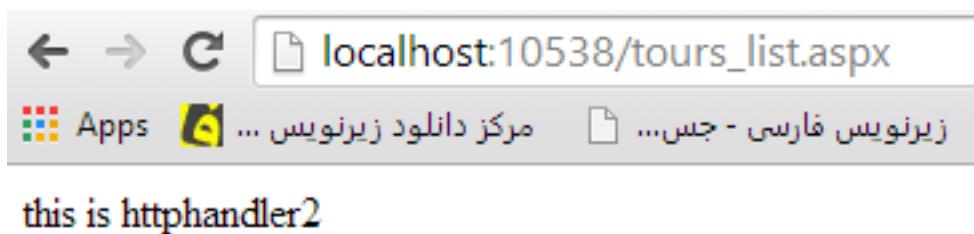
شی برای ساخت اشیاء با انتخاب بهترین constructor موجود بر اساس یک نوع Type مشخص به کار می‌رود و خروجی object را می‌گرداند؛ با یک تبدیل ساده، خروجی به قالب اصلی خود باز می‌گردد. برای مطالعه بیشتر در مورد این کلاس به [اینجا و اینجا](#) مراجعه کنید.

نحوه‌ی تعریف factory در وب کانفیگ مانند قبل است و فقط باید در Type به جای نام هندر نام فکتوری را نوشت. برنامه را اجرا

کنید تا نتیجه آن را ببینیم:
تصویر زیر نتیجه صدای زده شدن فایل default.aspx است:



تصویر زیر نتیجه صدای زده شدن فایل Tours_List.aspx است:



AsyncHttpHandlers

برای اینکه کار این اینترفیس را درک کنید بهتر هست [اینجا](#) را مطالعه کنید. در اینجا به خوبی تفاوت متدهای همزمان و غیرهمزمان توضیح داده شده است.

متن زیر خلاصه‌ترین و بهترین توضیح برای این پرسش است، چرا غیرهمزمان در اعمالی که disk I/O و یا network I/O دارند، پردازش موازی و اعمال async به شدت مقیاس پذیری سیستم را بالا می‌برند. به این ترتیب thread worker (که تعداد آن‌ها محدود است)، سریعتر آزاد شده و به worker pool بازگشت داده می‌شود تا بتواند به یک درخواست دیگر رسیده سرویس دهد. در این حالت می‌توان با منابع کمتری، درخواست‌های بیشتری را پردازش کرد. موقعی که اینترفیس IHttpHandler را ارث بری کنید (این اینترفیس نیز از IHttpHandler ارث بری کرده است و دو متده اضافه‌تر دارد)، باید دو متده دیگر را نیز پیاده سازی کنید:

```
public IAsyncResult BeginProcessRequest(HttpContext context, AsyncCallback callback, object obj)
{
}

public void EndProcessRequest(IAsyncResult result)
{}
```

پرآپریتی ISReusable هم موقعی که true برگشت بدهد، باعث می‌شود pooling فعال شده و این هندر در حافظه باقی بماند و تمامی درخواست‌ها از طریق همین یک نمونه اجرا شوند. به زبان ساده‌تر، این پرآپریتی می‌گوید اگر چندین درخواست از طرف کلاینت‌ها برسد، توسط یک نمونه یا instance از هندر پردازش خواهند شد؛ چون به طور پیش فرض موقعی که تمام درخواست‌های از pipeline بگذرند، هندرها توسط httpapplication در یک لیست بازیافت قرار گرفته و همه‌ی آن‌ها با null مقداردهی می‌شوند تا از حافظه پاک شوند ولی اگر این پرآپریتی true برگرداند، هندر مربوطه نال نشده و برای پاسخگویی به درخواست‌های بعدی در حافظه خواهد ماند. مهمترین مزیت این گزینه، این می‌باشد که کامپیوتر سیستم را بالا می‌برد و اشیا کمتری به GC پاس می‌شوند. ولی یک عیب هم دارد

که این تردھایی که ایجاد می‌کند، امنیت کمتری دارند و باید توسط برنامه نویس این امنیت بالاتر رود. این پراپرتی را در موقعی که با هندرهای همزمان کار می‌کنید برابر با `false` بگذارید چون این گزینه بیشتر بر روی هندرهای غیرهمزان اثر دارد و هم اینکه بعضی‌ها توصیه می‌کنند که بگذارید چون `GC` مدیریت خوبی در مورد هندرها دارد و هم این که ارزش یافتن باگ در کد را ندارد.

بر میگردیم سراغ کد نویسی هندر غیر همزمان. در آخرین قطعه کد نوشته شده، ما دو متدهای `BeginProcessRequest` است و خروجی آن کلاسی است که از اینترفیس `IAsyncResult` ارث بری کرده است. پس یک کلاس با ارث بری از این اینترفیس می‌نویسیم و در این کلاس نیاز است که ۴ پراپرتی را پیاده سازی کنیم که این کلاس به شکل زیر در خواهد آمد:

```
public class AsynchOperation : IAsyncResult
{
    private bool _completed;
    private Object _state;
    private AsyncCallback _callback;
    private HttpContext _context;

    bool IAsyncResult.IsCompleted { get { return _completed; } }
    WaitHandle IAsyncResult.AsyncWaitHandle { get { return null; } }
    Object IAsyncResult.AsyncState { get { return _state; } }
    bool IAsyncResult.CompletedSynchronously { get { return false; } }
}
```

متدهای `private` اجباری نیستند؛ ولی برای ذخیره مقادیر `get` و `set` نیاز است. همانطور که از اسمی آن‌ها پیداست مشخص است که برای چه کاری ساخته شده اند.
خب اجازه بدھید یک تابع سازنده به آن برای مقداردهی اولیه این متغیرهای خصوصی داشته باشیم:

```
public AsynchOperation(AsyncCallback callback, HttpContext context, Object state)
{
    _callback = callback;
    _context = context;
    _state = state;
    _completed = false;
}
```

همانطور که می‌بینید موارد موجود در متدهای `BeginProcessRequest` را تحویل می‌گیریم تا اطلاعات درخواستی مربوطه را داشته باشیم و مقدار `Completed` را هم برابر با `false` قرار می‌دهیم. سپس نوبت این می‌رسد که ما درخواست را در صف `pool` قرار دهیم. برای همین تکه کد زیر را اضافه می‌کنیم:

```
public void StartAsyncWork()
{
    ThreadPool.QueueUserWorkItem(new WaitCallback(StartAsyncTask),null);
}
```

با اضافه شدن درخواست به صف، هر موقع درخواست‌های قبلی تمام شوند و `callback` خودشان را ارسال کنند، نوبت درخواست‌های جدیدتر هم میرسد. `StartAsyncTask` هم متدهای است که وظیفه‌ی اصلی پردازش درخواست را به دوش دارد و موقعي که نوبت درخواست برسد، کدهای این متدها می‌گردد که ما در اینجا مانند مثال اول روی صفحه چیزی نوشتیم:

```
private void StartAsyncTask(Object workItemState)
{
    _context.Response.Write("<p>Completion IsThreadPoolThread is " +
    Thread.CurrentThread.IsThreadPoolThread + "</p>\r\n");

    _context.Response.Write("Hello World from Async Handler!");
    _completed = true;
    _callback(this);
}
```

دو خط اول اطلاعات را چاپ کرده و در خط سوم متغیر `completed` را `true` کرده و در آخر این درخواست را فراخوانی مجدد

می‌کنیم تا بگوییم که کار این درخواست پایان یافته است؛ پس این درخواست را از صف بیرون بکش و درخواست بعدی را اجرا کن. نهایتاً کل این کلاس را در متد BeginProcessRequest صدا بزنید:

```
context.Response.Write("<p>Begin IsThreadPoolThread is " + Thread.CurrentThread.IsThreadPoolThread + "</p>\r\n");
    AsynchOperation asynch = new AsynchOperation(callback, context, obj);
    asynch.StartAsyncWork();
    return asynch;
```

کل کد مربوطه : (توجه: کدها از داخل سایت msdn برداشته شده است و اکثر کدهای موجود در نت هم به همین قالب می‌نویسن)

```
public class MyHttpHandler : IHttpAsyncHandler
{
    public IAsyncResult BeginProcessRequest(HttpContext context, AsyncCallback callback, object obj)
    {
        context.Response.Write("<p>Begin IsThreadPoolThread is " +
Thread.CurrentThread.IsThreadPoolThread + "</p>\r\n");
        AsynchOperation asynch = new AsynchOperation(callback, context, obj);
        asynch.StartAsyncWork();
        return asynch;
    }

    public void EndProcessRequest(IAsyncResult result)
    {

    }
    public void ProcessRequest(HttpContext context)
    {
        throw new InvalidOperationException();
    }

    public bool IsReusable
    {
        get { return false; }
    }
}

public class AsynchOperation : IAsyncResult
{
    private bool _completed;
    private Object _state;
    private AsyncCallback _callback;
    private HttpContext _context;

    bool IAsyncResult.IsCompleted { get { return _completed; } }
    WaitHandle IAsyncResult.AsyncWaitHandle { get { return null; } }
    Object IAsyncResult.AsyncState { get { return _state; } }
    bool IAsyncResult.CompletedSynchronously { get { return false; } }

    public AsynchOperation(AsyncCallback callback, HttpContext context, Object state)
    {
        _callback = callback;
        _context = context;
        _state = state;
        _completed = false;
    }

    public void StartAsyncWork()
    {
        ThreadPool.QueueUserWorkItem(new WaitCallback(StartAsyncTask),null);
    }
    private void StartAsyncTask(Object workItemState)
    {
        _context.Response.Write("<p>Completion IsThreadPoolThread is " +
Thread.CurrentThread.IsThreadPoolThread + "</p>\r\n");

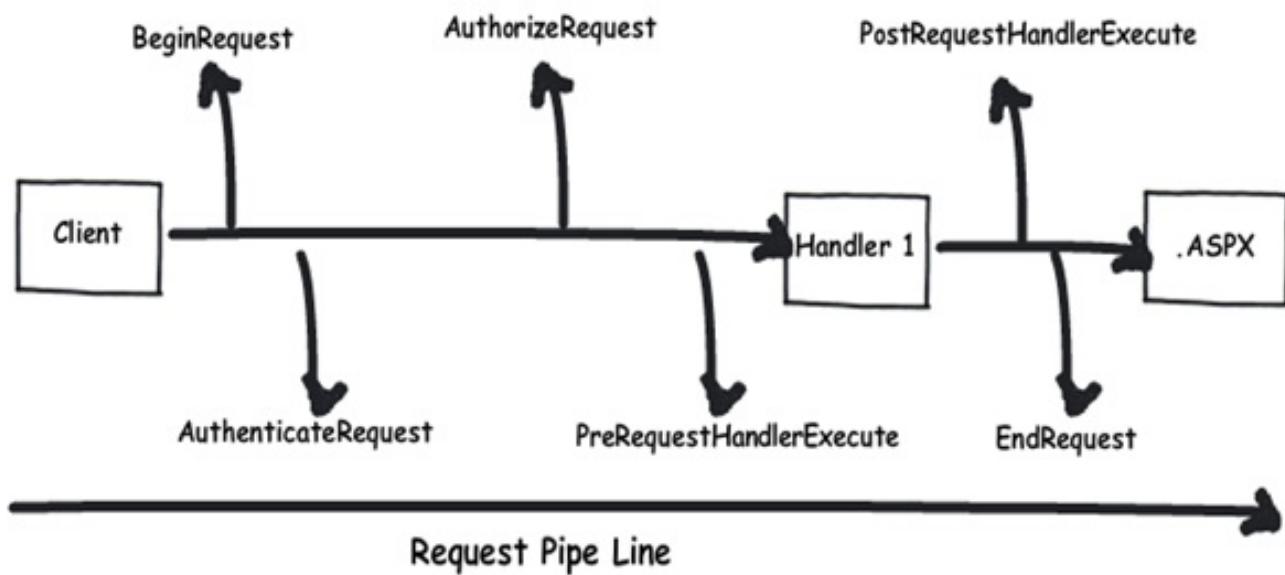
        _context.Response.Write("Hello World from Async Handler!");
        _completed = true;
        _callback(this);
    }
}
```

آشنایی با فایل ASHX

در مطالب بالاتر به فایل‌های Ashx اشاره کردیم. این فایل به نام Generic Web Handler شناخته می‌شوند و می‌توانید با Add New Item این نوع فایل‌ها را اضافه کنید. این فایل شامل هیچ UI ای نمی‌باشد و فقط شامل بخش کد می‌باشد. برای همین نسبت به سبک‌تر بوده و شامل یک directive به اسم `@WebHandler` است.

مايكروسافت در MSDN نوشته است که httpHandler در واقع فرآيندهای هستند (به اين فرآيندها بيشتر End Point می‌گويند) که در پاسخ به درخواست‌های رسیده شده توسط asp.net application اجرا می‌شوند و بيشترین درخواست‌های هم که می‌رسد از نوع صفحات Aspx می‌باشد و موقعی که كاربری درخواست صفحه‌ی aspx می‌کند هندرهای مربوط به page اجرا می‌شوند.

در متن بالا به خوبی روش نشست که دليل نداشتن UI، تعداد کمتری از handlerها را در مسیر Pipeline قرار می‌دهند و اجرای آن‌ها سريعتر است. غير از اين دو هندر aspx و ashx، هندر توکار دیگری چون asmx که مختص وب سرويس هست و axd مربوط به اعمال trace نيز وجود دارند.



created with Balsamiq Mockups - www.balsamiq.com

در [این لینک](#) که در بالاتر هم درج شده بود یک نمونه هندر برای نمایش تصویر نوشته است. اگر تصاویرتان را بدین صورت اجرا کنید می‌توان جلوی درخواست‌های رسیده از وب سایت‌های دیگر را سد کرد. برای مثال یک نفر مطالب شما را کپی می‌کند و در داخل و بلاگ یا وب سایتش می‌گذارد و شما در اينجا درخواست‌های رسیده خارج از وب سایت خود را لغو خواهید کرد و تصاویر کپی شده نمایش داده نخواهند شد.

پروژه‌ی [ASP.NET Identity](#) که نسل جدید سیستم Authentication و Authorization مخصوص ASP.NET است، دارای دو سری مثال رسمی است:

- الف) [مثال‌های کدپلکس](#)
- ب) [مثال نیوگت](#)

در ادامه قصد داریم مثال نیوگت آنرا که مثال کاملی است از نحوه‌ی استفاده از ASP.NET Identity در ASP.NET MVC، جهت اعمال الگوی واحد کار و تزریق وابستگی‌ها، بازنویسی کنیم.

پیش‌نیازها

- برای درک مطلب جاری نیاز است ابتدا [دوره‌ی مرتبطی را در سایت مطالعه کنید](#) و همچنین با [نحوه‌ی پیاده‌سازی الگوی واحد کار در EF Code First](#) آشنا باشید.
- به علاوه فرض بر این است که یک پروژه‌ی خالی 5 ASP.NET MVC را نیز آغاز کرده‌اید و توسط کنسول پاور شل نیوگت، فایل‌های مثال `Microsoft.AspNet.Identity.Samples` را به آن افزوده‌اید:

```
PM> Install-Package Microsoft.AspNet.Identity.Samples -Pre
```

ساختار پروژه‌ی تکمیلی

همانند مطلب [پیاده‌سازی الگوی واحد کار در EF Code First](#)، این پروژه‌ی جدید را با چهار اسمبلی class library دیگر به نام‌های

```
AspNetIdentityDependencyInjectionSample.DataLayer
AspNetIdentityDependencyInjectionSample.DomainClasses
AspNetIdentityDependencyInjectionSample.IocConfig
AspNetIdentityDependencyInjectionSample.ServiceLayer
```

تکمیل می‌کنیم.

ساختار پروژه‌ی `AspNetIdentityDependencyInjectionSample.DomainClasses`

مثال Microsoft.AspNet.Identity.Samples می‌توانید مشاهده کنید. در مطلب جاری، این نوع پیش‌فرض، به نوع `ApplicationUser` در فایل `Models\IdentityModels.cs` متداول int تغییر خواهد یافت. به همین جهت نیاز است کلاس‌های ذیل را به پروژه‌ی `DomainClasses` اضافه کرد:

```
using System.ComponentModel.DataAnnotations.Schema;
using Microsoft.AspNet.Identity.EntityFramework;

namespace AspNetIdentityDependencyInjectionSample.DomainClasses
{
    public class ApplicationUser : IdentityUser<int, CustomUserLogin, CustomUserRole, CustomUserClaim>
    {
        سایر خواص اضافی در اینجا //

        [ForeignKey("AddressId")]
        public virtual Address Address { get; set; }
        public int? AddressId { get; set; }
    }
}

using System.Collections.Generic;
```

```

namespace AspNetIdentityDependencyInjectionSample.DomainClasses
{
    public class Address
    {
        public int Id { get; set; }
        public string City { get; set; }
        public string State { get; set; }

        public virtual ICollection< ApplicationUser> ApplicationUsers { set; get; }
    }
}

using Microsoft.AspNet.Identity.EntityFramework;

namespace AspNetIdentityDependencyInjectionSample.DomainClasses
{
    public class CustomRole : IdentityRole<int, CustomUserRole>
    {
        public CustomRole() { }
        public CustomRole(string name) { Name = name; }

    }
}

using Microsoft.AspNet.Identity.EntityFramework;

namespace AspNetIdentityDependencyInjectionSample.DomainClasses
{
    public class CustomUserClaim : IdentityUserClaim<int>
    {

    }
}

using Microsoft.AspNet.Identity.EntityFramework;

namespace AspNetIdentityDependencyInjectionSample.DomainClasses
{
    public class CustomUserLogin : IdentityUserLogin<int>
    {

    }
}

using Microsoft.AspNet.Identity.EntityFramework;

namespace AspNetIdentityDependencyInjectionSample.DomainClasses
{
    public class CustomUserRole : IdentityUserRole<int>
    {

    }
}

```

در اینجا نحوه تغییر primary key از نوع `string` به نوع `int`, مشاهده می‌کنید. این تغییر نیاز به اعمال به کلاس‌های کاربران و همچنین نقش‌های آن‌ها نیز دارد. به همین جهت صرفاً تغییر کلاس ابتدایی `ApplicationUser` کافی نیست و باید کلاس‌های فوق را نیز اضافه کرد و تغییر داد.

بدیهی است در اینجا کلاس پایه کاربران را می‌توان سفارشی سازی کرد و خواص دیگری را نیز به آن افزود. برای مثال در اینجا یک کلاس جدید آدرس تعریف شده است که ارجاعی از آن در کلاس کاربران نیز قابل مشاهده است. سایر کلاس‌های مدل‌های اصلی برنامه که جداول بانک اطلاعاتی را تشکیل خواهند داد نیز در آینده به همین اسمبلی `DomainClasses` اضافه می‌شوند.

ساختار پروژه‌ی `AspNetIdentityDependencyInjectionSample.DataLayer` جهت اعمال الگوی واحد کار

اگر به همان فایل `Models\IdentityModels.cs` ابتدایی پروژه که اکنون کلاس `ApplicationUser` آن را به پروژه‌ی `DomainClasses` منتقل کرده‌ایم، مجدداً مراجعه کنید، کلاس `DbContext` مخصوص `ASP.NET Identity` نیز در آن تعریف شده است:

```
public class ApplicationDbContext : IdentityDbContext< ApplicationUser >
```

این کلاس را به پروژه‌ی DataLayer منتقل می‌کنیم و از آن به عنوان DbContext اصلی برنامه استفاده خواهیم کرد. بنابراین دیگر نیازی نیست چندین DbContext در برنامه داشته باشیم. در اصل از IdentityDbContext مشتق شده است. اینترفیس IUnitOfWork برنامه، در پروژه‌ی DataLayer چنین شکلی را دارد که نمونه‌ای از آن را در مطلب آشنایی با نحوه‌ی پیاده سازی الگوی واحد کار در EF Code First، پیشتر ملاحظه کرده‌اید.

```
using System.Collections.Generic;
using System.Data.Entity;

namespace AspNetIdentityDependencyInjectionSample.DataLayer.Context
{
    public interface IUnitOfWork
    {
        IDbSet< TEntity > Set< TEntity >() where TEntity : class;
        int SaveAllChanges();
        void MarkAsChanged< TEntity >( TEntity entity ) where TEntity : class;
        IList< T > GetRows< T >( string sql, params object[] parameters ) where T : class;
        IEnumerable< TEntity > AddThisRange< TEntity >( IEnumerable< TEntity > entities ) where TEntity : class;
        void ForceDatabaseInitialize();
    }
}
```

اکنون کلاس ApplicationDbContext منتقل شده به DataLayer یک چنین امضایی را خواهد یافت:

```
public class ApplicationDbContext :
    IdentityDbContext< ApplicationUser, CustomRole, int, CustomUserLogin, CustomUserRole,
    CustomUserClaim >,
    IUnitOfWork
{
    public DbSet< Category > Categories { set; get; }
    public DbSet< Product > Products { set; get; }
    public DbSet< Address > Addresses { set; get; }
```

تعريف آن باید جهت اعمال کلاس‌های سفارشی سازی شده‌ی کاربران و نقش‌های آن‌ها برای استفاده از نوع int primary key از نوع به شکل فوق، تغییر یابد. همچنین در انتهای آن مانند قبل، IUnitOfWork نیز ذکر شده است. پیاده سازی کامل این کلاس را از پروژه‌ی پیوست انتهای بحث می‌توانید دریافت کنید. کار کردن با این کلاس، هیچ تفاوتی با DbContext‌های متدالوول EF Code First ندارد و تمام اصول آن‌ها یکی است.

در ادامه اگر به فایل App_Start\IdentityConfig.cs مراجعه کنید، کلاس ذیل در آن قابل مشاهده است:

```
public class ApplicationDbContextInitializer : DropCreateDatabaseIfModelChanges< ApplicationDbContext >
```

نیازی به این کلاس به این شکل نیست. آن را حذف کنید و در پروژه‌ی DataLayer، کلاس جدید ذیل را اضافه نمایید:

```
using System.Data.Entity.Migrations;
namespace AspNetIdentityDependencyInjectionSample.DataLayer.Context
{
    public class Configuration : DbMigrationsConfiguration< ApplicationDbContext >
    {
        public Configuration()
        {
            AutomaticMigrationsEnabled = true;
            AutomaticMigrationDataLossAllowed = true;
        }
    }
}
```

در این مثال، بحث migrations به حالت خودکار تنظیم شده است و تمام تغییرات در پروژه‌ی DomainClasses را به صورت خودکار به بانک اطلاعاتی اعمال می‌کند. تا همینجا کار تنظیم DataLayer به پایان می‌رسد.

ساختمان پروژه‌ی AspNetIdentityDependencyInjectionSample.ServiceLayer

در ادامه مابقی کلاس‌های موجود در فایل App_Start\IdentityConfig.cs را به لایه سرویس برنامه منتقل خواهیم کرد. همچنین برای آن‌ها یک سری اینترفیس جدید نیز تعریف می‌کنیم، تا تزریق وابستگی‌ها به نحو صحیحی صورت گیرد. اگر به فایل‌های کنترلر این مثال پیش فرض مراجعه کنید (پیش از تغییرات بحث جاری)، هرچند به نظر در کنترلرها، کلاس‌های موجود در فایل App_Start\IdentityConfig.cs تزریق شده‌اند، اما به دلیل عدم استفاده از اینترفیس‌ها، وابستگی کاملی بین جزئیات پیاده سازی این کلاس‌ها و نمونه‌های تزریق شده به کنترلرها وجود دارد و عملاً معکوس سازی واقعی وابستگی‌ها رخ نداده است. بنابراین نیاز است این مسایل را اصلاح کنیم.

الف) انتقال کلاس ApplicationUserManager به لایه سرویس برنامه

کلاس ApplicationUserManager را به لایه سرویس منتقل می‌کنیم:

```
using System;
using System.Security.Claims;
using System.Threading.Tasks;
using AspNetIdentityDependencyInjectionSample.DomainClasses;
using AspNetIdentityDependencyInjectionSample.ServiceLayer.Contracts;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
using Microsoft.Owin.Security.Cookies;
using Microsoft.Owin.Security.DataProtection;

namespace AspNetIdentityDependencyInjectionSample.ServiceLayer
{
    public class ApplicationUserManager
        : UserManager<ApplicationUser, int>, IApplicationUserManager
    {
        private readonly IDataProtectionProvider _dataProtectionProvider;
        private readonly IIdentityMessageService _emailService;
        private readonly IApplicationRoleManager _roleManager;
        private readonly IIdentityMessageService _smsService;
        private readonly IUserStore<ApplicationUser, int> _store;

        public ApplicationUserManager(IUserStore<ApplicationUser, int> store,
            IApplicationRoleManager roleManager,
            IDataProtectionProvider dataProtectionProvider,
            IIdentityMessageService smsService,
            IIdentityMessageService emailService)
            : base(store)
        {
            _store = store;
            _roleManager = roleManager;
            _dataProtectionProvider = dataProtectionProvider;
            _smsService = smsService;
            _emailService = emailService;
        }

        protected void CreateApplicationUserManager();
    }

    public void SeedDatabase()
    {
    }

    private void CreateApplicationUserManager()
    {
        // Configure validation logic for usernames
        this.UserValidator = new UserValidator<ApplicationUser, int>(this)
        {
            AllowOnlyAlphanumericUserNames = false,
            RequireUniqueEmail = true
        };

        // Configure validation logic for passwords
        this.PasswordValidator = new PasswordValidator
        {
            RequiredLength = 6,
            RequireNonLetterOrDigit = true,
            RequireDigit = true,
            RequireLowercase = true,
            RequireUppercase = true,
        };
    }
}
```

```
// Configure user lockout defaults
this.UserLockoutEnabledByDefault = true;
this.DefaultAccountLockoutTimeSpan = TimeSpan.FromMinutes(5);
this.MaxFailedAccessAttemptsBeforeLockout = 5;

// Register two factor authentication providers. This application uses Phone and Emails as a step
of receiving a code for verifying the user
// You can write your own provider and plug in here.
this.RegisterTwoFactorProvider("PhoneCode", new PhoneNumberTokenProvider<ApplicationUser, int>
{
    MessageFormat = "Your security code is: {0}"
});
this.RegisterTwoFactorProvider("EmailCode", new EmailTokenProvider<ApplicationUser, int>
{
    Subject = "SecurityCode",
    BodyFormat = "Your security code is {0}"
});
this.EmailService = _emailService;
this.SmsService = _smsService;

if (_dataProtectionProvider != null)
{
    var dataProtector = _dataProtectionProvider.Create("ASP.NET Identity");
    this.UserTokenProvider = new DataProtectorTokenProvider<ApplicationUser, int>(dataProtector);
}
}
```

تغییراتی که در اینجا اعمال شده‌اند، به شرح زیر می‌باشند:

- متد استاتیک Create این کلاس حذف و تعاریف آن به سازنده‌ی کلاس منتقل شده‌اند. به این ترتیب با هر بار وله سازی این کلاس توسط IoC Container به صورت خودکار این تنظیمات نیز به کلاس پایه UserManager اعمال می‌شوند.
 - اگر به کلاس پایه UserManager دقت کنید، به آرگومان‌های جنریک آن یک int هم اضافه شده‌است. این مورد جهت استفاده از primary key از نوع int ضروری است.
 - در کلاس پایه UserManager تعدادی متد وجود دارند. تعاریف آن‌ها را به اینترفیس ApplicationUserManager منتقل خواهیم کرد. نیازی هم به پیاده سازی این متد‌ها در کلاس جدید ApplicationUserManager نیست؛ زیرا کلاس پایه UserManager پیشتر آن‌ها را پیاده سازی کرده‌است. به این ترتیب می‌توان به یک تزریق وابستگی واقعی و بدون وابستگی به پیاده سازی خاص سازی آن را جهت آزمون‌های واحد خواهد یافت.
 - در کلاس اصلی ApplicationDbContextInitializer پیش فرض این مثال، متد Seed هم قابل مشاهده‌است. این متد را از کلاس جدید Configuration اضافه شده به DataLayer حذف کرده‌ایم. از این جهت که در آن از متد‌های کلاس ApplicationUserManager مستقیماً استفاده شده‌است. متد Seed اکنون به کلاس جدید اضافه شده به لایه سرویس منتقل شده و در آغاز برنامه فراخوانی خواهد شد. DataLayer نباید وابستگی به لایه سرویس داشته باشد. لایه سرویس است که از امکانات DataLayer استفاده می‌کند.
 - اگر به سازنده‌ی کلاس جدید ApplicationUserManager دقت کنید، چند اینترفیس دیگر نیز به آن تزریق شده‌اند. اینترفیس IApplicationRoleManager را ادامه تعریف خواهیم کرد. سایر اینترفیس‌های تزریق شده مانند IIdentityMessageService و IDataProtectionProvider آن‌ها نیست. فقط کلاس‌های SmsService و EmailService و App_Start\IdentityConfig فایل c مربوط به لایه سرویس منتقل کرده‌ایم. این کلاس‌ها بر اساس تنظیمات IoC Container مورد استفاده، در اینجا به صورت خودکار ترزیق خواهند شد. حالت پیش فرض آن، وله سازی مستقیم است که مطابق کدهای فوق به حالت تزریق وابستگی‌ها پیمود یافته‌است.

ب) انتقال کلاس ApplicationSignInManager به لایه سرویس برنامه

کلاس ApplicationSignInManager فایل Start\IdentityConfig.c می کنیم.

```
using AspNetIdentityDependencyInjectionSample.DomainClasses;
using AspNetIdentityDependencyInjectionSample.ServiceLayer.Contracts;
using Microsoft.AspNet.Identity.Owin;
using Microsoft.Owin.Security;

namespace AspNetIdentityDependencyInjectionSample.ServiceLayer
{
    public class ApplicationSignInManager :
```

```

    SignInManager<ApplicationUser, int>, IApplicationSignInManager
{
    private readonly ApplicationUserManager _userManager;
    private readonly IAuthenticationManager _authenticationManager;

    public ApplicationSignInManager(ApplicationUserManager userManager,
        IAuthenticationManager authenticationManager) :
        base(userManager, authenticationManager)
    {
        _userManager = userManager;
        _authenticationManager = authenticationManager;
    }
}

```

در اینجا نیز اینترفیس جدید `IApplicationSignInManager` را برای مخفی سازی پیاده سازی کلاس پایه توکار `SignInManager` اضافه کرده‌ایم. این اینترفیس دقیقاً حاوی تعاریف متدهای کلاس پایه `SignInManager` است و نیازی به پیاده سازی مجدد در کلاس `ApplicationSignInManager` نخواهد داشت.

ج) انتقال کلاس ApplicationRoleManager به لایه سرویس برنامه
کلاس `ApplicationRoleManager` را نیز به لایه سرویس منتقل خواهیم کرد:

```

using AspNetIdentityDependencyInjectionSample.DomainClasses;
using AspNetIdentityDependencyInjectionSample.ServiceLayer.Contracts;
using Microsoft.AspNet.Identity;

namespace AspNetIdentityDependencyInjectionSample.ServiceLayer
{
    public class ApplicationRoleManager : RoleManager<CustomRole, int>, IApplicationRoleManager
    {
        private readonly IRoleStore<CustomRole, int> _roleStore;
        public ApplicationRoleManager(IRoleStore<CustomRole, int> roleStore)
            : base(roleStore)
        {
            _roleStore = roleStore;
        }

        public CustomRole FindRoleByName(string roleName)
        {
            return this.FindByName(roleName); // RoleManagerExtensions
        }

        public IdentityResult CreateRole(CustomRole role)
        {
            return this.Create(role); // RoleManagerExtensions
        }
    }
}

```

روش کار نیز در اینجا همانند دو کلاس قبل است. اینترفیس جدید `IApplicationRoleManager` را که حاوی تعاریف متدهای کلاس `ApplicationRoleManager` است، به لایه سرویس اضافه می‌کنیم. کنترلرهای برنامه با این اینترفیس بجای استفاده مستقیم از کلاس `ApplicationRoleManager` کار خواهند کرد.

تا اینجا کار تنظیمات لایه سرویس برنامه به پایان می‌رسد.

ساختمان پروژه‌ی `AspNetIdentityDependencyInjectionSample.IocConfig`

پروژه‌ی `IocConfig` جایی است که تنظیمات `StructureMap` را به آن منتقل کرده‌ایم:

```

using System;
using System.Data.Entity;
using System.Threading;
using System.Web;
using AspNetIdentityDependencyInjectionSample.DataLayer.Context;

```

```

using AspNetIdentityDependencyInjectionSample.DomainClasses;
using AspNetIdentityDependencyInjectionSample.ServiceLayer;
using AspNetIdentityDependencyInjectionSample.ServiceLayer.Contracts;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.Owin.Security;
using StructureMap;
using StructureMap.Web;

namespace AspNetIdentityDependencyInjectionSample.IocConfig
{
    public static class SmObjectFactory
    {
        private static readonly Lazy<Container> _containerBuilder =
            new Lazy<Container>(defaultContainer, LazyThreadSafetyMode.ExecutionAndPublication);

        public static IContainer Container
        {
            get { return _containerBuilder.Value; }
        }

        private static Container defaultContainer()
        {
            return new Container(ioc =>
            {
                ioc.For<IUnitOfWork>()
                    .HybridHttpOrThreadLocalScoped()
                    .Use<ApplicationDbContext>();

                ioc.For<ApplicationDbContext>().HybridHttpOrThreadLocalScoped().Use<ApplicationDbContext>();
                ioc.For<DbContext>().HybridHttpOrThreadLocalScoped().Use<ApplicationDbContext>();

                ioc.For<IUserStore<ApplicationUser, int>>()
                    .HybridHttpOrThreadLocalScoped()
                    .Use<UserStore<ApplicationUser, CustomRole, int, CustomUserLogin, CustomUserRole, CustomUserClaim>>();

                ioc.For<IRoleStore<CustomRole, int>>()
                    .HybridHttpOrThreadLocalScoped()
                    .Use<RoleStore<CustomRole, int, CustomUserRole>>();

                ioc.For<IAuthenticationManager>()
                    .Use(() => HttpContext.Current.GetOwinContext().Authentication);

                ioc.For<IAplicationSignInManager>()
                    .HybridHttpOrThreadLocalScoped()
                    .Use<AplicationSignInManager>();

                ioc.For<IAplicationUserManager>()
                    .HybridHttpOrThreadLocalScoped()
                    .Use<AplicationUserManager>();

                ioc.For<IAplicationRoleManager>()
                    .HybridHttpOrThreadLocalScoped()
                    .Use<AplicationRoleManager>();

                ioc.For<IIIdentityMessageService>().Use<SmsService>();
                ioc.For<IIIdentityMessageService>().Use<EmailService>();
                ioc.For<ICustomRoleStore>()
                    .HybridHttpOrThreadLocalScoped()
                    .Use<CustomRoleStore>();

                ioc.For<ICustomUserStore>()
                    .HybridHttpOrThreadLocalScoped()
                    .Use<CustomUserStore>();

                //config.For<IDataProtectionProvider>().Use(()=> app.GetDataProtectionProvider()); // In Startup class
            });
        }
    }
}

```

در اینجا نحوه اتصال اینترفیس‌های برنامه را به کلاس‌ها و یا نمونه‌هایی که آن‌ها را می‌توانند پیاده سازی کنند، مشاهده می‌کنید.

برای مثال `ApplicationDbContext` به `IUnitOfWork` مرتبط شده است و یا دوبار تعاریف متناظر با `DbContext` را مشاهده می‌کنید. از این تعاریف به صورت توکار توسط `ASP.NET Identity` زمانیکه قرار است `UserStore` و `RoleStore` را وله سازی کند، استفاده می‌شوند و ذکر آن‌ها الزامی است.

در تعاریف فوق، یک مورد را به فایل `Startup.cs` موقول کرده‌ایم. برای مشخص سازی نمونه‌ی پیاده سازی کننده در `Startup.cs` نیاز است به `IAppBuilder` کلاس `IDataProtectionProvider` برنامه دسترسی داشت. این کلاس آغازین `Owin` اکنون به `CreatePerOwinContext` نحو ذیل بازنویسی شده است و در آن، تنظیمات `IDataProtectionProvider` را به همراه وله سازی مشاهده می‌کنید:

```
using System;
using AspNetIdentityDependencyInjectionSample.IocConfig;
using AspNetIdentityDependencyInjectionSample.ServiceLayer.Contracts;
using Microsoft.AspNet.Identity;
using Microsoft.Owin;
using Microsoft.Owin.Security.Cookies;
using Microsoft.Owin.Security.DataProtection;
using Owin;
using StructureMap.Web;

namespace AspNetIdentityDependencyInjectionSample
{
    public class Startup
    {
        public void Configuration(IAppBuilder app)
        {
            configureAuth(app);
        }

        private static void configureAuth(IAppBuilder app)
        {
            SmObjectFactory.Container.Configure(config =>
            {
                config.For<IDataProtectionProvider>()
                    .HybridHttpOrThreadLocalScoped()
                    .Use(()=> app.GetDataProtectionProvider());
            });
            SmObjectFactory.Container.GetInstance<IApplicationUserManager>().SeedDatabase();

            // Configure the db context, user manager and role manager to use a single instance per
request
            app.CreatePerOwinContext(() =>
SmObjectFactory.Container.GetInstance<IApplicationUserManager>());

            // Enable the application to use a cookie to store information for the signed in user
            // and to use a cookie to temporarily store information about a user logging in with a
third party login provider
            // Configure the sign in cookie
            app.UseCookieAuthentication(new CookieAuthenticationOptions
            {
                AuthenticationType = DefaultAuthenticationTypes.ApplicationCookie,
                LoginPath = new PathString("/Account/Login"),
                Provider = new CookieAuthenticationProvider
                {
                    // Enables the application to validate the security stamp when the user logs in.
                    // This is a security feature which is used when you change a password or add an
external login to your account.
                    OnValidateIdentity =
SmObjectFactory.Container.GetInstance<IApplicationUserManager>().OnValidateIdentity()
                }
            });
            app.UseExternalSignInCookie(DefaultAuthenticationTypes.ExternalCookie);

            // Enables the application to temporarily store user information when they are verifying
the second factor in the two-factor authentication process.
            app.UseTwoFactorSignInCookie(DefaultAuthenticationTypes.TwoFactorCookie,
TimeSpan.FromMinutes(5));

            // Enables the application to remember the second login verification factor such as phone
or email.
            // Once you check this option, your second step of verification during the login process
will be remembered on the device where you logged in from.
            // This is similar to the RememberMe option when you log in.

            app.UseTwoFactorRememberBrowserCookie(DefaultAuthenticationTypes.TwoFactorRememberBrowserCookie);
        }
    }
}
```

}

این تعاریف از فایل پیش فرض App_Start.cs دریافت و جهت کار با IoC Container برنامه، بازنویسی شده‌اند.

تنظیمات برنامه‌ی اصلی ASP.NET MVC، جهت اعمال تزریق وابستگی‌ها

الف) ابتدا نیاز است فایل Global.asax.cs را به نحو ذیل بازنویسی کنیم:

```
using System;
using System.Data.Entity;
using System.Web;
using System.Web.Mvc;
using System.Web.Optimization;
using System.Web.Routing;
using AspNetIdentityDependencyInjectionSample.DataLayer.Context;
using AspNetIdentityDependencyInjectionSample.IocConfig;
using StructureMap.Web.Pipeline;

namespace AspNetIdentityDependencyInjectionSample
{
    public class MvcApplication : HttpApplication
    {
        protected void Application_Start()
        {
            AreaRegistration.RegisterAllAreas();
            FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            BundleConfig.RegisterBundles(BundleTable.Bundles);

            setDbInitializer();
            //Set current Controller factory as StructureMapControllerFactory
            ControllerBuilder.Current.SetControllerFactory(new StructureMapControllerFactory());
        }

        protected void Application_EndRequest(object sender, EventArgs e)
        {
            HttpContextLifecycle.DisposeAndClearAll();
        }

        public class StructureMapControllerFactory : DefaultControllerFactory
        {
            protected override IController GetControllerInstance(RequestContext requestContext, Type controllerType)
            {
                if (controllerType == null)
                    throw new InvalidOperationException(string.Format("Page not found: {0}", requestContext.HttpContext.Request.RawUrl));
                return SmObjectFactory.Container.GetInstance(controllerType) as Controller;
            }
        }

        private static void setDbInitializer()
        {
            Database.SetInitializer(new MigrateDatabaseToLatestVersion<ApplicationDbContext, Configuration>());
            SmObjectFactory.Container.GetInstance<IUnitOfWork>().ForceDatabaseInitialize();
        }
    }
}
```

در اینجا در متدها setDbInitializer، نحوه‌ی استفاده و تعریف فایل Configuration Data لایه را ملاحظه می‌کنید؛ به همراه متدهای آغاز بانک اطلاعاتی و اعمال تغییرات لازم به آن در ابتدای کار برنامه. همچنین ControllerFactory برنامه نیز به تنظیم شده‌است تا کار تزریق وابستگی‌ها به کنترلرهای برنامه به صورت خودکار میسر شود. در پایان کار هر درخواست نیز منابع Disposable رها می‌شوند.

ب) به پوشه‌ی Models برنامه مراجعه کنید. در اینجا در هر کلاسی که Id از نوع string وجود داشت، باید تبدیل به نوع int

شوند. چون primary key برنامه را به نوع int تغییر داده‌ایم. برای مثال کلاس‌های EditUserViewModel و RoleViewModel باید تغییر کنند.

ج) اصلاح کنترلرهای برنامه جهت اعمال تزریق وابستگی‌ها

اکنون اصلاح کنترلرهای اعمال تزریق وابستگی‌ها ساده‌است. در ادامه نحوه تغییر امضای سازنده‌های این کنترلرهای جهت استفاده از اینترفیس‌های جدید مشاهده می‌کنید:

```
[Authorize]
public class AccountController : Controller
{
    private readonly IAuthenticationManager _authenticationManager;
    private readonly IApplicationSignInManager _signInManager;
    private readonly IApplicationUserManager _userManager;
    public AccountController(IApplicationUserManager userManager,
                            IApplicationSignInManager signInManager,
                            IAuthenticationManager authenticationManager)
    {
        _userManager = userManager;
        _signInManager = signInManager;
        _authenticationManager = authenticationManager;
    }

    [Authorize]
    public class ManageController : Controller
    {
        // Used for XSRF protection when adding external logins
        private const string XsrfKey = "XsrfId";

        private readonly IAuthenticationManager _authenticationManager;
        private readonly IApplicationUserManager _userManager;
        public ManageController(IApplicationUserManager userManager, IAuthenticationManager authenticationManager)
        {
            _userManager = userManager;
            _authenticationManager = authenticationManager;
        }

        [Authorize(Roles = "Admin")]
        public class RolesAdminController : Controller
        {
            private readonly IApplicationRoleManager _roleManager;
            private readonly IApplicationUserManager _userManager;
            public RolesAdminController(IApplicationUserManager userManager,
                                       IApplicationRoleManager roleManager)
            {
                _userManager = userManager;
                _roleManager = roleManager;
            }

            [Authorize(Roles = "Admin")]
            public class UsersAdminController : Controller
            {
                private readonly IApplicationRoleManager _roleManager;
                private readonly IApplicationUserManager _userManager;
                public UsersAdminController(IApplicationUserManager userManager,
                                           IApplicationRoleManager roleManager)
                {
                    _userManager = userManager;
                    _roleManager = roleManager;
                }
            }
        }
    }
}
```

پس از این تغییرات، فقط کافی است بجای خواص برای مثال RoleManager سابق از فیلد‌های تزریق شده در کلاس، مثلا _roleManager جدید استفاده کرد. امضای متدهای یکی است و تنها به یک search و replace نیاز دارد. البته تعدادی اکشن متدهای id string وجود دارند که از int استفاده می‌کنند. این‌ها را باید به Id? تغییر داد تا با نوع جدید مورد استفاده طباق پیدا کنند.

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید:

نظرات خوانندگان

نویسنده: سیروان عفیفی
تاریخ: ۱۶:۱۱ ۱۳۹۳/۱۰/۰۵

ممنون از مطلب شما!

برای تغییر نام جداول تشکیل شده، درون متد OnModelCreating اما با بروز رسانی دیتابیس تغییری در اسمی جداول حاصل نشد:

```
modelBuilder.Entity<ApplicationUser>().ToTable("User").Property(p => p.Id).HasColumnName("Id");
modelBuilder.Entity<CustomUserRole>().ToTable("UserRole").HasKey(p => new { p.RoleId,
p.UserId });
modelBuilder.Entity<CustomUserLogin>().ToTable("UserLogin").HasKey(p => new {
p.LoginProvider, p.ProviderKey, p.UserId });
modelBuilder.Entity<CustomUserClaim>().ToTable("UserClaim").HasKey(p => p.Id).Property(p =>
p.Id).HasColumnName("UserClaimId");
modelBuilder.Entity<CustomRole>().ToTable("Role").Property(p =>
p.Id).HasColumnName("RoleId");
```

نویسنده: وحید نصیری
تاریخ: ۱۸:۵۳ ۱۳۹۳/۱۰/۰۵

این روش پس از امتحان، [جواب داد](#):

```
protected override void OnModelCreating(DbModelBuilder builder)
{
    base.OnModelCreating(builder);

    builder.Entity< ApplicationUser >().ToTable("Users");
    builder.Entity< CustomRole >().ToTable("Roles");
    builder.Entity< CustomUserClaim >().ToTable("UserClaims");
    builder.Entity< CustomUserRole >().ToTable("UserRoles");
    builder.Entity< CustomUserLogin >().ToTable("UserLogins");
}
```

اگر متد `base.OnModelCreating(builder)` را در انتهای کار قرار دهید، تنظیمات پیش فرض کلاس پایه `IdentityDbContext` (یعنی همان [نام‌های قدیمی](#)) اعمال می‌شوند و تنظیمات شما بازنویسی خواهند شد. این متد باید در ابتدای کار فراخوانی شود.

نویسنده: فخاری
تاریخ: ۱۹:۲۱ ۱۳۹۳/۱۰/۰۸

با سلام

ممنون از مطلب مفید شما. واقعاً عالی بود

ولی قسمتی از کد برام نامفهوم بود:

```
public ApplicationDbContext()
    : base("connectionString1")
{
    //this.Database.Log = data => System.Diagnostics.Debug.WriteLine(data);
    // فقط تعریف شده تا یک برک پوینت در اینجا قرار داده شود برای آزمایش تعداد بار فراخوانی آن//
}

protected override void Dispose(bool disposing)
{
    base.Dispose(disposing);
    // فقط تعریف شده تا یک برک پوینت در اینجا قرار داده شود برای آزمایش فراخوانی آن//
```

مخصوصاً بخش `.Dispose`

چون قبل از الگوی واحد کار استفاده می‌شد این بخش وجود نداشت !
آیا وجود این کدها که در بالا اومده الزامی است؟

نویسنده: وحید نصیری
تاریخ: ۱۹:۲۴ ۱۳۹۳/۱۰/۰۸

خیر. متدهای `Dispose` را حذف کنید (در کلاس پایه هست). وجود سازنده هم صرفاً جهت ساده سازی تعریف و انتخاب رشته اتصالی تعريف شده در `web.config` است.

نویسنده: وحید نصیری
تاریخ: ۱۳:۲۷ ۱۳۹۳/۱۰/۱۱

نکته‌ای مهم در مورد مدیریت استراکچرمپ در این مثال اگر از `SmObjectFactory` مطلب فوق استفاده می‌کنید، آن با `Container` یکی نیست یا به عبارتی `ObjectFactory` اطلاعی در مورد تنظیمات کلاس سفارشی `SmObjectFactory` ندارد. بنابراین دیگر نباید از `ObjectFactory` قدیمی استفاده کنید. در این حالت هر جایی `ObjectFactory.Container` قدمی را داشتید، با `SmObjectFactory` تعویض می‌شود.

نویسنده: صابر فتح الهی
تاریخ: ۱۸:۴۶ ۱۳۹۳/۱۰/۱۴

نحوه دسترسی به کاربری که لاگین کرده چطوریه؟
من با دستور `HttpContext.Current.User` کار می‌کنم
بعضی اوقات نال بر می‌گردونه

نویسنده: وحید نصیری
تاریخ: ۱۹:۱۷ ۱۳۹۳/۱۰/۱۴

: `User.Identity` را در مثال فوق جستجو کنید

```
User.Identity.GetUserName()
User.Identity.GetUserId()
```

نویسنده: سیروان عفیفی
تاریخ: ۱۹:۲۱ ۱۳۹۳/۱۰/۱۴

برای مشخص کردن نمونه پیاده‌سازی کننده `IDataProtectionProvider` در یک برنامه کنسول نیز باید از فایل `Startup` استفاده کرد؟ بیشتر هدف `Seed` کردن دیتابیس است (مثلاً ایمپورت تعداد زیادی کاربر از طریق یک فایل و...). اینکار رو در متدهای `SeedDatabase` هم انجام دادم ولی هر بار استثنای `UserId not found` رو در:

```
result = this.SetLockoutEnabled(user.Id, false);
```

صادر می‌کنند، با گذاشتن `Breakpoint` متوجه شدم که برای `Id` صفر رو در نظر می‌گیره! از این جهت ترجیح دادم برای اینکار از طریق برنامه کنسول ویندوزی هم آن را تست کنم، مثل روشی که در [اینجا](#) برای ایجاد کاربر نوشته شده.

نویسنده: صابر فتح الهی
تاریخ: ۱:۱۵ ۱۳۹۳/۱۰/۱۵

در قسمت تزریق وابستگی کد زیر وجود دارد

```
ioc.For<IIIdentityMessageService>().Use<SmssService>();
ioc.For<IIIdentityMessageService>().Use<EmailService>();
```

در صورت استفاده از اینترفیس مربوطه کدوم سرویس نمونه سازی میشے؟ در صورتی که هر دو سرویس وجود داشته باشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۱۵ ۱:۵۵

این مشکل اصلاح شد. باید از named instance در این حالت خاص استفاده شود.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۳/۱۰/۱۵ ۶:۶

با انجام اصلاحات فوق، زمانی که کاربر اقدام به تایید شماره تلفن همراه کنه با اینکه PhoneCode انتخاب میشه اما بازم قسمت ارسال ایمیل اجرا میشه.

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۳/۱۰/۱۵ ۱۱:۴۸

از همان متده SeedDatabase استفاده کردم، مشکل این بود که در متده Create نوع پسورد از این لحاظ که حداقل باید 6 کاراکتر باشه و درست بودن ایمیل و... نیز بررسی میشه، اگر مقادیر معتبر نباشه مقدار user.Id برابر با صفر میشه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۱۵ ۱۳:۲۲

نیاز به تنظیمات setter injection هم داشت که اضافه شد.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۳/۱۰/۱۵ ۲۰:۲۳

با این تنظیمات

```
// map same interface to different concrete classes
ioc.For<IIIdentityMessageService>().Use<SmsService>().Named("smsService");
ioc.For<IIIdentityMessageService>().Use<EmailService>().Named("emailService");
ioc.For<IApplicationUserManager>().HybridHttpOrThreadLocalScoped()
    .Use<ApplicationUserManager>()
    .Ctor<IIIdentityMessageService>("smsService").Is<SmsService>()
    .Ctor<IIIdentityMessageService>("emailService").Is<EmailService>()
    .Setter<IIIdentityMessageService>(userManager =>
userManager.SmsService).Is<SmsService>()
    .Setter<IIIdentityMessageService>(userManager =>
userManager.EmailService).Is<EmailService>();
```

و این کد سازنده UserManager

```
public ApplicationUserManager(
    IApplicationUserStore store,
    IApplicationRoleManager roleManager,
    IDataProtectionProvider dataProtectionProvider,
    IIIdentityMessageService smsService,
    IIIdentityMessageService emailService)
    : base(store as IUserStore<User,int>)
{
    this.roleManager = roleManager;
    this.dataProtectionProvider = dataProtectionProvider;
    EmailService = emailService;
    SmsService = smsService;
    Debug.WriteLine("Ticks = {0}", DateTime.Now.Ticks);
    Debug.WriteLine(emailService.ToString());
    Debug.WriteLine(smsService.ToString());
    CreateApplicationUserManager();
}
```

در زمان دیباگ اینجور خروجی توی پنجره دیباگ گرفتم

```
Ticks = 635560859158196052
PWS.ServiceLayer.EF.Identity.EmailService
PWS.ServiceLayer.EF.Identity.SmsService
Ticks = 635560859158246098
PWS.ServiceLayer.EF.Identity.EmailService
PWS.ServiceLayer.EF.Identity.EmailService
```

نمی‌دونم چرا دوبار سازنده فراخوانی شده در بار اول تزریق وابستگی درست انجام شده اما در اخیرین بار هم سرویس ایمیل به ایمیل و پیامک هر تزریق شده.

نویسنده: وحید نصیری
تاریخ: ۲:۴۵ ۱۳۹۳/۱۰/۱۶

جهت مدیریت تک و هله‌ای ApplicationUserManger [این تغییرات](#) را اعمال کنید؛ یا از این [فایل نهایی](#) استفاده کنید.

نویسنده: نیکناز
تاریخ: ۱۲:۳۴ ۱۳۹۳/۱۰/۲۳

سلام؛ در روش برنامه نویسی لایه ای به این صورتی که شما آموزش دادید امکان دیدن جدول‌های بانک در بخش Server Explorer نیست؟ منظورم قبل از اینزرت اطلاعات در آنهاست؟ چون می‌خواهیم یکسری از جداول رو دستی مستقیم در بانک پر کنم.

و اینکه بعد از اینکه dbcontext را در پوشه App_Data حذف کردم چه جوری اتصال کاملشو از بین برم چون برنامه ام ارور پیدا نکردن dbcontext را می‌ده و چه جوری می‌تونم دوباره ReCreate کنمش.

نویسنده: وحید نصیری
تاریخ: ۱۳:۲ ۱۳۹۳/۱۰/۲۳

- در سازنده‌ی کلاس [ApplicationDbContext](#)، به `ConnectionString1` اشاره شده که تعریف آن در فایل [web.config](#) موجود است. به عبارتی در این مثال چنین رشته‌ی اتصالی تعریف شده است:

```
<add name="connectionString1"
      connectionString="Data Source=(local);Initial Catalog=TestDbIdentity;Integrated Security = true"
      providerName="System.Data.SqlClient" />
```

بنابراین بانک اطلاعاتی پیش فرض آن TestDbIdentity نام دارد (جهت اتصال به آن، برای مشاهده جداول و یا تغییر و ثبت اطلاعات). این رشته اتصالی هم مخصوص SQL Server تنظیم شده است که می‌توانید توسط management studio و یا سایر ابزارهای مشابه، همانند قبیل به آن متصل شوید.

- «در پوشه App_Data حذف کردم» ... از مثال نهایی کامل شده استفاده کنید ([^](#)) و نیازی به تکرار این مراحل نیست تا خطای یافت نشدن dbcontext را دریافت نکنید.

- برای ReCreate فقط کافی هست که بانک اطلاعاتی TestDbIdentity را drop کنید. بعد برنامه را مجدداً از نو اجرا کنید. چون مراحل migrations آن به [حالت خودکار](#) تنظیم شده است، بانک اطلاعاتی را به صورت خودکار ایجاد می‌کند یا تغییرات [کلاس‌های دومین برنامه](#) را به صورت خودکار به بانک اطلاعاتی اعمال خواهد کرد.

- برای آشنایی بیشتر با مباحث تعریف رشته اتصالی EF Code First و مباحث Migrations آن، [سربی EF Code First](#) را در سایت یکبار مطالعه کنید.

نویسنده: اقبال
تاریخ: ۱۶:۱۰ ۱۳۹۳/۱۰/۲۷

ممنون از مطالب شما اگر بخواهیم از IApplicationUserManager و IApplicationRoleManager یا به طور کلی از ایترفیس‌ها در سفارشی (یا فیلتر هایی) که ایجاد کرده ایم استفاده کنیم ، تنظیمات strucutremap به چه صورت خواهد بود.

```
public class CustomAuthorizeAttribute : AuthorizeAttribute
{
    public IApplicationRoleManager _roleManager { get; set; }
    public IApplicationUserManager _userManager { get; set; }

    public CustomAuthorizeAttribute(IApplicationUserManager userManager,
                                    IApplicationRoleManager roleManager)
    {
        _userManager = userManager;
        _roleManager = roleManager;
    }
    protected override bool AuthorizeCore(HttpContextBase httpContext)
    {
        // Put your custom logic here, returning true for success and false for failure,
        // or return base.AuthorizeCore(httpContext) to defer to the base implementation
        IPrincipal user = httpContext.User;
        IIIdentity identity = user.Identity;

        if (!identity.IsAuthenticated)
        {
            return false;
        }

        bool isAuthorized = true;
        // TODO: perform custom authorization against the App
        var qry = _userManager.Users.Where(u => u.UserName == identity.Name).ToList();

        return isAuthorized;
    }
}
```

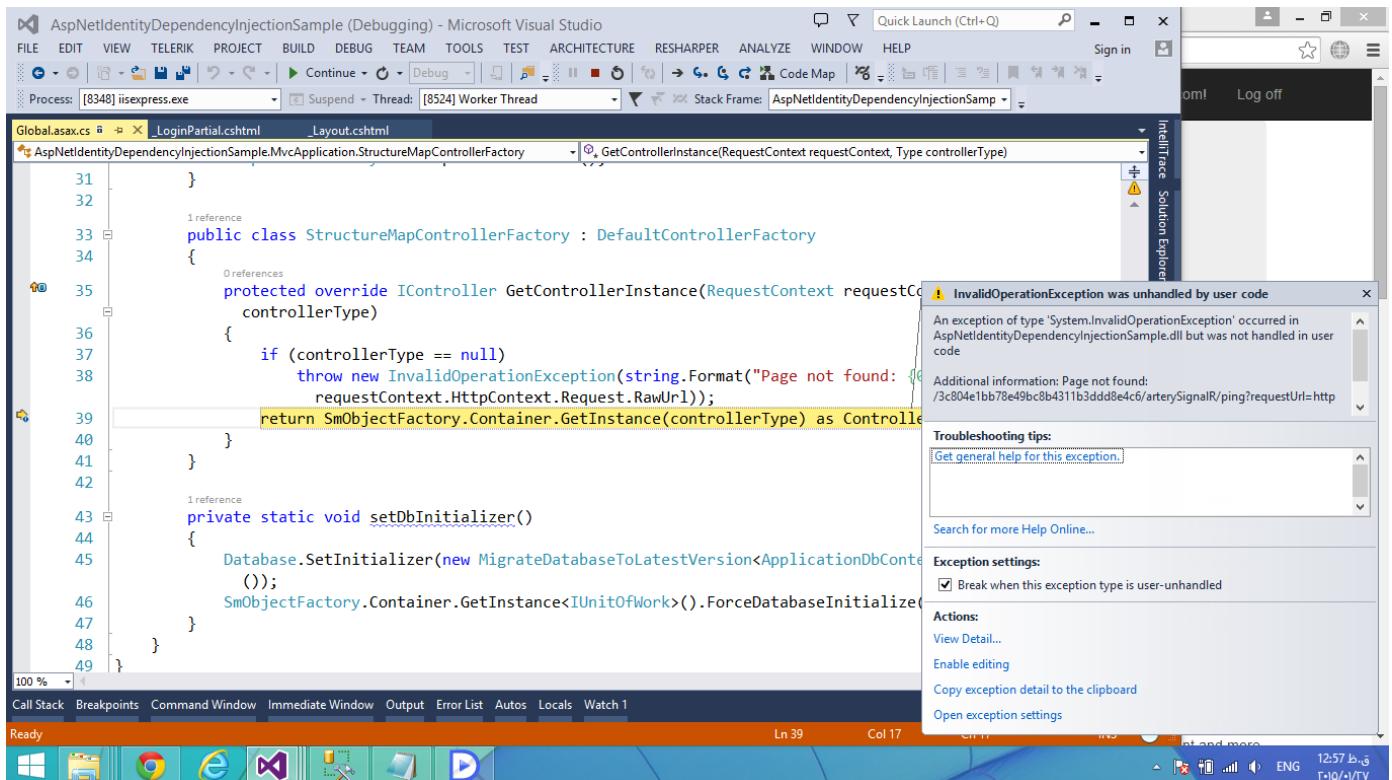
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۲۷ ۱۷:۴۱

« [تزریق وابستگی‌ها در فیلترهای ASP.NET MVC](#) »

نویسنده: م سلیمانی
تاریخ: ۱۳۹۳/۱۱/۰۷ ۱۲:۳۳

با سلام؛ پروژه این مثال و دانلود کردم و سپس اجرаш کردم - مشکلی نداره ولی بعد چند دقیقه ناگهان با این پیغام مواجه میشم.

اعمال تزریق وابستگی‌ها به مثال رسمی



لطفا راهنمایی بفرمایید. با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۱/۰۷ ۱۳:۲۳

- لطفا خطاهای را تصویری ارسال نکنید. متن موجود در تصاویر، قابلیت جستجو و پیگیری ندارند.
- متن آن واضح است و به عمد تنظیم شده است. عنوان کرده است که آدرس خاصی را یافت نمی‌کند و این آدرس‌ها قابلیت تزریق وابستگی ندارند. آدرس آن (arterySignalR/ping) مرتبط است به [browser link](#) که به این صورت می‌تواند آن را غیرفعال کنید:

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
    routes.IgnoreRoute("{*browserlink}", new { browserlink = @".*/arterySignalR/ping" });
    //...
}
```

و یا

```
<appSettings>
    <add key="vs:EnableBrowserLink" value="false" />
</appSettings>
```

نویسنده: س محمد رضا برنتی
تاریخ: ۱۳۹۴/۰۱/۱۶ ۱۶:۳۸

تنظیمات جهت اعمال تزریق وابستگی‌ها در WEB API چگونه خواهد بود؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۱/۱۶ ۱۶:۵۷

تنظیمات اولیه‌ی IoC Container در این حالت، با نمونه‌ی موجود [پکی](#) است. فقط Web API نیاز به معرفی GlobalConfiguration را به SmObjectFactory.Container اطلاعات بیشتر

نویسنده: سروش شیرالی
تاریخ: ۱۸:۳۵ ۱۳۹۴/۰ ۱/۲۸

با سلام؛ خیلی معماري جالبي بود. اما يه سوالی ذهن مرا مشغول کرده . البته من ساختار structureMap را درست نمی‌دانم ولی اکثر IOC Containerها نمونه singlenton را از اينترفيس‌ها برمنی گردانند . که اگر اينگونه باشد به معنی وجود يك Unit OF Work و application بوده در كل DbContext که اين برای يك وب اپلیکیشن به منزله يك فاجعه است. اگر ممکن است مقداری راهنمایی بفرمایید. سپاسگزارم.

نویسنده: وحید نصیری
تاریخ: ۱۹:۶ ۱۳۹۴/۰ ۱/۲۸

طول عمر پیش فرض اشیاء ایجاد شده‌ی توسط استراکچرمپ از نوع [transient](#) است. به این معنا که هر درخواستی از آن شیء، سبب ایجاد يك ولهه‌ی جدید از آن خواهد شد. [اطلاعات بیشتر](#)

نویسنده: سروش شیرالی
تاریخ: ۱۷:۵۸ ۱۳۹۴/۰ ۱/۳۰

فرض کنید ما از این پروژه استفاده کنیم . اما از الگوی repository و unit of work بر روی آنها نیز استفاده کنیم. به این ترتیب IOC از یک طرف applicationDbContext را به کلاس identity store در کلاس pас داده و از طرف دیگر به کلاس unit of work آیا به نظر شما مشکلی پیش نخواهد آمد . با توجه به آنکه به فرموده شما IOC در هر HttpRequest فقط يك آبجکت از کلاس applicationDbContext خواهد ساخت اگر هر دو در همان HttpRequest اقدام به اعمال savechange کنند چه؟ اصلا چنین چیزی ممکن است؟

نویسنده: وحید نصیری
تاریخ: ۱۸:۳۲ ۱۳۹۴/۰ ۱/۳۰

context و uow يك هستند. [قسمت‌های 11 و 12](#) سری EF Code first سایت را نیاز است یکبار مطالعه کنید؛ به همراه تمام نظرات آن‌ها.

نویسنده: محمد رحیم پورابراهیم
تاریخ: ۱۳:۳۰ ۱۳۹۴/۰ ۲/۲۸

من يك پروژه انجام شده دارم که Id از نوع string هست ولی در مثالی که شما زدید int هست ججوری می‌تونم این روشی که گفتید رو پیاده سازی کنم ولی نوع فیلد Id را تغییر ندم، اگه بخواهم تغییر بدم تمام اطلاعات دیتابیس من دستخوش تغییرات زیادی میشه؟

نویسنده: وحید نصیری
تاریخ: ۱۴:۱۰ ۱۳۹۴/۰ ۲/۲۸

- متن را یکبار مطالعه کردید؟ یکبار مثال اصلی را با مثال فوق انطباق دادید؟
- در متن ذکر شده؛ مثال اصلی آن نوع ID را از نوع string تعریف کرده‌است. در اینجا برای نزدیک شدن مثال به دنیای واقعی و آن چیزی که مرسوم است، نوع Id به int تغییر داده شد.
اگر مورد نظر شما نیست، هرجایی که int مشاهده می‌کنید، تغییرش دهید به string. مثلا:

```
 ApplicationUser : IdentityUser<int, CustomUserLogin, CustomUserRole, CustomUserClaim>
```

این int را تغییر دهید به string و در هر جای دیگری که تعریف شده و تغییر کرده. این تغییرات را با انطباق مثال تغییر یافته با

نمونه‌ی اصلی که لینکش در متن هست می‌توانید پیدا کنید.

نویسنده: نرگس حقیقی
تاریخ: ۱۹:۲ ۱۳۹۴/۰۳/۲۹

چطور در پروژه DomainClass که از نوع classLibrary هست using Microsoft.AspNet.Identity.EntityFramework را اضافه کردیم؟

نویسنده: وحید نصیری
تاریخ: ۲۰:۰ ۱۳۹۴/۰۳/۲۹

- وابستگی‌های هر پروژه را با مراجعه به فایل [packages.config](#) آن می‌توانید بررسی کنید.

- بعد از مشخص شدن لیست وابستگی‌ها، فقط کافی است تک آن‌ها را با دستور زیر نصب کنید:

```
PM> install-package name_here
```

همانطور که در مطلب قبلی گفتم، در این مطلب قرار است به WAS بپردازیم؛ در دنباله متن قبلی گفتیم که دو مین وظیفه WWW Service این است: موقعی که یک درخواست جدید در صفحه درخواست‌ها وارد شد، به اطلاع WAS برساند.

Windows Process Activation Service با WAS

در نسخه 7 به بعد، WAS مدیریت پیکربندی application pool و پروسه‌های کارگر را به جای WWW Service به عهده گرفته است. این مورد شما را قادر می‌سازد تا همان پیکربندی که برای Http در نظر گرفته‌اید، بر روی درخواست‌هایی که Http نیستند هم اعمال کنید. همچنین موقعی که سایت شما نیازی به درخواست‌های Http ندارد می‌توانید WAS را بدون WWW Service راه اندازی کنید. به عنوان یک مثال فرض کنید شما یک وب سرویس WCF Listener Adapter را از طریق WCF Listener Adapter مدیریت می‌کنید و احتیاجی به درخواست‌های نوع Http ندارید و http.sys کاری برای انجام ندارد پس نیازی هم به راه اندازی www service نیست.

پیکربندی مدیریتی در WAS

در زمان شروع کار IIS، سرویس WAS اطلاعاتی را از فایل ApplicationHost.config می‌خواند و آن‌ها را به دست listener adapter مربوطه می‌رساند و آن‌ها ارتباط بین WAS و listener adapter می‌داشته‌اند. آن‌ها می‌توانند از آن‌ها بر اساس آن تنظیمات یا پیکربندی‌ها، به درخواست‌ها گوش فرا دهند.

در مورد WCF، ابتدا WAS تنظیمات را برای آداتپور WCF که NetTcpActivator نام دارد ارسال کرده و این آداتپور بر اساس آن listener مربوطه را پیکربندی کرده تا به درخواست‌هایی که از طریق پروتوكول net.tcp می‌رسد گوش فرا دهد. لیست زیر تعدادی از اطلاعاتی را که از فایل پیکربندی می‌خواند و ارسال می‌کند را بیان کرده است:

Global configuration information
Protocol configuration information for both HTTP and non-HTTP protocols

Application pool configuration, such as the process account information

Site configuration, such as bindings and applications

Application configuration, such as the enabled protocols and the application pools to which the applications belong

نکته پایانی اینکه اگر فایل ApplicationHost.config تغییری کند، WAS یک اعلان دریافت کرده و اطلاعات آداتپورها را به روز می‌کند.

مدیریت پروسه‌ها

گفتیم که مدیریت پول و پروسه‌های کارگر جزء وظایف این سرویس به شمار می‌رود. موقعی که یک protocol listener درخواستی را دریافت می‌کند، WAS چک می‌کند که آیا یک پروسه کارگر در حال اجراست یا خیر. اگر application pool پروسه‌ای داشته باشد که در حال سرویس دهی به درخواست هاست، آداتپور درخواست را به پروسه کارگر ارسال می‌کند. در صورتی که پروسه‌ای در application pool در حال اجرا نباشد، WAS یک پروسه جدید را آغاز می‌کند و آداتپور درخواست را به آن پاس می‌کند.

نکته: از آنجایی که WAS هم پروسه‌های http و هم non-http را مدیریت می‌کند، پس می‌توانید از یک application pool برای چندین protocol استفاده کنید. به عنوان مثال شما یک سرویس XML دارید که می‌توانید از آن برای سرویس دهی به پروتوكول‌های Http و net.tcp بهره بگیرید.

IIS مژول‌ها

قبل مقاله ای در مورد moduleها با نام "کمی در مورد [moduleها](#)" قرار داده بودیم که بهتر است برای آشنایی بیشتر، به آن رجوع کنید. به غیر از وب کانفیگ که برای معروفی مژول‌ها استفاده می‌کردیم، می‌توانید به صورت گرافیکی و دستی هم این کار را

انجام بدھید. ابتدا یک پروژہ class library ایجاد کرده و مازول خود را بنویسید و سپس آن را به یک dll تبدیل کنید و dll را در شاخه bin که این شاخه در ریشه وب سایتتان قرار دارد کپی کنید. سپس در IIS قسمت Add گزینه module را انتخاب کنید و در قسمت اول نامی برای آن و در قسمت بعدی دقیقا همان قوانین type که در وب کانفیگ مشخص می‌کردید را مشخص کنید:

Namespace.ClassName

گزینه invoke only for requests to asp.net and manage handlers را هم تیک بزنید. کار تمام است.

ماژول‌های کد ماشین یا native

این مازول‌ها به صورت پیش فرض به سیستم اضافه شده‌اند و در صورتی که میخواهید جایگزینی به منظور خصوصی سازی انجام دهید آن‌ها را پاک کنید و مازول جدید را اضافه کنید.

جدول مازول‌های HTTP

نام مازول	توضیحات	نام فایل منبع
CustomErrorModule	موقعی که هنگام response، کد خطای تویید می‌گردد، پیام خطای را پیکربندی و سپس ارسال می‌کند.	Inetsrv\Customerr.dll
HttpRedirectionModule	تنظيمیات redirection برای درخواست‌های http را در دسترس قرار می‌دهد.	Inetsrv\Redirect.dll
ProtocolSupportModule	انجام عملیات مربوط به پروتوكلهای بر عهده این مازول است؛ مثل تنظیم کردن .response قسمت هدر برای response.	Inetsrv\Protosup.dll
RequestFilteringModule	این مازول از IIS 7.5 به بعد اضافه شد. درخواست‌ها را فیلتر می‌کند تا پروتوكل و رفتار محتوا را کنترل کند.	Inetsrv\modrqfslt.dll
WebDAVModule	این مازول از IIS 7.5 به بعد اضافه شد. امنیت بیشتر در هنگام انتشار محتوا روی HTTP SSL.	Inetsrv\WebDAV.dll

ماژول‌های امنیتی

نام مازول	توضیحات	نام فایل منبع
AnonymousAuthenticationModule	موقعی که هیچ کدام از عملیات authentication با موفقیت روبرو نشود، Anonymous authentication عملیات انجام می‌شود.	Inetsrv\Authanon.dll
BasicAuthenticationModule	عمل ساده و اساسی authentication را انجام می‌دهد.	Inetsrv\Authbas.dll
CertificateMappingAuthenticationModule	انجام عمل Certificate Mapping Active Directory در authentication	Inetsrv\Authcert.dll
DigestAuthenticationModule	Digest authentication	Inetsrv\Authmd5.dll
IISCertificateMappingAuthenticationModule	همان Certificate Mapping authentication ولی اینبار با IIS Certificate mapping . Certificate nModule	Inetsrv\Authmap.dll

آشنایی با ساختار IIS قسمت سوم

نام مازول	توضیحات	نام فایل منبع
RequestFilteringModule	عملیات اسکن URL از قبیل نام صفحات و دایرکتوری‌ها ، نوع verb و یا کاراکترهای مشکوک و خطرآفرین	Inetsrv\Modrqflt.dll
UrlAuthorizationModule	عمل URL authorization	Inetsrv\Urlauthz.dll
WindowsAuthenticationModule	عمل NTLM integrated authentication	Inetsrv\Authsspi.dll
IpRestrictionModule	محدود کردن IP‌های نسخه 4 لیست شده در IP Security در قسمت پیکربندی	Inetsrv\iprestr.dll

ماژول‌های محتوا

نام مازول	توضیحات	نام فایل منبع
CgiModule	ایجاد پردازش‌های Common Gateway Interface (CGI) به منظور ایجاد خروجی response	Inetsrv\Cgi.dll
DefaultDocumentModule	تلash برای ساخت یک سند پیش فرض برای درخواست‌هایی که دایرکتوری والد ارسال می‌شود	Inetsrv\Defdoc.dll
DirectoryListingModule	لیست کردن محتوای یک دایرکتوری	Inetsrv\dirlist.dll
IsapiModule	میزبانی فایل‌های ISAPI	Inetsrv\Isapi.dll
IsapiFilterModule	پشتیبانی از فیلتر‌های ISAPI	Inetsrv\Filter.dll
ServerSideIncludeModule	پردازش کدهای include شده سمت سرور	Inetsrv\Iis_ssi.dll
StaticFileModule	ارائه فایل‌های ایستا	Inetsrv\Static.dll
FastCgiModule	پشتیبانی از CGI	Inetsrv\iisfcgi.dll

ماژول‌های فشرده سازی

Inetsrv\Comdyn.dll	DynamicCompressionModule	فشرده سازی پاسخ gzip با response
Inetsrv\Compstat.dll	StaticCompressionModule	فشرده سازی محتوای ایستا

ماژول‌های کش کردن

Inetsrv\Cachfile.dll	FileCacheModule	تهیه کش در مد کاربری برای فایل‌ها.
Inetsrv\Cachhttp.dll	HTTPCacheModule	تهیه کش مد کاربری و مد کرنل برای http.sys
Inetsrv\Cachtokn.dll	TokenCacheModule	تهیه کش مد کاربری بر اساس جفت نام کاربری و یک token که توسط Windows user principals تولید شده است.
Inetsrv\Cachuri.dll	UriCacheModule	تهیه یک کش مد کاربری از اطلاعات URL

ماژول‌های عیب‌یابی و لاغ کردن

Inetsrv\Logcst.dll	بارگزاری ماژول‌های خصوصی سازی شده جهت لاغ کردن	CustomLoggingModule
Inetsrv\Iisfreb.dll	برای ردیابی درخواست‌های ناموفق	FailedRequestsTracingModule
Inetsrv\Loghttp.dll	دریافت اطلاعات و پردازش وضعیت http.sys برای لاغ کردن	HttpLoggingModule
Inetsrv\Iisreqs.dll	ردیابی درخواست‌هایی که در حال حاضر در پروسه‌های کارگر در حال اجرا هستند و گزارش اطلاعاتی در مورد وضعیت اجرا و کنترل رابط برنامه نویسی کاربردی.	RequestMonitorModule
Inetsrv\Iisetw.dll	گزارش رخدادهای Microsoft Event ETW یا به اختصار Tracing for Windows	TracingModule

ماژول‌های مدیریتی و نظارتی بر کل ماژول‌ها

Microsoft.NET\Framework\v2.0.50727\webengine.dll	مدیریتی بر ماژول‌های غیر native که در پایین قرار دارند.	ManagedEngine
Inetsrv\validcfg.dll	اعتبارسنجی خطاهای، مثل موقعی که برنامه در حالت integrated اجرا شده و ماژول‌ها یا هندلرهای web system تعریف شده‌اند.	ConfigurationValidationModule

از IIS6 به بعد در حالت integrated و ماقبل، در حالت کلاسیک می‌باشند. اگر [مقاله ماژول ها](#) را خوانده باشید می‌دانید که تعريف آن‌ها در وب کانفیگ در بین این دو نسخه متفاوت هست و رویداد سطر آخر در جدول بالا این موقعیت را چک می‌کند و اگر به خاطر داشته باشید با اضافه کردن یک خط اعتبارسنجی آن را قطع می‌کردیم. در مورد هندلرهای هم به همین صورت می‌باشد. به علاوه ماژول‌های native IIS این امکان را فراهم می‌آورند تا از ماژول‌های کد مدیریت شده (یعنی CLR) برای توسعه توابع و کارکرد IIS بهره مند شوید:

ماژول	توضیحات	منبع
AnonymousIdentification	مدیریت منابع تعیین هویت برای کاربران asp.net profile ناشناس مانند	System.Web.Security.AnonymousIdentificationModule
DefaultAuthentication	اطمینان از وجود شی Authentication در context مربوطه	System.Web.Security.DefaultAuthenticationModule
FileAuthorization	تایید هویت کاربر برای دسترسی به فایل درخواست	System.Web.Security.FileAuthorizationModule
FormsAuthentication	با این قسمت که باید کاملاً آشنا باشید؛ برای تایید هویت کاربر	System.Web.Security.FormsAuthentication

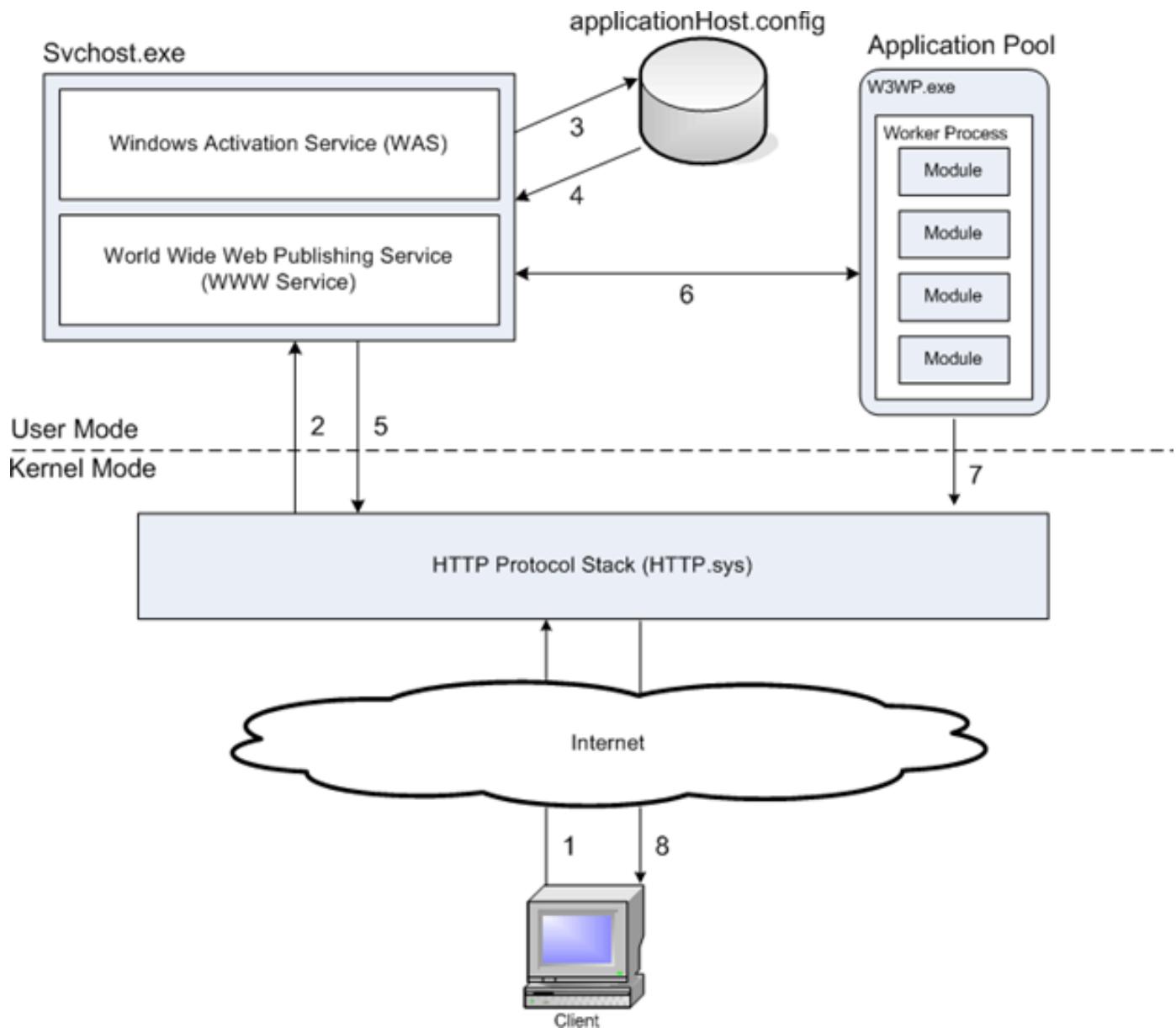
منبع	توضیحات	ماژول
System.Web.Caching.OutputCacheModule	مدیریت کش OutputCache	
System.Web.Profile.ProfileModule	مدیریت پروفایل کاربران که تنظیماتش را در یک منبع داده‌ای چون دیتابیس ذخیره و بازیابی می‌کند.	Profile
System.Web.Security.RoleManagerModule	مدیریت نقش و سمت کاربران RoleManager	
System.Web.SessionState.SessionStateModule	مدیریت session ها	Session
System.Web.Security.UrlAuthorizationModule	آیا کاربر حاری حق دسترسی به URL درخواست را دارد؟	UrlAuthorization
System.Web.UrlMappingsModule	تبديل یک Url واقعی به یک کاربرپسند	UrlMappingsModule
System.Web.Security.WindowsAuthenticationModule	شناسایی و تایید و هویت یک کاربر بر اساس لاغین او به ویندوز	WindowsAuthentication

پردازش درخواست‌های HTTP در IIS

بگذارید در این قسمت خلاصه‌ای از درخواست‌های نوع HTTP را که تا به الان گفته‌ایم، به همراه شکل بیان کنیم: موقعی که کلاینت درخواست خود را مبنی بر یکی از منابع سرور ارسال می‌کند، Http.sys این درخواست را می‌گیرد. WAS با http.sys تماس گرفته و درخواست می‌کند تا اطلاعات پیکربندی یا تنظیمات IIS را برای نحوه‌ی برخورد با درخواست، برایش بفرستد.

WAS هم اطلاعات پیکربندی شده را از محل ذخیره داده‌ها که applicationHost.config هست، می‌خواند. application WWW Service که یک آدأپتور برای Http.sys هست، اطلاعات را از WAS دریافت می‌کند. این اطلاعات شامل پیکربندی pool و سایت می‌باشد.

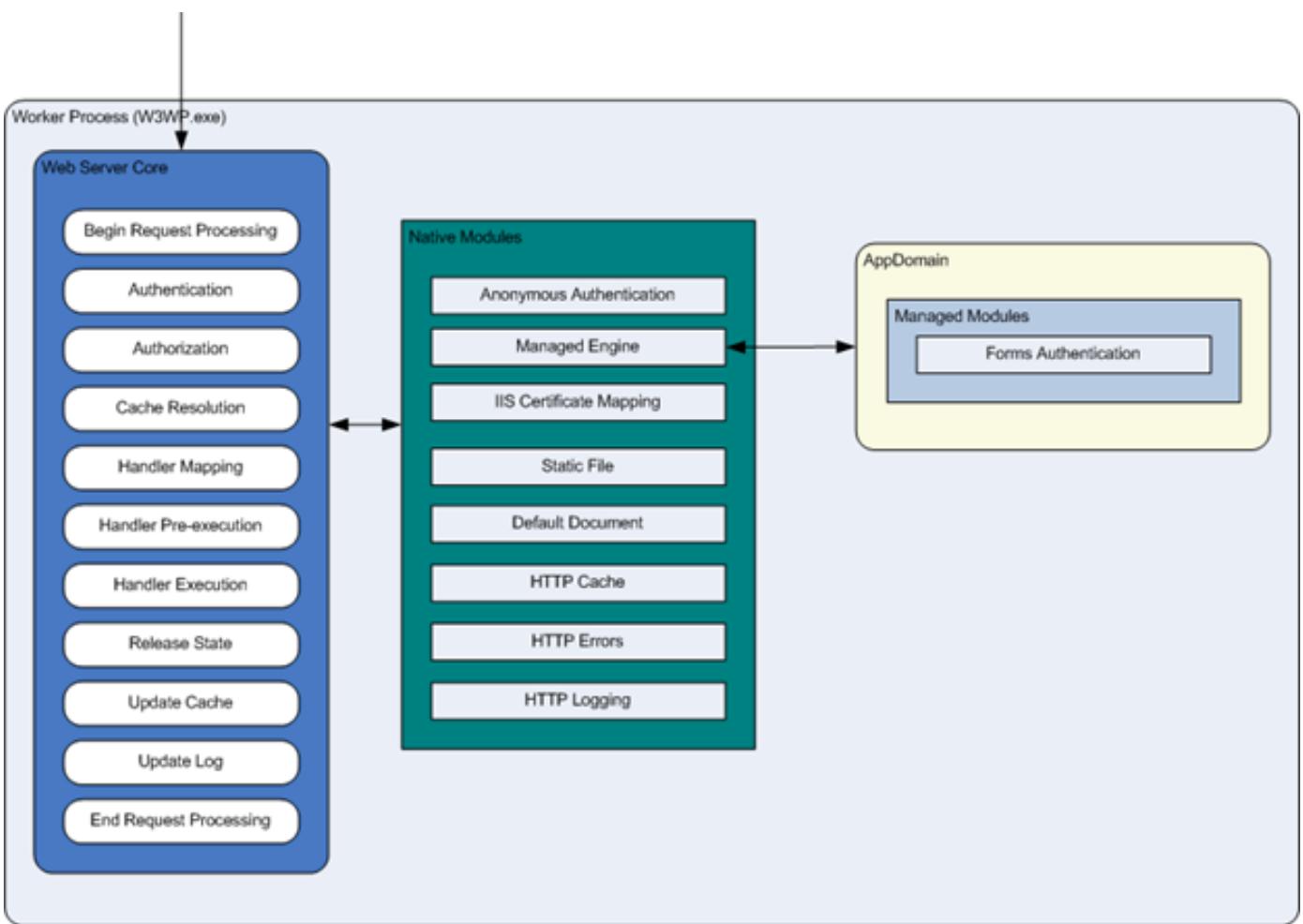
WWW Service اطلاعات را برای Http.sys می‌فرستد. WAS یک پروسه کارگر را در application pool ایجاد می‌کند تا درخواست رسیده مورد پردازش قرار بگیرد. پروسه‌های کارگر درخواست را پردازش کرده و خروجی یا response مورد نظر را تولید می‌کنند. Http.sys نتیجه را دریافت و برای کلاینت می‌فرستد.



حال بباید بینیم موقعی که درخواست وارد پروسه کارگر میشود چه اتفاقی میافتد؟

در پروسه های کارگر، یک درخواست از مراحل لیست شده ای به ترتیب عبور می کند. در هسته وب سرور، رویدادهای را فراخوانی می کند که در هر رویداد چندین مازول native برای کارهایی چون authentication یا events logs دارد و در صورتیکه درخواستی نیاز به یک مازول مدیریت شده CLR داشته باشد، از مازول native managedEngine کمک گرفته و یک app domain را ایجاد می کند تا مازول های مدیریت شده، عملیات لازم خودشان را انجام دهدن. مثل authentication form و ...

موقعی هم که درخواست، از تمامی این رویدادها عبور کند، http.sys برای response ارسال می شود تا به کلاینت بازگشت داده شود. شکل زیر نحوه ورود یک درخواست به پروسه کارگر را نشان می دهد.



از نسخه 7 به بعد، IIS از یک معماری ماژولار استفاده می‌کند و این ویژگی، سه فایده دارد:
Componentization یا کامپوننت سازی Extensibility یا توسعه پذیری یا قابل گشترش

ASP.NET Integration

Componentization

همه خصوصیات و ویژگی‌های این وب سرور، توسط کامپوننت‌ها مدیریت می‌شوند که باعث می‌شود شما به راحتی بتوانید کامپوننتی را اضافه، حذف یا جایگزین کنید و این باعث می‌شود که چندین امتیاز از IIS قبلي جلوتر باشد:
باعث کاهش [attack surface](#) می‌شود که در نتیجه امنیت سیستم را بالا می‌برد. با ویژگی حذف کامپوننت‌ها شما می‌توانید ویژگی‌های غیرقابل استفاده IIS را حذف کنید تا ورودی‌های سیستم کاهش یابد. پس با کاهش ویژگی‌هایی که از آن هرگز استفاده نخواهید کرد، مدخل ورود هکر را از بین برده تا امنیت سرور بالاتر برسد.
افزایش کارآیی و کاهش مصرف حافظه. با حذف ویژگی‌هایی که هرگز استفاده نمی‌کنید، در مصرف حافظه و بهینه استفاده شدن منابع سرور صفحه جوی، کنید.

با وجود ویژگی افزودن و جایگزینی کامپوننت‌ها، ناخودآگاه ذهن ما به سمت کاستوم سازی یا خصوصی سازی کشیده می‌شود. با این کار شما به راحتی یک custom server ایجاد می‌کنید که این سرور بر اساس علائق شما کارش را انجام می‌دهد و به راحتی امکاناتی چون افزودن third party را به توسعه دهنده می‌دهد.

Extensibility

با توجه به موارد بالا، خصوصی سازی باعث گسترش امکانات IIS می‌شود که می‌تواند به دلایل زیر اتفاق بیفتد: قدرت بخشی به برنامه‌های وب. امکانات و قدرتی که می‌تواند در این حالت به برنامه‌های در حال اجرا داد به مراتب بیشتر از استفاده از لایه‌های داخلی خود برنامه هست. برای اینکار شما می‌توانید کدهای خود را با ASP.NET نوشته یا از کدهای native چون C++ استفاده کنید.

تجربه‌ای از توسعه پذیری ساده‌تر و راحت‌تر استفاده از قدرت و تمامی امکانات را به شما می‌دهد و می‌توانید تمام دستورات را برای همه منابع حتی فایل‌های ایستا، ASP ، CGI و دیگر منابع اجرا کنید.

ASP.NET Integration

تمامی موارد گفته شده بالا در این گزینه خلاصه می‌شود : محیط ASP.NET Integration به شما امکان استفاده از تمامی امکانات و منابع را به طور کامل می‌دهد. [دانلود مازول‌های مدیریت شده](#) [دانلود مازول‌های native](#)

در مطلوب قبلی در مورد ماژول‌بار بودن IIS زیاد صحبت کردیم، ولی اجازه بدھید این مورد را به صورت کاربردی‌تر و موشکافانه‌تر بررسی کنیم. برای اینکه به مشکلی در طول این سری از مطالب برخورید، IIS را به صورت کامل یعنی full feature نصب نمایید. از بخش control panel>programs & features>Turn Windows features on or off اقدام نمایید و هرچه زیر مجموعه Internet information service هست را برگزینید. در صورتی که از نسخه‌های ویندوز سرور 2008 استفاده می‌کنید از طریق server manager>roles>web server اقدام نمایید.
 برای نصب یک ماژول باید دو مرحله را انجام داد:
 نصب ماژول
 فعال سازی ماژول

نکته‌ای که در مورد ماژول‌های native وجود دارد این هست که این ماژول‌ها دسترسی بدون محدودیتی به منابع سروری دارند و از این رو حتما باید این نکته را دقت کنید که ماژول native شما از یک منبع مورد اعتماد دریافت شده باشد.

نصب یک native module
 برای نصب می‌توانید یکی از سه راه زیر را استفاده کنید:
 ویرایش دستی فایل کانفیگ و از نسخه IIS7.5 به بعد هم می‌توانید از configuration editor هم استفاده کنید.
 استفاده از محیط گرافیکی IIS
 استفاده از خط فرمان با دستور Appcmd

مزیت روش دستی این هست که شما دقیقاً می‌دانید در پشت صحنه چه اتفاقی می‌افتد و نتیجه هر کدام از این سه روش، اضافه شدن یک مدخل ورودی به تگ `<globalmodules>` است. برای اعمال تغییرات، مسیر زیر را بروید:

```
%windir%\system32\inetsrv\config\applicationhost.config
```

کسی که نیاز به دسترسی به این مسیر و انجام تغییرات دارد باید در بالاترین سطح مدیریتی سرور باشد. اگر فایل را باز کنید و تگ `<globalmodule` را پیدا کنید متوجه می‌شوید که تمامی ماژول‌ها در این قسمت معرفی شده‌اند و برای خود یک مدخل ورودی یا همان تگ add را دارند که در آن مسیر فایل dll هم ذکر شده است:

```
<globalModules>
<add name="DefaultDocumentModule" image="%windir%\system32\inetsrv\defdoc.dll"/>
<add name="DirectoryListingModule" image="%windir%\system32\inetsrv\dirlist.dll"/>
<add name="StaticFileModule" image="%windir%\system32\inetsrv\static.dll"/>
...
</globalModules>
```

برای حذف یا جایگزینی یک ماژول را به راحتی می‌توانید مدخل ورودی یک ماژول را به صورت دستی حذف نمایید و برای جایگزینی هم بعد از حذف، ماژول خود را معرفی کنید. ولی توجه داشته باشید که این حذف به معنی حذف این ماژول از تمامی اپلیکیشن‌های موجود بر روی IIS هست و سپس اضافه کردن یک ماژول به این بخش. همچنین اگر قصد شما فقط حذف یک ماژول از روی یکی از اپلیکیشن‌ها باشد باید از طریق فایل کانفیگ سایت از مسیر `<system.webserver><modules>` و با استفاده از تگ‌های `remove` و `add` به معرفی یک ماژول مختص این اپلیکیشن و یا حذف یک ماژول خاص اقدام نمایید.

PreConditions

این ویژگی می‌تواند در خط معرفی ماژول، مورد استفاده قرار بگیرد. اگر به فایل نگاه کنید می‌بینید که در بعضی خطوط این ویژگی ذکر شده است. تعریف این ویژگی به هسته IIS می‌گوید که این ماژول در چه مواردی و به چه شیوه‌ای باید به کار گرفته شود.

```

<add name="ManagedEngine64" image="%windir%\Microsoft.NET\Framework64\v2.0.50727\webengine.dll"
preCondition="integratedMode,runtimeVersionv2.0,bitness64" />

<add name="ManagedEngine" image="%windir%\Microsoft.NET\Framework\v2.0.50727\webengine.dll"
preCondition="integratedMode,runtimeVersionv2.0,bitness32" />

<add name="ManagedEngineV4.0_32bit" image="%windir%\Microsoft.NET\Framework\v4.0.30319\webengine4.dll"
preCondition="integratedMode,runtimeVersionv4.0,bitness32" />

<add name="ManagedEngineV4.0_64bit"
image="%windir%\Microsoft.NET\Framework64\v4.0.30319\webengine4.dll"
preCondition="integratedMode,runtimeVersionv4.0,bitness64" />

```

مقداری که میتواند بگیر شامل موارد زیر هستند: **bitness**

این گزینه به دو صورت **bitness64** و **bitness32** یافت میشود. امروزه پردازندهای 64 بیتی بسیار متداول شده اند و بسیاری از تولید کنندگان دارند به سمت عرضه ابزارهای 64 بیتی رو می آورند و به زودی عرضه های 32 بیتی را متوقف می کنند و به سمت سیستم عامل های 64 بیتی هم هنوز برنامه های 32 بیتی زیادی هستند که مورد استفاده قرار می گیرند و نمی توان آنها را نادیده گرفت. برای همین ویدو زهای 64 بیتی مایکروسافت در کنار محیط 64 بیتی شان از یک محیط 32 بیت به اسم **WOW64** استفاده می کنند. در این حالت این امتیاز به شما داده می شود که از پروسه های کارگر 32 بیتی در کنار پروسه های کارگر 64 بیتی استفاده کنید و **PreCondition** به **bitness32** یا **bitness64** تنظیم می شود تا از صحبت بارگذاری **dll** در یک محیط درست مطمئن شود. در صورتی که این خصوصیت ذکر نشود یک هندر 32 بیتی و 64 بیتی و یک **map module** اجرا می شود.

Runtime version

اگر مژول خاصی برای اجرا به ورژن خاصی از **.net framework** نیاز دارد، این ویژگی ذکر می شود. در صورتی که مژولی قصد اجرای بر روی فریم ورک اشتباهی داشته باشد سبب خطأ خواهد شد.

ManagedHandler

با معرفی IIS7 ما با یک مدل توسعه پذیر روبرو شدیم و می توانستیم مژولها و هندرهای خود را بنویسیم و مستقیما در Pipeline قرار دهیم ولی سوییچ کردن بین دو بخش کدهای مدیریت شده و **native** یک عمل سنگین برای سیستم به شمار می آید و به منظور کاهش این بار گزینه **managedhandler** قرار داده شده است تا تعیین کند موقعی که درخواست نیازی به این مژول ندارد، این مژول اجرا نگردد. به عنوان مثال فایل های ایستا چون **jpg** یا **html** و... شامل این مژول نخواهد شد. واضح ترین مثال در این زمینه **Forms Authentication** می باشد و موقعی اجرا می شوند که درخواست فایل های **aspx** شده باشد و اگر یک فایل **html** را درخواست کنید این مژول امنیتی روی آن اثری ندارد و عملیات شناسایی هویت روی آن اجرا نمی شود و اگر می خواهید روی همه فایل ها، این عملیات شناسایی انجام شود باید خصوصیت **precondition="managedhandler"** حذف شود. در صورتی که تگ **module** را به صورت زیر بنویسید تمامی مژولها برای تمامی درخواست ها اجرا خواهد شد، صرف نظر از اینکه آیا مژولی به صورت **precondition="managedmodule"** مقداردهی شده است یا خیر.

```
<modules runAllManagedModulesForAllRequests="true"/>
```

The Mode Precondition

تا به الان گفتیم که چگونه میتوانیم یک مژول و یا هندر مدیریت شده ای را به Pipeline اضافه کنیم؛ ولی IIS7 به بالا نیاز دارد که تا پروسه های کارگر را به رو شی خاص به این منظور اجرا کند و فریم ورک دات نت را برای اجرای آنها بارگذاری کند. همچنین به اجرای مژولی به اسم **dll** برای مدیریت مژولهای مدیریت شده نیازمند است و خود IIS در مورد کدهای مدیریت شده چیزی متوجه نمی شود. پس ما برای اینکه IIS را متوجه این موضوع نمائیم، باید **Integrated mode** را به آن معرفی کنیم.

در نسخه های قبلی IIS یک روش قدیمی برای کدهای مدیریت شده وجود داشت که از طریق اینترفیسی به نام ISAPI صورت می گرفت. در این حالت **ASPNET_ISAPI.DLL** مسئول این کار بود و اگر هنوز هم میخواهید از این **dll** در نسخه های جدیدتر IIS کمک بگیرید باید به جای معرفی **classic mode**، **integration mode** را معرفی کنید.

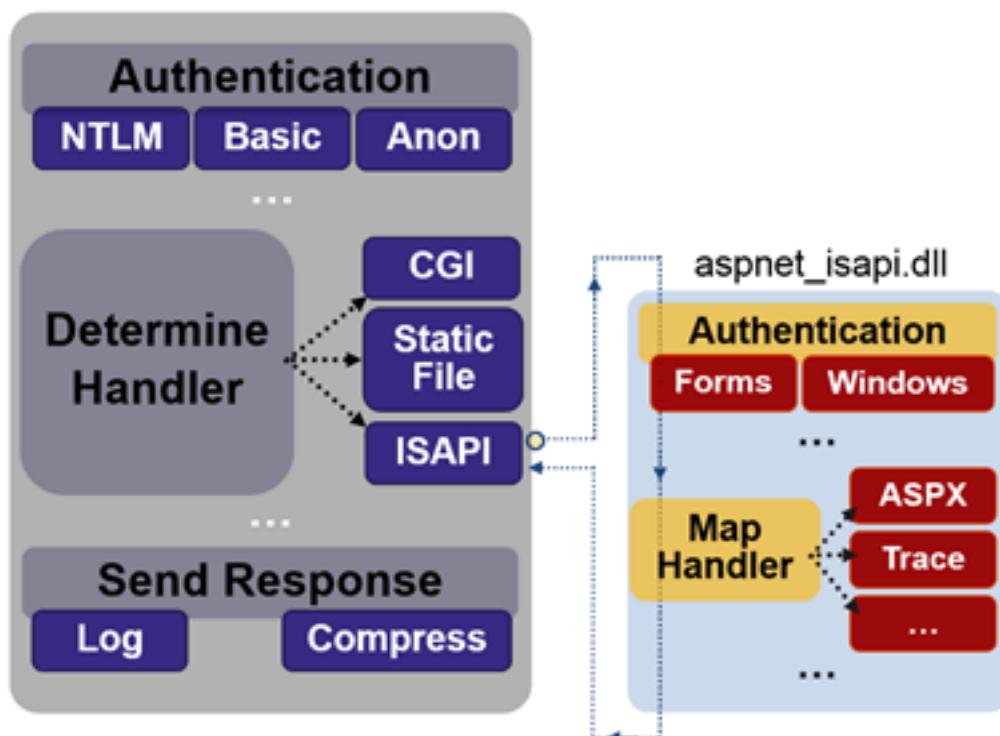
با برابر کردن **precondtion** به مقدار "integratedmode" هندر یا مژول شما در یک پول با خصوصیت **barazari** بارگذاری خواهد شد و اگر مقدار آن "classicmode" باشد در یک پول بدون خصوصیت **barazari** می شود.

تفاوت بین دو روش کلاسیک و مجتمع **integrated** بر سر این هست که در روش جدید، مژول شما به عنوان یک پلاگین برای IIS

دیده نمی‌شود و کد شما را جزئی از کامپوننت‌های خود به شمار می‌آورد. به صورت واضح‌تر در حالت کلاسیک موقعی که درخواستی وارد pipeline می‌شود ابتدا از کامپوننت‌ها و ماژول‌های داخلی خود IIS عبور داده می‌شد و بعد فایل ASPNET_ISAPI.DLL جهت پردازش کدهای مدیریت شده صدا زده می‌شد و با توجه به کدهای شما، بعضی مراحل تکرار می‌شد؛ مثلاً اگر کد شما در مورد Authentication بود و بعد از گذر از مراحل auth داخل خود IIS و بقیه موارد دوباره نوبت کد شما و گذر از مراحل authentication بود. یعنی وجود دو pipeline؛ ولی در حالت مجتمع این دوباره انجام وظیفه از بین رفته است چرا که کدهای شما به طور مستقیم در pipeline قرار دارند و آن‌ها را جزئی از خود می‌داند، نه یک پلاگین که افزون بر فعالیت خودشان، اجرای کدهای شما رو هم بر دوش بکشند.

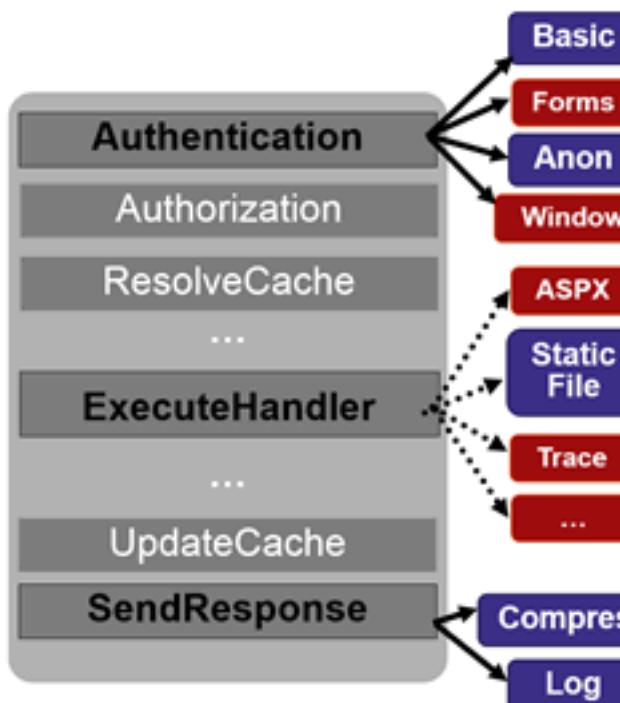
شکل زیر نمونه‌ای از حالت کلاسیک را نشان می‌دهد که در آن دو بار عمل auth دارد انجام می‌گیرد.

IIS6 ASP.NET Integration



شکل زیر هم نمونه‌ای حالت مجتمع هست:

IIS7 ASP.NET Integration



- **Classic Mode**
 - runs as ISAPI
- **Integrated Mode**
 - .NET modules / handlers plug directly into pipeline
 - Process all requests
 - Full runtime fidelity

در کل امروزه دیگر استفاده از روش کلاسیک راهکار درستی نیست و این ویژگی تنها به عنوان یک سازگاری با نمونه کارهای قدیمی است.

تگ‌هایی که از خصوصت precondition استفاده می‌کنند به شرح زیر هستند:

ISAPI filters

globalModules

handlers

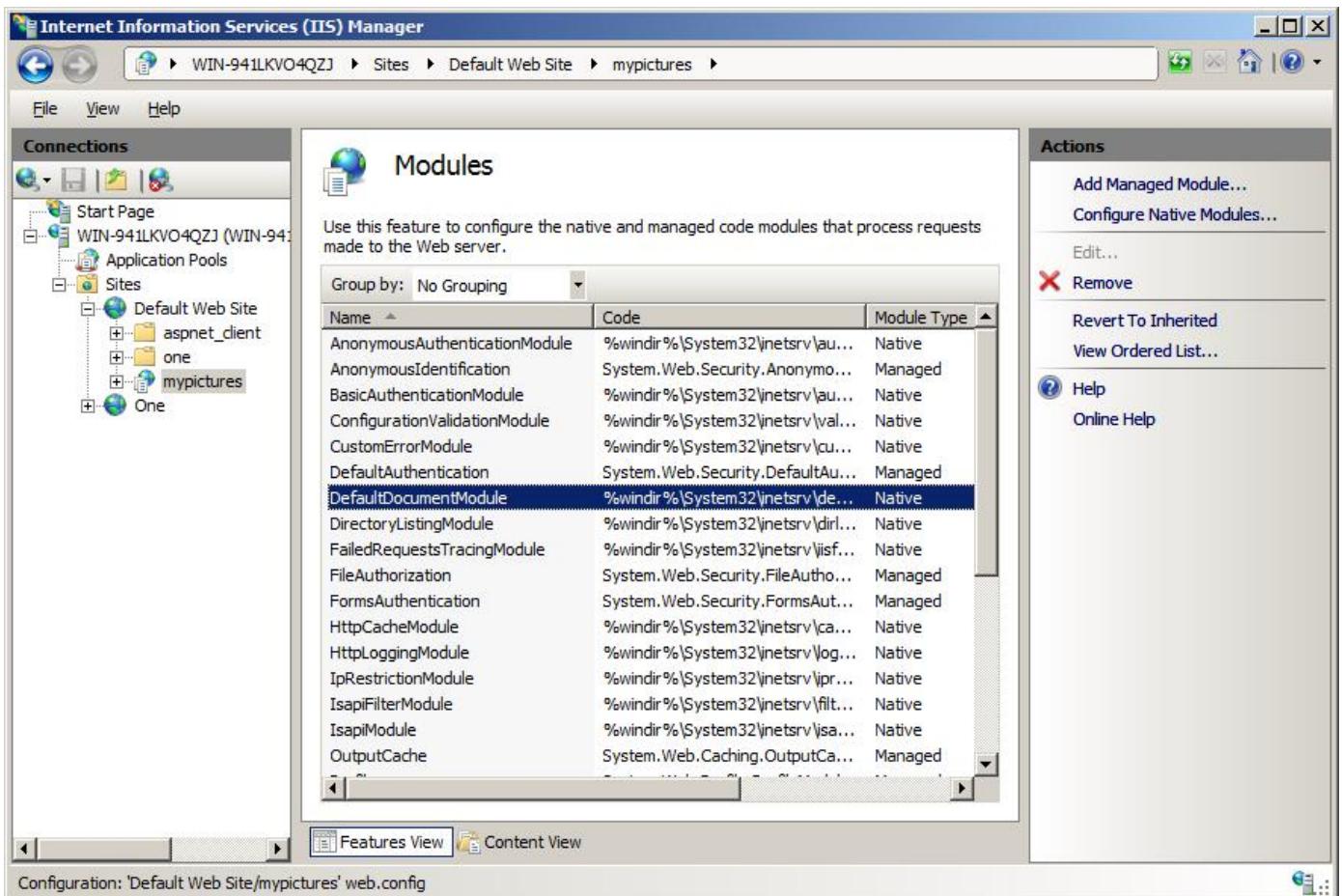
modules

در مورد بقیه تگ‌ها در آینده بیشتر بحث می‌کنیم. بهتر هست مطلب را با توضیح precondition جهت ممانعت از طولانی و طومار شدن در اینجا ببندیم و در قسمت آینده دیگر روش‌های نصب ماژولمان را دنبال کنیم.

در مطلب قبلی روش دستی را برای اضافه کردن ماثول‌های خود، نام بردیم. در اینجا به روش‌های دیگر اضافه کردن ماثول‌ها می‌پردازیم.
استفاده از محیط گرافیکی IIS جهت لیست کردن، اضافه و حذف ماثول‌ها



به بخش modules در IIS بروید. در پنل سمت راست همه امکانات جهت افزودن و ویرایش و حذف وجود دارند:



روش معرفی ماژول در خط فرمان با استفاده از دستور Appcmd

```
Appcmd.exe install module /name:MODULE_NAME /image:PATH_TO_DLL
```

قسمت name که نام ماژول است و قسمت image هم مسیر قرار گرفتن فایل dll هست.

برای نمونه:

```
%windir%\system32\inetsrv\appcmd.exe install module /name:DefaultDocumentModule
/image:%windir%\system32\inetsrv\defdoc.dll
```

در صورتیکه ماژولی که قبلا افزوده شده باشد را بخواهید اضافه کنید، خطای زیر را دریافت خواهید کرد:

```
ERROR ( message:Failed to add duplicate collection element "DefaultDocumentModule". )
```

جهت حذف ماژول دستور زیر را صادر کنید:

```
Appcmd.exe uninstall module MODULE_NAME
```

نمونه:

```
%windir%\system32\inetsrv\appcmd.exe uninstall module DefaultDocumentModule
```

گرفتن کوئری یا لیستی از مژول‌های فعال برای یک اپلیکیشن یا عمومی:

```
Appcmd.exe list modules [/app.name:APPLICATION_NAME]
```

سویچ `aap.name` اختیاری است ولی اگر نام یک اپلیکیشن را به آن بدهید، فقط مژول‌هایی را که روی این اپلیکیشن اجرا می‌شوند، لیست می‌کند.

نمونه:

```
%windir%\system32\inetsrv\appcmd.exe list modules /app.name:"Default Web Site"
```

کد زیر هم نمونه‌ای برای لیست کردن تمامی مژول‌های عمومی که بر روی تمامی اپلیکیشن‌ها اجرا می‌شوند:

```
%windir%\system32\inetsrv\appcmd.exe list modules
```

خط زیر یک مژول را برای همه اپلیکیشن‌ها یا اپلیکیشن خاصی فعال می‌کند که بستگی دارد سویچ `type` چگونه مقداردهی شده باشد:

```
Appcmd.exe add module /name:MODULE_NAME /type:MGD_TYPE
```

برای مثال خط زیر باعث می‌شود مژول `default web site` فقط برای وب اپلیکیشن `Forms Authentication` فعال شود:

```
%windir%\system32\inetsrv\appcmd.exe add module /name:FormsAuthentication  
/type:System.Web.Security.FormsAuthenticationModule /app.name:"Default Web Site"
```

یا در پایین آن را به صورت عمومی یا `global` فعال می‌کند:

```
%windir%\system32\inetsrv\appcmd.exe add module /name:FormsAuthentication  
/type:System.Web.Security.FormsAuthenticationModule
```

برای غیرفعال کردن یک مژول از دستور زیر استفاده می‌شود:

```
Appcmd.exe delete module MODULE_NAME [/app.name:APPLICATION_NAME]
```

اگر غیرفعال کردن یک مژول در یک اپلیکیشن خاص مدنظر شما باشد دستور زیر نمونه آن است:

```
%windir%\system32\inetsrv\appcmd.exe delete module FormsAuthentication /app.name:"Default Web Site"
```

اگر قصد دارد آن را بر روی تمامی اپلیکیشن‌ها غیرفعال کنید، دستور زیر نمونه آن است:

```
%windir%\system32\inetsrv\appcmd.exe delete module FormsAuthentication
```

حفظ کردن یا به خاطر سپردن دستورات بالا ممکن است کار سخت و دشواری باشد، به همین جهت از `help` کمک بگیرید:

```
Appcmd.exe module /?
```

یا به شکل اختصاصی‌تر برای یک دستور

```
Appcmd.exe install module /?Appcmd add module /?
```

در حین بروز استثناهای Entity framework، می‌توان توسط ابزارهای Logging متنوعی مانند [ELMAH](#)، جزئیات متداول آن‌ها را برای بررسی‌های آتی ذخیره کرد. اما این جزئیات فاقد SQL نهایی تولیدی و همچنین پارامترهای ورودی توسط کاربر یا تنظیم شده توسط برنامه هستند. برای اینکه بتوان این جزئیات را نیز ثبت کرد، می‌توان یک IDbCommandInterceptor جدید را طراحی کرد.

EfExceptionsInterceptor کلاس

در اینجا نمونه‌ای از یک پیاده سازی اینترفیس IDbCommandInterceptor را مشاهده می‌کنید. همچنین طراحی یک متد عمومی که می‌تواند به جزئیات SQL نهایی و پارامترهای آن دسترسی داشته باشد، در اینترفیس IEfExceptionsLogger ذکر شده است.

```
public interface IEfExceptionsLogger
{
    void LogException<TResult>(DbCommand command,
                                DbCommandInterceptionContext<TResult> interceptionContext);
}

using System.Data.Common;
using System.Data.Entity.Infrastructure.Interception;

namespace ElmahEFLogger
{
    public class EfExceptionsInterceptor : IDbCommandInterceptor
    {
        private readonly IEfExceptionsLogger _efExceptionsLogger;

        public EfExceptionsInterceptor(IEfExceptionsLogger efExceptionsLogger)
        {
            _efExceptionsLogger = efExceptionsLogger;
        }

        public void NonQueryExecuted(DbCommand command, DbCommandInterceptionContext<int> interceptionContext)
        {
            _efExceptionsLogger.LogException(command, interceptionContext);
        }

        public void NonQueryExecuting(DbCommand command, DbCommandInterceptionContext<int> interceptionContext)
        {
            _efExceptionsLogger.LogException(command, interceptionContext);
        }

        public void ReaderExecuted(DbCommand command, DbCommandInterceptionContext<DbDataReader> interceptionContext)
        {
            _efExceptionsLogger.LogException(command, interceptionContext);
        }

        public void ReaderExecuting(DbCommand command, DbCommandInterceptionContext<DbDataReader> interceptionContext)
        {
            _efExceptionsLogger.LogException(command, interceptionContext);
        }

        public void ScalarExecuted(DbCommand command, DbCommandInterceptionContext<object> interceptionContext)
        {
            _efExceptionsLogger.LogException(command, interceptionContext);
        }

        public void ScalarExecuting(DbCommand command, DbCommandInterceptionContext<object> interceptionContext)
        {
            _efExceptionsLogger.LogException(command, interceptionContext);
        }
    }
}
```

تهیه یک پیاده سازی سفارشی از IEfExceptionsLogger از ELMAH توسط

اکنون که ساختار کلی IDbCommandInterceptor سفارشی برنامه مشخص شد، می‌توان پیاده سازی خاصی از آن را جهت استفاده از [ELMAH](#) به نحو ذیل ارائه داد:

```
using System;
using System.Data.Common;
using System.Data.Entity.Infrastructure.Interception;
using Elmah;

namespace ElmahEFLocator.CustomElmahLogger
{
    public class ElmahEfExceptionsLogger : IEfExceptionsLogger
    {
        /// <summary>
        /// Manually log errors using ELMAH
        /// </summary>
        public void LogException<TResult>(DbCommand command,
            DbCommandInterceptionContext<TResult> interceptionContext)
        {
            var ex = interceptionContext.OriginalException;
            if (ex == null)
                return;

            var sqlData = CommandDumper.LogSqlAndParameters(command, interceptionContext);
            var contextualMessage = string.Format("{0}{1}OriginalException:{1}{2} {1}", sqlData,
Environment.NewLine, ex);

            if (!string.IsNullOrWhiteSpace(contextualMessage))
            {
                ex = new Exception(contextualMessage, new ElmahEfInterceptorException(ex.Message));
            }

            try
            {
                ErrorSignal.FromCurrentContext().Raise(ex);
            }
            catch
            {
                ErrorLog.GetDefault(null).Log(new Error(ex));
            }
        }
    }
}
```

در اینجا شیء Command به همراه SQL نهایی تولید و پارامترهای مرتبط است. همچنین `interceptionContext.OriginalException` جزئیات عمومی استثنای رخداده را به همراه دارد. می‌توان این اطلاعات را پس از اندکی نظم بخشیدن، به متدهای [Raise](#) مربوط به ELMAH ارسال کرد تا جزئیات بیشتری از استثنای رخداده شده، در لگ‌های آن ظاهر شوند.

استفاده از ElmahEfExceptionsLogger جهت طراحی یک Interceptor عمومی

```
public class ElmahEfInterceptor : EfExceptionsInterceptor
{
    public ElmahEfInterceptor()
        : base(new ElmahEfExceptionsLogger())
    { }
```

برای استفاده از `EfExceptionsLogger` و تهیه یک `Interceptor` عمومی، می‌توان با ارث بری از کلاس `Interceptor` ابتدای بحث شروع کرد و ولهای از `ElmahEfExceptionsLogger` را به سازنده‌ی آن تزریق نمود (یکی از چندین روش ممکن). سپس برای

ثبت جزئیات استثناهای Entity framework توسط ELMAH

استفاده از آن کافی است به ابتدای متد Application_Start فایل Global.asax.cs مراجعه و در ادامه سطر ذیل را اضافه نمود:

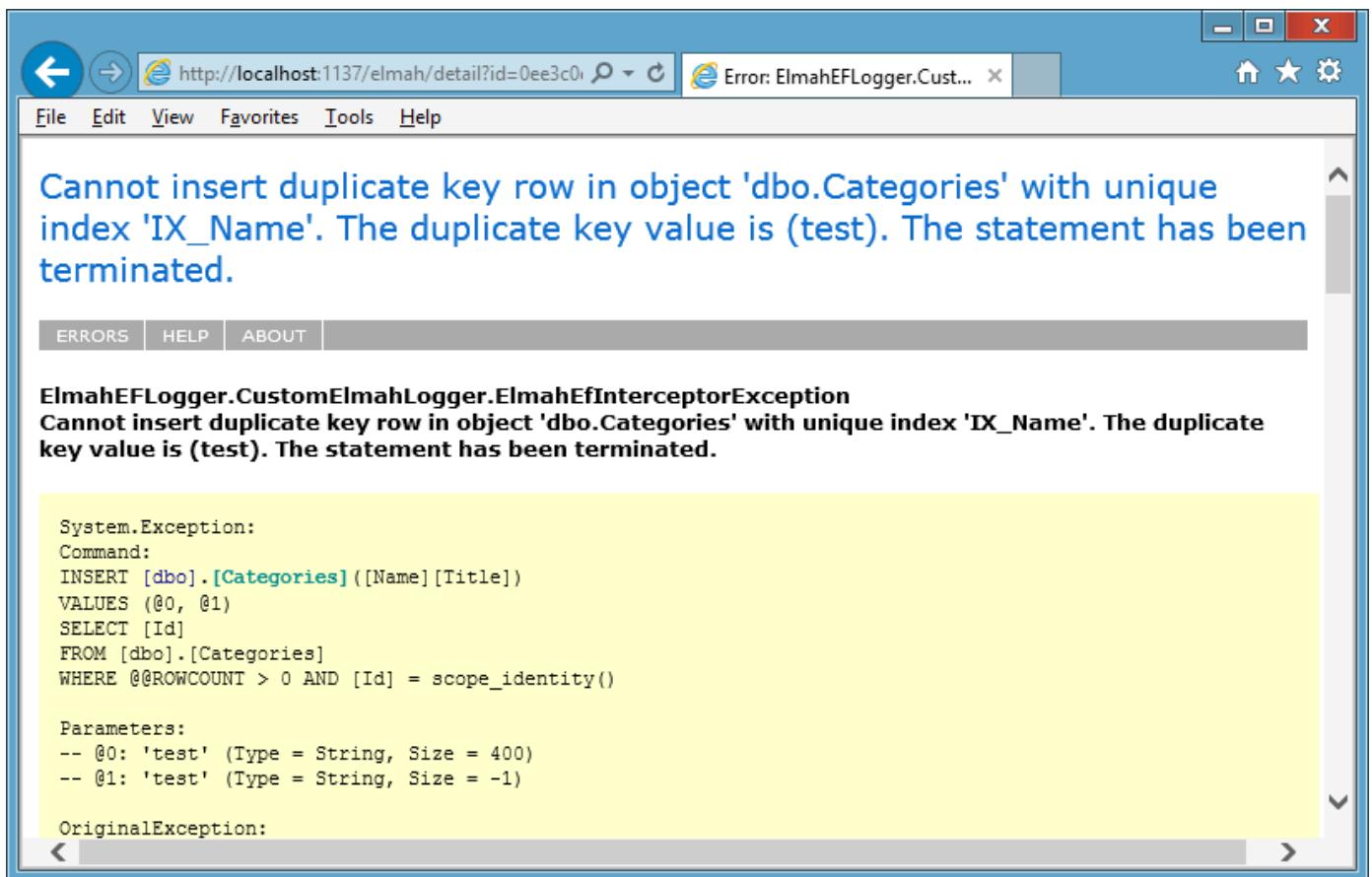
```
DbInterception.Add(new ElmahEfInterceptor());
```

پس از آن جزئیات کلیه استثناهای EF در لگ‌های نهایی ELMAH به نحو ذیل ظاهر خواهد شد:

The screenshot shows a web browser window displaying the ELMAH error log. The title bar says "Error log for /LM/W3SVC/4...". The main content area is titled "Error Log for ROOT on VAHIDPC". Below the title is a navigation bar with links for "RSS FEED", "RSS DIGEST", "DOWNLOAD LOG", "HELP", and "ABOUT". The main content area displays a table of errors. The table has columns for Host, Code, Type, Error, User, Date, and Time. There are seven entries in the table, all from the host VAHIDPC on December 29, 2014, at 10:27:49. The errors are categorized by type: Sql, ElmahEfInterceptor, and ElmahEfInterceptor. Most errors are related to duplicate key values in the 'dbo.Categories' table. The last three errors are related to invalid object names like 'dbo.EdmMetadata', 'dbo.__MigrationHistory', and 'dbo.__MigrationHistory'. Each error entry includes a "Details..." link.

Host	Code	Type	Error	User	Date	Time
VAHIDPC	0	Sql	Cannot insert duplicate key row in object 'dbo.Categories' with unique index 'IX_Name'. The duplicate key value is (test). The statement has been terminated. Details...		29/12/2014	10:27:49
VAHIDPC	0	ElmahEfInterceptor	Cannot insert duplicate key row in object 'dbo.Categories' with unique index 'IX_Name'. The duplicate key value is (test). The statement has been terminated. Details...		29/12/2014	10:27:49
VAHIDPC	0	ElmahEfInterceptor	Invalid object name 'dbo.EdmMetadata'. Details...		29/12/2014	10:27:49
VAHIDPC	0	ElmahEfInterceptor	Invalid object name 'dbo.__MigrationHistory'. Details...		29/12/2014	10:27:49
VAHIDPC	0	ElmahEfInterceptor	Invalid object name 'dbo.__MigrationHistory'. Details...		29/12/2014	10:27:49
VAHIDPC	0	ElmahEfInterceptor	Invalid object name 'dbo.__MigrationHistory'. Details...		29/12/2014	10:27:49

Powered by [ELMAH](#), version 1.2.14706.955. Copyright (c) 2004, Atif Aziz. All rights reserved. Licensed under [Apache License, Version 2.0](#). Server date is Monday, 29 December 2014. Server time is 10:27:49. All dates and times displayed are in the Iran Standard Time zone. This log is provided by the XML File-Based Error Log.



کدهای کامل این پروژه را از اینجا می‌توانید دریافت کنید:

[ElmahEFLogger](#)

نظرات خوانندگان

نویسنده: صابر فتح الهی
تاریخ: ۲۰:۴۷ ۱۳۹۳/۱۰/۰۸

با سلام و تشکر از مطلب شما

در صورت استفاده از پکیج بالا باید فیلتر کلی که در [پست آموزش MVC قسمت 16](#) توضیح دادین برای ثبت استثناهای elmah حذف بشود؟

نویسنده: وحید نصیری
تاریخ: ۲۱:۲۵ ۱۳۹۳/۱۰/۰۸

خیر. کار این Interceptor صرفا لاغ کردن سراسری و نامرئی ریز جزئیات استثناهای EF در کل برنامه است. استثناهای رخداده، در ادامه از فیلتر ELMAH اصلی تنظیم شده رد خواهند شد و یکبار هم در آنجا لاغ می‌شوند. این مورد چون منحصر به نیست، بنابراین ضرورتی به حذف آن هم نیست.

در این قسمت بیشتر یک سری از مازول‌ها را به شما در قالب جداول گروه بندی شده معرفی خواهیم کرد :

همانطور که در قسمت‌های قبلی گفتیم سرور IIS آماده خصوصی سازی و کار بر اساس علاقه شماست؛ ولی توجه داشته باشید حذف تمامی مازول‌ها ممکن است اثرات جانبی هم داشته باشد. در اینجا ما مازول‌هایی را به شما معرفی می‌کنیم که بدانید کار هر مازول چیست تا مثلاً با حذف مازولی، امنیت وب سایت خود را به خطر نیندازید :

مازول‌های سودمند یا utility

نام مازول:	
UriCacheModule	<p>توضیح: این مازول نوعی کش برای URL‌ها به شمار می‌رود. موقعی که درخواست می‌شود، اطلاعات در اولین درخواست خوانده شده و کش می‌شود و اگر دوباره همان url درخواست شود، بدون خواندن تنظیمات و بر اساس تنظیمات قبلی، کار url مربوطه را انجام میدهد تا اطلاعات پیکربندی تغییر کند و بر اساس اطلاعات جدید، خود را به روز کند.</p>
لازم ندارد	تگ قابل پیکربندی:
ندارد	وابستگی:
کارایی سیستم کاهش می‌یابد و سیستم مجبور است برای هر درخواست فایل پیکربندی را بخواند.	اثرات حذف آن:
نام مازول :	
FileCacheModule	<p>توضیح: فایل هندل فایل‌هایی که قبلاً در سرور باز شده‌اند را کش می‌کند تا در صورت نیاز در دفعات بعدی سریعتر عمل کند.</p>
لازم ندارد.	تگ قابل پیکربندی :
ندارد.	وابستگی :
کارایی سیستم کاهش می‌یابد. سیستم در هر اجرای دستور مربوط به فایل‌ها باید فایل هندل را به دست آورد.	اثرات حذف آن :

TokenCacheModule	نام مژول :
توکن‌های امنیتی ویندوز که پسوردهایی بر اساس anonymous authentication schemes هستند را کش می‌کند (authentication, basic authentication, IIS client (certificate authentication	توضیح :
لازم ندارد	تگ قابل پیکربندی :
ندارد	وابستگی :
کارایی سیستم به شدت پایین می‌آید. کاربر باید با هر درخواستی لاگین کند. یکی از اصلی‌ترین ضربه‌ها با حذف این مژول این است که اگر مثلاً یک پسورد از یک فایل html محافظت می‌کند و این صفحه به 50 تصویر ارجاع دارد، 51 بار باید درخواست لاگین اجرا گردد یا شاید هم بدتر	اثرات حذف آن :

MANAGED ENGINE: ASP.NET INTEGRATION

ManagedEngine	نام مژول :
مدیریت مژول‌های native و مدیریت شده	توضیح :
	تگ قابل پیکربندی :
ندارد	وابستگی :
مشخصاً غیرفعال شدن asp.net integrated و غیر فعال شدن تمامی مژول‌ها و هندرلهای تگ وب کانفیگ یا داخل فایل کانفیگ IIS که در مقالات قبلی به تفصیل بیان کردہ‌ایم.	اثرات حذف آن :

IIS 7 NATIVE MODULES

HttpCacheModule	نام مژول :
مدیریت کش خروجی در http.sys بر اساس پیکربندی مثل تعریف سایز کش و ...	توضیح :
System.webServer/caching	تگ قابل پیکربندی :

HttpCacheModule	نام مژول :
ندارد.	وابستگی :
محتوای دیگر به صورت کرنل مد، کش نمی‌شود و کش پروفایل هم نمیدید گرفته می‌شود و احتمالاً بر کارآیی و استفاده از منابع هم اثر می‌گذارد.	اثرات حذف آن :
DynamicCompressionModule	نام مژول :
پیاده سازی in-memory compression در محتوای پویا	توضیح :
system.webServer/httpCompression and system.webServer/urlCompression.	تگ قابل پیکربندی :
وابستگی ندارد چرا که به طور پیش فرض غیرفعال است.	وابستگی :

StaticCompressionModule	نام مژول :
پیاده سازی فشرده سازی در محتوای ایستا و برای فایل‌های سیستمی از نوع <i>in memory</i>	توضیح :
system.webServer/httpCompression and system.webServer/urlCompression	تگ قابل پیکربندی :
ندارد.	وابستگی :
در صورت عدم فشرده سازی بر مصرف ترافیک تاثیر گذار است.	اثرات حذف آن :
DefaultDocumentModule	نام مژول :
پیاده سازی یک لیست سند پیش فرض. درخواست‌ها مدام بشت سر هم می‌آیند و این درخواست‌های پشت سرهم، به سند پیش فرض هدایت می‌شوند. همان پنجره‌ای که شما به ترتیب فایل‌های index.htm, index.asp, default.aspx و ... را تعیین می‌کنید.	توضیح :
system.webServer/defaultDocument	تگ قابل پیکربندی :

StaticCompressionModule	نام ماژول :
ندارد.	وابستگی :
درخواست را به ریشه هدایت می‌کند. مثلاً برای <code>localhost</code> صفحه 404 باز می‌گرداند و اگر <code>directoryBrowsing</code> فعال باشد لیستی از دایرکتوری ریشه را باز می‌گرداند.	اثرات حذف آن :

DirectoryListingModule	نام ماژول :
پیاده سازی لیستی از محتویات یک دایرکتوری	توضیح :
<code>system.webServer/directoryBrowse</code>	تگ قابل پیکربندی :
ندارد.	وابستگی :
اگر این ماژول و ماژول قبلی غیرفعال باشند <code>response</code> نهایی خالی است.	اثرات حذف آن :
ProtocolSupportModule	نام ماژول :
پیاده سازی اختصاصی از <code>response header</code>	توضیح :
پیاده سازی تنظیمات <code>HTTP verbs</code> و <code>trace</code>	
پیاده سازی تنظیمات مربوطه به <code>keep-alive</code> بر اساس پیکربندی	
<code>system.webServer/httpProtocol</code>	تگ قابل پیکربندی :
ندارد.	وابستگی :
بازگرداندن پیام خطای "Method not allowed 405"	اثرات حذف آن :

HttpRedirectionModule	نام ماژول :
پیاده سازی عملیات انتقال یا <code>redirect</code>	توضیح :

HttpRedirectionModule	نام مازول :
system.webServer/httpRedirect	تگ قابل پیکربندی :
ندارد.	وابستگی :
خطر امنیتی: اگر منابعی با redirect کردن محافظت می‌شوند، از این پس در دسترسند.	اثرات حذف آن :
ServerSideIncludeModule	نام مازول :
حمایت از فایل shtml یا ...	توضیح :
system.webServer/serverSideInclude	تگ قابل پیکربندی :
ندارد.	وابستگی :
این فایل‌ها به صورت متنی نمایش داده می‌شوند	اثرات حذف آن :
StaticFileModule	نام مازول :
فایل‌های ایستا را به همراه پسوند ارسال می‌کند. مثل jpg,html و نوع محتوا را بر اساس staticContent/mimeMap پیکربندی می‌کند.	توضیح :
system.webServer/staticContent	تگ قابل پیکربندی :
ندارد.	وابستگی :
فایل‌های ایستا دیگر ارائه نشده و به جای آن خطای 404 بازگشت داده می‌شود.	اثرات حذف آن :
AnonymousAuthenticationModule	نام مازول :
پیاده سازی سیستم شناسایی افراد ناشناس. همانطور که میدانید در یک وب سایت حداقل محتوایی برای افرادی بدون داشتن اکانت هم وجود دارد. برای اینکار یک شیء httpUser ایجاد می‌کند.	توضیح :
	تگ قابل پیکربندی :

StaticFileModule	نام مازول :
system.webServer/security/authentication/anonymousAuthentication	وابستگی :
ندارد.	
حداقل باید یک سیستم امنیتی برای شناسایی یا authenticate وجود داشته باشد. httpuser یک ساختار داده ای در IIS میباشد و در صورت نبودن هیچ سیستم شناسایی وجود نداشته و در نبود شیء httpuser سیستم خطای 401.2 را تولید میکند.	اثرات حذف آن :
CertificateMappingAuthenticationModule	نام مازول :
مجوز SSL را به Active Directory نگاشت میکند.	توضیح :
system.webServer/security/authentication/clientCertificateMappingAuthentication	تگ قابل پیکربندی :
برای اینکه این مازول وظیفه خود را انجام دهد باید تنظیمات SSL انجام شود و همچنین سیستم IIS جزئی از دامنه Active directory باشد	وابستگی :
درخواست‌ها، نرمال رسیدگی میشوند انگار SSL وجود ندارد.	اثرات حذف آن :
BasicAuthenticationModule	نام مازول :
پیاده سازی پایه‌ای و روتین شناسایی کاربران بر اساس آن چیزی که در استاندارد زیر آمده است	توضیح :
. RFC 2617	
system.webServer/security/authentication/basicAuthentication	تگ قابل پیکربندی :
None.	وابستگی :
حداقل باید یک سیستم امنیتی برای شناسایی یا authenticate وجود داشته باشد. httpuser یک ساختار	اثرات حذف آن :

CertificateMappingAuthenticationModule	نام مازول :
داده‌ای در IIS می‌باشد و در صورت نبود، هیچ سیستم شناسایی یافت نشده و نبود شیء httpuser در سیستم، خطای 401.2 را تولید می‌کند.	
WindowsAuthenticationModule	نام مازول :
((windows Authentication (NTLM or Negotiate (Kerberos	توضیح :
system.webServer/security/authentication/windowsAuthentication	تگ قابل پیکربندی :
. ندارد.	وابستگی :
حداقل باید یک سیستم امنیتی برای شناسایی یا authenticate وجود داشته باشد. httpuser یک ساختار داده‌ای در IIS می‌باشد و در صورت نبود، هیچ سیستم شناسایی یافت نشده و نبود شیء httpuser در سیستم، خطای 401.2 را تولید می‌کند.	اثرات حذف آن :
DigestAuthenticationModule	نام مازول :
پیاده سازی سیستم شناسایی دیاجست بر اساس . RFC 2617	توضیح :
system.webServer/security/authentication/digestAuthentication	تگ قابل پیکربندی :
IIS باید بخشی از دامنه Active Directory باشد.	وابستگی :
حداقل باید یک سیستم امنیتی برای شناسایی یا authenticate وجود داشته باشد. httpuser یک ساختار داده‌ای در IIS می‌باشد و در صورت نبود، هیچ سیستم شناسایی یافت نشده و نبود شیء httpuser در سیستم، خطای 401.2 را تولید می‌کند.	اثرات حذف آن :

IISCertificateMappingAuthenticationModule	نام مژول :
پیاده سازی نگاشت مجوزهای IIS، نگهداری و ذخیره اطلاعات همه نگاشتها و مجوزهای کاربری چون SSL client certificates	توضیح :
system.webServer/iisClientCertificateMappingAuthentication	تگ قابل پیکربندی :
اطلاعات SSL به همراه دریافت client certificates جهت پیکربندی این مژول	وابستگی :
حداقل باید یک سیستم امنیتی برای شناسایی يا authenticate وجود داشته باشد. httpuser یک ساختار داده ای در IIS میباشد و در صورت نبود، هیچ سیستم شناسایی یافت نشده و نبود شیء httpuser در سیستم، خطای 401.2 را تولید میکند.	اثرات حذف آن :
UrlAuthorizationModule	نام مژول :
پیاده سازی authorization بر اساس قوانین پیکربندی شده	توضیح :
system.webServer/security/authorization	تگ قابل پیکربندی :
ندارد.	وابستگی :
محتواهای محافظت شده توسط authorization دیگر محافظت نمیشوند.	اثرات حذف آن :

IsapiModule	نام مژول :
پیاده سازی ISAPI Extension	توضیح :
system.webServer/isapiCgiRestriction	تگ قابل پیکربندی :
ندارد.	وابستگی :

IsapiModule	نام ماژول :
هندلرهای معرفی شده در بخش IsapiModule و تگ handlers دیگر اجرا نمی‌شوند	اثرات حذف آن :
IsapiFilterModule	نام ماژول :
پیاده سازی ISAPI filter	توضیح :
system.webServer/isapiFilters	تگ قابل پیکربندی :
ندارد.	وابستگی :
اگر برنامه‌ای از ISAPI filter استفاده می‌کند، در اجرا دچار مشکل خواهد شد.	اثرات حذف آن :

IpRestrictionModule	نام ماژول :
یک سیستم تشخیص دسترسی بر اساس آی پی‌های ورژن 4	توضیح :
system.webServer/security/ipSecurity	تگ قابل پیکربندی :
IPv4 stack باید نصب شود.	وابستگی :
کلاینت‌هایی که IP هایشان در IPsecurity در لیست شده‌اند ندید گرفته می‌شوند	اثرات حذف آن :
RequestFilteringModule	نام ماژول :
پیاده سازی یک مجموعه قدرتمند از قوانین امنیتی که درخواست‌های مشکوک را پس می‌زنند.	توضیح :
system.webServer/security/requestFiltering	تگ قابل پیکربندی :
ندارد.	وابستگی :
دیگر قوانین امنیتی اجرا نخواهند شد و سبب وجود مشکلات امنیتی می‌شود.	اثرات حذف آن :

CustomLoggingModule	نام ماژول :
پیاده سازی اینترفیس <code>ILogPlugin</code> در سمت IIS، به مشتریان اجازه میدهد تا لگ های خود را توسعه دهند. هر چند این روش توصیه نمی شود و توصیه کارشناس مایکروسافت استفاده از یک ماژول دست نویس از نوع <code>RQ_LOG_REQUEST</code> می باشد.	توضیح :
<code>Implements the ILogPlugin interface on top of IIS. ILogPlugin is a previous COM implementation that allowed customers to extend IIS logging. We do not recommend extending IIS using this interface. Instead, customers should write a module and subscribe to the .RQ_LOG_REQUEST notification</code>	
<code>system.webServer/httpLogging and system.applicationHost/sites/site/logFile/customLogPluginClsid</code>	تگ قابل پیکربندی :
ندارد.	وابستگی :
مسلمانه این اینترفیس از کار می افتد که سیستم ODBC Logging را جز آن است.	اثرات حذف آن :
CustomErrorModule	نام ماژول :
پیاده سازی مدیریت خطاهای ویژه	توضیح :
<code>system.webServer/httpErrors</code>	تگ قابل پیکربندی :
None.	وابستگی :
در صورتی که خطای از هسته باشد، نتیجه یک صفحه، با توضیح مختصری از خطا خواهد بود. در غیر این صورت اگر خطا از برنامه یا کامپوننتی باشد جزئیات خطا فاش خواهد شد	اثرات حذف آن :

HttpLoggingModule	نام ماژول :
پیاده سازی سیستم <code>logging</code> استاندارد <code>http.sys</code>	توضیح :
<code>system.applicationHost/log and</code>	تگ قابل پیکربندی :

HttpLoggingModule	نام مازول :
system.webServer/httpLogging	
ندارد.	وابستگی :
از کار افتادن سیستم لاغ	اثرات حذف آن :
FailedRequestsTracingModule	نام مازول :
پیاده سازی سیستم ردیابی درخواست‌های ناموفق و اجرای قوانین، طبق پیکربندی	توضیح :
system.webServer/tracing and system.webServer/httpTracing	تگ قابل پیکربندی :
ندارد.	وابستگی :
Tracing http requests will no longer work.	اثرات حذف آن :

RequestMonitorModule	نام مازول :
پیاده سازی IIS Run-time State and Control Interface یا RSCA به اختصار . به کاربران اجازه می‌دهد از اطلاعات، حين اجرا، کوئری بگیرند. مثل درخواست درحال اجرای جاری، آغاز به کار یا توقف وب سایت و دامنه‌های اپلیکیشن در حال اجرای جاری	توضیح :
ندارد.	تگ قابل پیکربندی :
ندارد.	وابستگی :
ابزارهای مرتبط با این موضوع از کار می‌افتد	اثرات حذف آن :
CgiModule	نام مازول :
پیاده سازی CGI در سمت IIS	توضیح :
system.webServer/cgi and system.webServer/isapiCgiRestriction	تگ قابل پیکربندی :

RequestMonitorModule	نام مژول :
ندارد.	وابستگی :
برنامه‌های CGI متوقف می‌شوند	اثرات حذف آن :

TracingModule	نام مژول :
پیاده سازی سیستم ردیابی ETW	توضیح :
system.webServer/httpTracing	تگ قابل پیکربندی :
ندارد.	وابستگی :
باعث از کار افتادن سیستم مربوطه می‌شود	اثرات حذف آن :
ConfigurationValidationModule	نام مژول :
اعتبارسنجی تنظیمات برنامه ASP.Net که به حالت integrate انتقال یافته است	توضیح :
system.webServer/Validation	تگ قابل پیکربندی :
ندارد.	وابستگی :
عدم اعتبارسنجی و در نتیجه عدم نمایش خطاهای	اثرات حذف آن :

:MANAGED MODULES

OutputCache	نام مژول :
output caching	توضیح :
system.web/caching/outputCache	تگ قابل پیکربندی :
. ManagedEngine نیاز به	وابستگی :
عدم اجرای output cache	اثرات حذف آن :

OutputCache	نام مازول :
Session	نام مازول :
مدیریت سشن ها	توضیح :
system.web/sessionState	تگ قابل پیکربندی :
. ManagedEngine نیاز به	وابستگی :
سشن ها از دسترس خارج می شوند.	اثرات حذف آن :

WindowsAuthentication	نام مازول :
اینجا	توضیح :
system.web/authentication	تگ قابل پیکربندی :
. ManagedEngine نیاز به	وابستگی :
این حالت قابل اجرا نخواهد بود	اثرات حذف آن :
FormsAuthentication	نام مازول :
اینجا	توضیح :
system.web/authentication	تگ قابل پیکربندی :
. ManagedEngine نیاز به	وابستگی :
این حالت قابل اجرا نیست و کاربران مجوز دار هم نمی توانند به منابع محافظت شده دسترسی داشته باشند.	اثرات حذف آن :

DefaultAuthentication	نام مازول :
اطمینان از وجود شی Authentication در context مربوطه	توضیح :
system.web/authentication	تگ قابل پیکربندی :
. ManagedEngine نیاز به	وابستگی :
اگر مد Forms authentication انتخاب شده باشد بر روی بعضی از کاربران ناشناس کار نخواهد کرد و رویداد DefaultAuthentication.OnAuthenticate اجرا نخواهد شد.	اثرات حذف آن :
RoleManager	نام مازول :
اینجا	توضیح :
	تگ قابل پیکربندی :
. ManagedEngine نیاز به	وابستگی :
این قابلیت در دسترس نمیباشد	اثرات حذف آن :
UrlAuthorization	نام مازول :
اینجا	توضیح :
system.web/authorization.	تگ قابل پیکربندی :
. ManagedEngine نیاز به	وابستگی :
باعث از کار افتادن asp.net authorization و فاش شدن بعضی اطلاعات و همچنین دیگر تهدیدات امنیتی	اثرات حذف آن :

UrlAuthorization	نام مازول :
AnonymousIdentification	نام مازول :
اینجا	توضیح :
	تگ قابل پیکربندی :
. ManagedEngine نیاز به	وابستگی :
The anonymous identification feature used by the ASP.NET Profile will not work.	اثرات حذف آن :

Profile	نام مازول :
اینجا	توضیح :
	تگ قابل پیکربندی :
ManagedEngine module must be installed.	وابستگی :
ASP.Net Profile از کار خواهد افتاد	اثرات حذف آن :
UrlMappingsModule	نام مازول :
تبديل یک Url واقعی به یک Url کاربرپسند	توضیح :
	تگ قابل پیکربندی :
. ManagedEngine نیاز به	وابستگی :
نگاشت Urlها صورت نمی‌گیرد	اثرات حذف آن :

پس از بررسی مفاهیم، بهتر هست وارد یک کار عملی شویم. مثال مورد نظر، یک مثال از وب سایت شرکت مایکروسافت است که هنگام نمایش تصاویر، بر حسب پیکربندی موجود، یک پرچسب یا تگ را در گوشاهی از تصویر درج می‌کند. البته تصویر را ذخیره نمی‌کنیم و تگ را بر روی تصویر اصلی قرار نمی‌دهیم. تنها هنگام نمایش به کاربر، روی response خروجی آن را درج می‌کنیم.

قبلما در این [مقاله](#) به بررسی `httpandler` پرداخته‌ایم، ولی بهتر هست در این مثال کمی حالت پیشرفته‌تر آن را بررسی کنیم.

ابتدا اجازه دهید کمی قابلیت‌های فایل کانفیگ IIS را گسترش دهیم.
مسیر زیر را باز کنید:

```
%windir%\system32\inetsrv\config\schema
```

یک فایل `xml` را با نام `imagecopyright.xml` ساخته و تگ‌های زیر را داخلش قرار دهید:

احتمال زیاد دسترسی برای ویرایش این دایرکتوری به خاطر مراتب امنیتی با مشکل برخواهد خورد برای ویرایش این نکته امنیتی از [اینجا](#) یا به خصوص از [اینجا](#) کمک بگیرید.

```
<configSchema>
```

```
  <sectionSchema name="system.webServer/imageCopyright">
    <attribute name="enabled" type="bool" defaultValue="false" />
    <attribute name="message" type="string" defaultValue="Your Copyright Message" />
    <attribute name="color" type="string" defaultValue="Red"/>
  </sectionSchema>
</configSchema>
```

با این کار ما یک شیما یا اسکیما را ایجاد کردیم که دارای سه خصوصیت زیر است:
`enabled`: آیا این هندرلر فعال باشد یا خیر.

`message`: پیامی که باید به عنوان تگ درج شود.
`color`: رنگ متن که به طور پیش فرض قرمز رنگ است.

به هر کدام از تگ‌های بالا یک مقدار پیش فرض داده ایم تا اگر مقداردهی نشدنده، مازول طبق مقادیر پیش فرض کار خود را انجام هد.

بعد از نوشتن شما، لازم هست که آن را در فایل `applicationhost.config` نیز به عنوان یک `section` جدید در زیر مجموعه `system.webserver` معرفی کنیم:

```
<configSections>
```

```
...
  <sectionGroup name="system.webServer">
    <section name="imageCopyright" overrideModeDefault="Allow"/>
  </sectionGroup>
</configSections>
```

تعريف کد بالا به شما اجازه میدهد تا در زیر مجموعه تگ `system.webserver`، برای هندرلر خود تگ تعريف کنید. در کد بالا، شمای خود را بر اساس نام فایل مشخص می‌کنیم و خصوصیت `overrideModeDefault`، یک قفل گذار امنیتی برای تغییر محتوای است. در صورتی که `allow` باشد هر کسی در هر مرحله‌ی دسترسی در سیستم و در هر فضای نامی، در فایل‌های وب کانفیگ می‌تواند به مقادیر این `section` دسترسی یافته و آن‌ها را تغییر دهد. ولی اگر با `Deny` مقداردهی شده باشد، مقادیر قفل شده و هیچ دسترسی برای تغییر آن‌ها وجود ندارد.

در مثال زیر ما به مازول `windows Authentication` اجازه می‌دهیم که هر کاربری در هر سطح دسترسی به این `section`

دسترسی داشته باشد؛ از تمامی سایت‌ها یا اپلیکشین‌ها یا virtual directories موجود در سیستم و در بعضی موارد این گزینه باعث افزایش ریسک امنیتی می‌گردد.

```
<section name="windowsAuthentication" overrideModeDefault="Allow" />
```

در کد زیر اینبار ما دسترسی را بستیم و در تعاریف دامنه‌های دسترسی، دسترسی را فقط برای سطح مدیریت سایت AdministratorSite باز گذاشته‌ایم:

```
<location path="AdministratorSite" overrideMode="Allow">
  <security>
    <authentication>
      <providers>
        <windowsAuthentication enabled="false">
          </providers>
          <add value="Negotiate" />
          <add value="NTLM" />
    </location>
    </windowsAuthentication>
  </authentication>
</security>
```

برای خارج نشدن بیش از اندازه از بحث، به ادامه تعریف هندر می‌پردازیم. بعد از معرفی یک section برای هندر خود، میتوانیم به راحتی تگ آن را در قسمت system.webserver تعریف کنیم. این کار می‌تواند از طریق فایل web.config سایت یا applicationhost.config صورت بگیرد یا میتواند از طریق ویرایش دستی یا خط فرمان appcmd معرفی شود؛ ولی در کل باید به صورت زیر تعریف شود:

```
<system.webServer>
  <imageCopyright />
</system.webServer>
```

در کد بالا این تگ تنها معرفی شده است؛ ولی مقادیر آن پیش فرض می‌باشند. در صورتی که بخواهید مقادیر آن را تغییر دهید کد به شکل زیر تغییر می‌کند:

```
<system.webServer>
  <imageCopyright enabled="true" message="an example of www.dotnettips.info" color="Blue" />
</system.webServer>
```

در صورتی که میخواهید از خط فرمان کمک بگیرید به این شکل بنویسید:

```
%windir%\system32\inetsrv\appcmd set config -section:system.webServer/imageCopyright /color:yellow
/message:"Dotnettips.info" /enabled:true
```

برای اطمینان از این که دستور شما اجرا شده است یا خیر، یک کوئری یا لیست از تگ مورد نظر در system.webserver بگیرید:

```
%windir%\system32\inetsrv\appcmd list config -section:system.webServer/imageCopyright
```

در این مرحله یک دایرکتوری برای پروژه تصاویر ایجاد کنید و در این مثال ما فقط تصاویر jpg را ذخیره می‌کنیم و در هنگام درج تگ، تصاویر jpg را هندر می‌کنیم؛ برای مثال ما:

```
c:\inetpub\mypictures
```

در این مرحله دایرکتوری ایجاد شده را به عنوان یک application معرفی می‌کنیم:

```
%windir%\system32\inetsrv\appcmd add app -site.name:"Default Web Site" -path:/mypictures -
physicalPath:%systemdrive%\inetpub\mypictures
```

و برای آن مازول DirectoryBrowse را فعال می‌کنیم. برای اطلاعات بیشتر به [مقاله قبلی](#) که به تشریح وظایف مازول‌ها پرداختیم رجوع کنید. فقط به این نکته اشاره کنم که اگر کاربر آدرس localhost/mypictures را درخواست کند، فایل‌های این قسمت را برای ما لیست می‌کند. برای فعال سازی، کد زیر را فعال می‌کنیم:

```
%windir%\system32\inetsrv\appcmd set config "Default Web Site/mypictures" -section:directoryBrowse -
enabled:true
```

حال زمان این رسیده است تا کد نوشته و فایل cs آن را در مسیر زیر ذخیره کنیم:

```
c:\inetpub\mypictures\App_Code\imagecopyrighthandler.cs
```

هندل مورد نظر در زبان سی شارپ :

```
#region Using directives
using System;
using System.Web;
using System.Drawing;
using System.Drawing.Imaging;
using Microsoft.Web.Administration;
#endregion

namespace IIS7Demos
{
    public class imageCopyrightHandler : IHttpHandler
    {
        public void ProcessRequest(HttpContext context)
        {
            ConfigurationSection imageCopyrightHandlerSection =
                WebConfigurationManager.GetSection("system.webServer/imageCopyright");

            HandleImage(    context,
                            (bool)imageCopyrightHandlerSection.Attributes["enabled"].Value,
                            (string)imageCopyrightHandlerSection.Attributes["message"].Value,
                            (string)imageCopyrightHandlerSection.Attributes["color"].Value
                            );
        }

        void HandleImage(    HttpContext context,
                            bool enabled,
                            string copyrightText,
                            string color
                            )
        {
            try
            {
                string strPath = context.Request.PhysicalPath;
                if (enabled)
                {
                    Bitmap bitmap = new Bitmap(strPath);
                    // add copyright message
                    Graphics g = Graphics.FromImage(bitmap);
                    Font f = new Font("Arial", 50, GraphicsUnit.Pixel);
                    SolidBrush sb = new SolidBrush(Color.FromName(color));
                    g.DrawString(    copyrightText,
                                    f,
                                    sb,
                                    5,
                                    bitmap.Height - f.Height - 5
                                    );
                    f.Dispose();
                    g.Dispose();
                    // slow, but good looking resize for large images
                    context.Response.ContentType = "image/jpeg";
                    bitmap.Save(
                                context.Response.OutputStream,
                                System.Drawing.Imaging.ImageFormat.Jpeg
                                );
                    bitmap.Dispose();
                }
                else
                {
                    context.Response.WriteFile(strPath);
                }
            }
            catch (Exception e)
            {
                context.Response.Write(e.Message);
            }
        }

        public bool IsReusable
        {
            get { return true; }
        }
    }
}
```

در خط `WebConfigurationManager.GetSection`، در صورتیکه تگ `imagecopyright` تعریف شده باشد، همه اطلاعات این تگ را از فایل کانفیگ بپردازیم و داخل شیء `imageCopyrightHandlerSection` از نوع `ImageCopyrightHandlerSection` قرار می‌دهیم. سپس اطلاعات هر سه گزینه را خوانده و به همراه `context` (اطلاعات درخواست) به تابع `handleImage` که ما آن را نوشته ایم ارسال می‌کنیم. کار این تابع درج تگ می‌باشد.

در خطوط اولیه تابع، ما آدرس فیزیکی منبع درخواست شده را به دست آورده و در صورتیکه مقدار گزینه `enable` با `true` مقدار دهی شده باشد، آن را به شیء `bitmap` نسبت می‌دهیم و با استفاده از دیگر کلاس‌های گرافیکی، تگ مورد نظر را با متن و رنگ مشخص شده ایجاد می‌کنیم. در نهایت شیء `bitmap` را ذخیره و نوع خروجی `response` را از نوع `image/jpeg` تعریف می‌کنیم تا مرورگر بداند که خروجی ما یک تصویر است. ولی در صورتی که `enabled` با `false` مقداردهی شده باشد، همان تصویر اصلی را بدون درج تگ ارسال می‌کنیم.

فضای نام `Microsoft.Web.Administration` برای اجرای خود نیاز دارد تا اسمبلی آن رفرنس شود. برای اینکار به درون دایرکتوری `mypictures` رفته و در داخل فایل `web.config` که بعد از تبدیل این دایرکتوری به اپلیکیشن ایجاد شده بنویسید:

```
<system.web>
  <compilation>
    <assemblies>
      <add assembly="Microsoft.Web.Administration, Version=7.0.0.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35, processorArchitecture=MSIL"/>
    </assemblies>
  </compilation>
</system.web>
```

در صورتی که کلاس خود را کامپایل کنید می‌توانید آن را داخل پوشه‌ی `App_Code` قرار دهید و نیاز به رفرنس کرده اسمبلی `Microsoft.Web.Administration` نیز ندارید.

در آخرین مرحله فقط باید به IIS بگویید که تنها فایل‌های `jpg` را برای این هندر، هندر کن. این کار را از طریق خط فرمان نجام می‌دهیم:

```
appcmd set config "Default Web Site/mypictures/" -section:handlers
/+[name='JPGImageCopyrightHandler',path='*.jpg',verb='GET',type='IIS7Demos.imageCopyrightHandler']
```

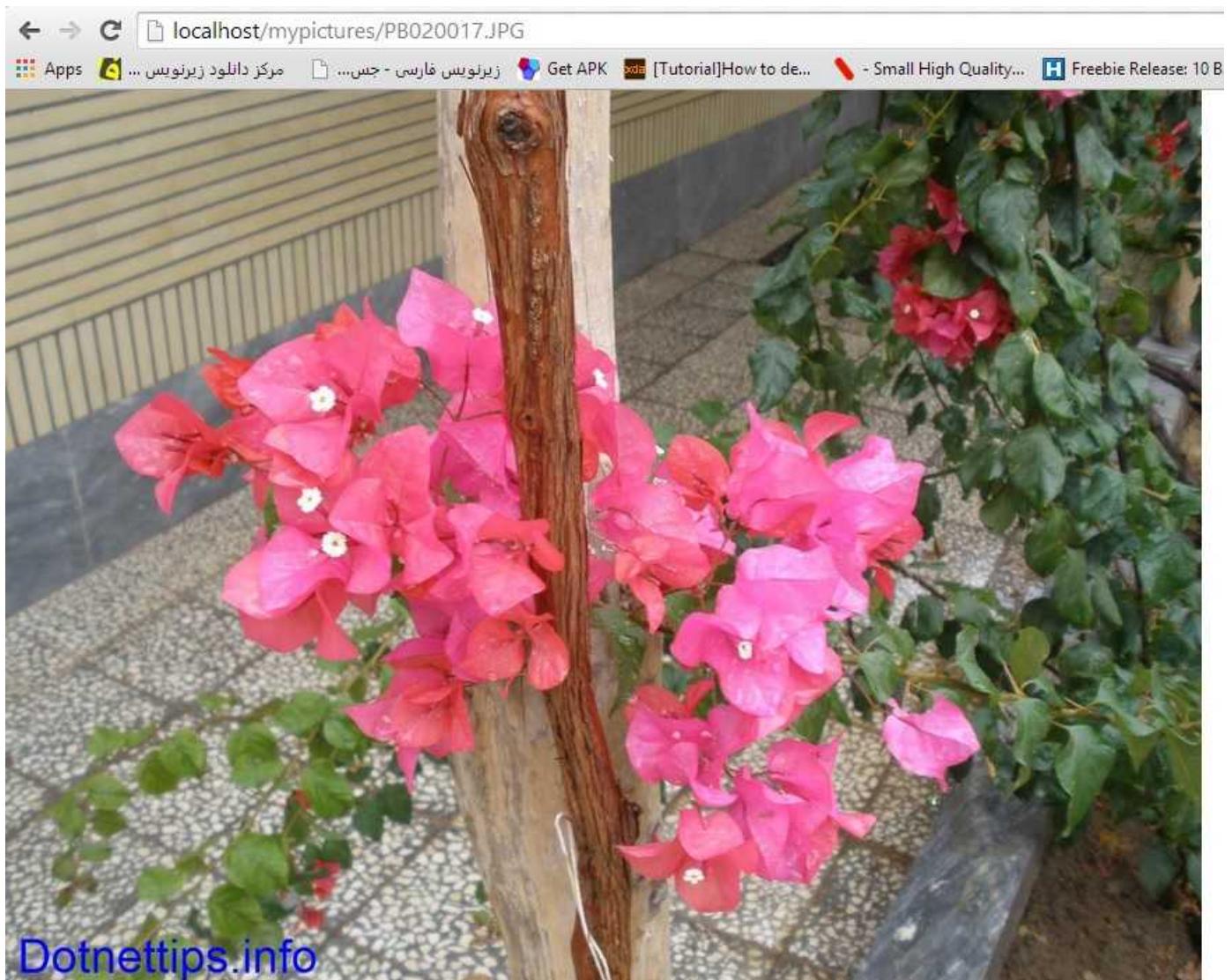
هندر مورد نظر تنها برای این اپلیکیشن و در مسیر `mypicture` فعال شده و در قسمت `name`، یک نام اختیاری بدون فاصله و `unique` بر می‌گذاریم. در قسمت `path` نوع فایل‌هایی را که نیاز به هندر هست، مشخص کردیم و در قسمت `verb` گفته‌ایم که تنها برای درخواست‌های نوع GET، هندر را اجرا کن و در قسمت `type` هم که اگر [مقاله](#) را خوانده باشید می‌دانید که به معرفی هندر می‌پردازیم؛ اولی نام فضای نام هست و بعد از نام کلاس، که در اینجا می‌شود :

```
'IIS7Demos.imageCopyrightHandler'
```

الان همه چیز برای اجرا آماده است و فقط یک مورد برای احتیاط الزامی است و آن هم این است که پروسه‌های کارگر، ممکن است از قبل در حال اجرا بوده باشند و هنوز شمای جدید ما را شناسایی نکرده باشند، برای همین باید آن‌ها را با تنظیمات جدیدمان آشنا کنیم تا احیانا برایمان استثناء صادر نشود:

```
appcmd recycle AppPool DefaultAppPool
```

کارمان تمام شده، چند تصویر داخل دایرکتوری قرار داده و درخواست تصاویر موجود را بدھید تا تگ را ببینید:



فعلا تا بدینجا کافی است. در قسمت آینده این هندر را کمی بیشتر توسعه خواهیم داد.

همانطور که از نامش پیداست `j.cookie` یک پلاگین `jquery` است. این پلاگین به شما این اجازه را می‌دهد تا هر نوع داده‌ای را که مایل هستید از قبیل رشته‌ها، آرایه‌ها و `object` را در قالب `json` با رمزگذاری `base 64` ذخیره نمایید. استفاده از این رمزگذاری باعث کوچکتر شدن حجم کوکی تا ۷۰ درصد می‌شود. در این مقاله شما یاد می‌گیرید که چطور برای ذخیره و بازیابی کوکی از آن استفاده کرده و چگونه در یک زبان سمت سرور، مثل سی شارپ نیز کوکی مورد نظر را با همان فرمت بخوانید. جهت دانلود فایل `j.cookie` به [اینجا](#) مراجعه کنید.

برای ساخت یک کوکی به روش زیر اقدام می‌کنیم. استفاده از `$.jCookies` دو خاصیت به نام‌های `name` کوکی و `value` مقدار کوکی را در دسترس می‌گذارد:

```
var d = new Date();
$.jCookies({
  name: 'dotnettips.info',
  value: { Title: 'jCookie ساخت کوکی با jCookie', Author: 'علی یگانه مقدم', Seen: d.getDate(), Favorite: true }
});
```

همانطور که می‌بینید ذخیره اطلاعات توسط `j.cookie` بسیار ساده و راحت بوده و هر نوع داده‌ای در آن به راحتی قابل ذخیره سازیست. برای مثال می‌توانید اطلاعات یک کلاس را خیلی راحت و سریع با آن ذخیره کنید. به طور پیش فرض تاریخ انقضای کوکی ۲۷ روز بعد از ایجاد آن می‌باشد. در صورتی که تمایل دارید این تاریخ را تغییر دهید یکی از خاصیت‌های `seconds, minutes, hours, days` در دسترس شماست و مقادیری که جلوی آن‌ها به کارگرفته می‌شود باید نوع صحیح بوده و در صورتی که مقدار نامعتبر وارد شود خاصیت مورد نظر نادیده گرفته می‌شود.

```
$.jCookies ({ name : 'User', value : { username : 'Bob' , level : 5 } , minutes : 60 });
```

برای تغییر پیش فرض‌های ساخت کوکی مانند انقضای ۲۷ روز به عدد پیش فرض خودتان فایل `jcookies.js` را باز کرده و تنظیمات پیش فرض آن را تغییر بدهید. برای تغییر دنبال کد زیر بگردید:

```
$.jCookies.defaults =
{
  name : '',
  value : '',
  days : 27
}
```

بازیابی کوکی

برای بازیابی کوکی مجدداً از `$.jCookies` استفاده می‌شود ولی تنها باید یک خاصیت `get` که نام کوکی هست را بنویسید:

```
var values = $.jCookies ({ get: 'dotnettips.info' });
```

در صورتی که نام کوکی‌ای که درخواست کرده آید وجود نداشته باشد یا اینکه تاریخ انقضای آن سر رسیده است و از سیستم کلاینت حذف شده است یا اینکه هنگام درخواست کوکی با خطای مواجه شده باشد، مقداً برگشتی `false` خواهد بود و اگر نیاز دارید که بدانید آیا نوع برگشتی `false` به خاطر خطأ بوده است یا خیر یک خاصیت نوع بول به نام `error` هم اضافه می‌شود:

```
var values = $.jCookies({ get: 'Rutabaga', error: true });
```

در صورتی که خطای داده شود `response` مقدار `values` در مرورگر کروم به شکل زیر خواهد بود. در هر مرورگر نحوه نمایش خطأ می‌تواند متفاوت باشد.

```
Error : {
  arguments : undefined,
  message : "Invalid base64 data",
  stack : "-",
}
```

ایجاد کوکی با jcookie

```
} type : undefined
```

بازیابی همه کوکی ها

در صورتی که به خاصیت get مقدار * را بدھید تمامی کوکی ها برگشت داده خواهند شد و به صورت آرایه ای از نام کوکی ها در دسترسی خواهند بود:

```
var values = $.jCookies({ get: '*' });
alert(values["dotnettips.info"].Title);
alert(values["data2"].Title);
```

حذف کوکی

نحوه کدنویسی حذف کوکی هم دقیقا مشابه خواندن کوکی است: با این تفاوت که به جای استفاده از خاصیت get از خاصیت erase استفاده می کنیم و با دادن نام کوکی به این خاصیت، کوکی حذف خواهد شد:

```
var value = $.jCookies({ erase: 'dotnettips.info' });
```

در صورتی که کوکی وجود داشته باشد، آن را حذف کرده و مقدار true را برگشت خواهد زد و در صورتی که کوکی وجود نداشته باشد مقدار false را بر میگرداند.

بازیابی کوکی در سمت سرور با سی شارپ

در این روش ما ابتدا با همان دستور معمولی دات نت یعنی page.request.cookie درخواست دریافت کوکی را می دهیم ولی از آنجا که در jcookie دو عمل روی داده ها صورت گرفته است باید دو کار اضافه تر را انجام داد:
برگشت داده ها از حالت رمزگذاری base64
داده ها در فرمت json هستند و باید به حالت قابل استفاده در محیط شی گرا تبدیل شوند.

برای بازگردانی از حالت base64 از کلاس و متod Convert.FromBase64String در فضای نام system.convert استفاده می کنیم که آرایه ای از نوع بایت را بر میگرداند و از Encoding.UTF8.GetString هم برای decode کردن آرایه به نوع رشته استفاده می کنیم. تا به اینجای کار داده های ما به صورت یک json خوانا با فرمت string درآمده است. برای داده های موجود در این فرمت باید آنها را Deserialize کنیم که این کار را از طریق کلاس JavaScriptSerializer در فضای نام System.Web.Script.Serialization انجام می دهیم و از کلاس دیکشنری برای ذخیره داده های برگشتی استفاده می کنیم که نوع string برای نام خاصیت و نوع آبجکت برای ذخیره مقدار خاصیت خواهد بود. یعنی برای بازگردانی اولین مثال بالا باید داده های در نوع دیکشنری به صورت زیر لیست شوند

jcookie	Title
علی یگانه مقدم	Author
2015/1/14	Seen
true	Favorite

```
byte[] from64 = Convert.FromBase64String(Page.Request.Cookies["dotnettips.info"].Value);
string json = Encoding.UTF8.GetString(from64);
Dictionary<string, object> article =new
JavaScriptSerializer().Deserialize<Dictionary<string, object>>(json);
Page.Response.Write("Title: " + (string)article["Title"]);
```

پشتیبانی از یونیکد

موقعی که من اولین مثال بالا را نوشتیم و مقادیر را به صورت فارسی وارد کردم متوجه شدم که رشته های یونیکد را انکود نمی کند و در نتیجه زبان فارسی در آن پشتیبانی نمی شود. برای همین تغییراتی در فایل js ایجاد کرده و عبارت value قبل از تبدیل به utf-16 انکود کردم و در هنگام خواندن کوکی هم به صورت utf-16 دیکود کردم و مشکل زبان فارسی هم در این حالت حل شد. البته کدی که اضافه کردم قابلیت های انکودینگ بیشتری هم دارد.
 فقط تنها مورد این هست که برای خواندن کوکی در سمت سرور باید یک تغییر کوچک یک کلمه ای بدھیم: باید کلمه UTF8 را به

که می‌شود همان UTF-16 در کد تغییر دهیم، که به کد زیر تغییر خواهد یافت:

```
byte[] from64 = Convert.FromBase64String(Page.Request.Cookies["dotnettips.info"].Value);
        string json = Encoding.Unicode.GetString(from64);
        Dictionary<string, object> article =new
JavaScriptSerializer().Deserialize<Dictionary<string, object>>(json);
Page.Response.Write("Title: " + (string)article["Title"]);
```

برای دریافت jcookie با پشتیبانی از زبان فارسی به [اینجا](#) مراجعه کنید.
کدهای بالا در فایل زیر قرار گرفته اند. [jcookie.zip](#)

یکی از وب سرویس‌های سایت [name api](#)، امکان [تشخیص موقتی بودن ایمیل](#) مورد استفاده‌ی جهت ثبت نام در یک سایت را فراهم می‌کند. آدرس WSDL آن نیز [در اینجا](#) قرار دارد. اگر مطابق معمول استفاده از سرویس‌های وب در دات نت، بر روی ارجاعات پروژه کلیک راست کرده و گزینه‌ی Add service reference را انتخاب کنیم و سپس آدرس WSDL یاد شده را به آن معرفی کنیم، بدون مشکل ساختار این وب سرویس دریافت و برای استفاده‌ی از آن به یک چنین کدی خواهیم رسید:

```
var client = new SoapDisposableEmailAddressDetectorClient();
var context = new soapContext
{
    //todo: get your API key here: http://www.nameapi.org/en/register/
    apiKey = "test"
};
var result = client.isDisposable(context, "DaDiDoo@mailinator.com");
if (result.disposable.ToString() == "YES")
{
    Console.WriteLine("YES! It's Disposable!");
}
```

متد `isDisposable` ارائه شده‌ی توسط این وب سرویس، دو پارامتر `context` که در آن باید [API Key](#) خود را مشخص کرد و همچنین آدرس ایمیل مورد بررسی را دریافت می‌کند. اگر به همین ترتیب این پروژه را اجرا کنید، با خطای Bad request از طرف سرور متوقف خواهید شد:

Additional information: The remote server returned an unexpected response: (400) Bad Request.

اگر به خروجی این وب سرویس در [فیدلر](#) مراجعه کنیم، چنین شکلی را خواهد داشت:

```
<html><head><title>Bad Request</title></head><body><h1>Bad Request</h1><p>No api-key provided!</p></body></html>
```

عنوان کرده‌است که api-key را، در درخواست وب خود ذکر نکرده‌ایم.

اگر همین وب سرویس را توسط امکانات سایت wsdlbrowser.com بررسی کنید، بدون مشکل کار می‌کند. اما تفاوت در کجاست؟

خروجی ارسالی به سرور، توسط سایت <http://wsdlbrowser.com> به این شکل است:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<ns1:isDisposable>
<context>
<apiKey>test</apiKey>
</context>
<emailAddress>sdsdg@site.com</emailAddress>
</ns1:isDisposable>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

و نمونه‌ی تولید شده‌ی توسط WCF (امکان Add service reference یک WCF Client را ایجاد می‌کند) به صورت زیر می‌باشد:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
<s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<xsd:isDisposable>
```

```

xmlns="http://disposableemailaddressdetector.email.services.v4_0.soap.server.nameapi.org/">
    <context xmlns=""><apiKey>test</apiKey></context>
    <emailAddress xmlns="">DaDiDoo@mailinator.com</emailAddress>
    </isDisposable>
</s:Body>
</s:Envelope>

```

از لحاظ اصول XML، خروجی تولیدی توسط WCF هیچ ایرادی ندارد. از این جهت که نام فضای نام مرتبط با را به `s` تنظیم کرده است و سپس با استفاده از این نام، `Envelope` را تشکیل داده است. اما ... این وب سرور جاوایی دقیقاً با نام SOAP-ENV کار می‌کند و فضای نام `ns1` بعدی آن. کاری هم به اصول XML ندارد که باید بر اساس نام `xmlns` ذکر شده، کار ورودی دریافتی صورت گیرد و نه بر اساس یک رشته‌ی ثابت از پیش تعیین شده. بنابراین باید راهی را پیدا کنیم تا بتوان این `s` را تبدیل به SOAP-ENV کرد.

برای این منظور به سه کلاس ذیل خواهیم رسید:

```

public class EndpointBehavior : IEndpointBehavior
{
    public void AddBindingParameters(ServiceEndpoint endpoint, BindingParameterCollection
bindingParameters)
    {
    }

    public void ApplyDispatchBehavior(ServiceEndpoint endpoint, EndpointDispatcher endpointDispatcher)
    {
    }

    public void Validate(ServiceEndpoint endpoint)
    {
    }

    public void ApplyClientBehavior(ServiceEndpoint endpoint, ClientRuntime clientRuntime)
    {
        clientRuntime.MessageInspectors.Add(new ClientMessageInspector());
    }
}

public class ClientMessageInspector : IClientMessageInspector
{
    public void AfterReceiveReply(ref Message reply, object correlationState)
    {
    }

    public object BeforeSendRequest(ref Message request, System.ServiceModel.IClientChannel channel)
    {
        request = new MyCustomMessage(request);
        return request;
    }
}

/// <summary>
/// To customize WCF envelope and namespace prefix
/// </summary>
public class MyCustomMessage : Message
{
    private readonly Message _message;

    public MyCustomMessage(Message message)
    {
        _message = message;
    }

    public override MessageHeaders Headers
    {
        get { return _message.Headers; }
    }

    public override MessageProperties Properties
    {
        get { return _message.Properties; }
    }

    public override MessageVersion Version
    {
        get { return _message.Version; }
    }

    protected override void OnWriteStartBody(XmlDictionaryWriter writer)
    {
    }
}

```

```
writer.WriteStartElement("Body", "http://schemas.xmlsoap.org/soap/envelope/");  
}  
  
protected override void OnWriteBodyContents(XmlDictionaryWriter writer)  
{  
    _message.WriteBodyContents(writer);  
}  
  
protected override void OnWriteStartEnvelope(XmlDictionaryWriter writer)  
{  
    writer.WriteStartElement("SOAP-ENV", "Envelope", "http://schemas.xmlsoap.org/soap/envelope/");  
    writer.WriteAttributeString("xmlns", "ns1", null, value:  
"http://disposableemailaddressdetector.email.services.v4_0.soap.server.nameapi.org/");  
}
```

که پس از تعریف `client` به نحو ذیل معرفی می‌شوند:

```
var client = new SoapDisposableEmailAddressDetectorClient();  
client.EndpointBehaviors.Add(new EndpointBehavior());
```

توسط `EndpointBehavior` سفارشی، می‌توان به متده `OnWriteStart Envelope` دسترسی یافت و سپس آنرا با `SOAP-ENV` درخواستی این وب سرویس جایگزین کرد. اکنون اگر برنامه را اجرا کنید، بدون مشکل کار خواهد کرد و دیگر پیام یافت نشدن `API-Key` را صادر نمی‌کند.

کدهای کامل این مثال را [از اینجا](#) می‌توانید دریافت کنید.

در اکثر برنامه های سازمانی، مثل برنامه های مدیریت آرشیو استناد، همواره این نیاز جزو خواسته های کاربران بوده که بتوانند به صورت مستقیم و از طریق تنها یک کلیک، تصویر مورد نظر را اسکن کرده و به صورت خودکار در برنامه وارد کنند؛ یعنی بدون اینکه نیاز باشد با استفاده از یک برنامه دیگر ابتدا تصویر را اسکن کرده و سپس در فرم وب، فایل اسکن شده را Browse کنند. این نیاز اساساً به معنی دسترسی به سخت افزار کاربر از طریق مرورگر می باشد که به دلایل متعددی امکان پذیر نیست! مشکلات امنیتی ایجاد شده در این راه بسیار جدی است. اما خوشبختانه راههایی برای رسیدن به این هدف وجود دارند:

1- ActiveX : که به صورت native فقط در IE پشتیبانی می شود (در نسخه های جدیدتر IE نیاز به [بروز رسانی پلاگین ActiveX](#) می باشد) و برای استفاده از آن در مرورگرهای Chrome و Firefox هم باید از پلاگین های جانبی روی هر مرورگر استفاده کرد که مثلا برای اجرای بر روی Firefox، باید افزونه [IE Tab](#) را نصب کرد که تنها کارش این است که سایت را از طریق موتور IE در پنجره فایرفاکس اجرا کند! که البته این مورد مثل این می ماند که سایت در IE باز شده باشد که ممکن است زیاد خوشایند نباشد؛ در غیر اینصورت باید پروژه را از اول بر مبنای اجرای بر روی IE طراحی و پیاده سازی کرد. در ضمن از مشکلات اجرای پلاگین نوشته شده توسط برنامه نویس در IE و نسخه های مختلف آن هم چشم پوشی می کنیم. یکی از مزیت های این روش نسبت به بقیه این هست که دانلود و نصب پلاگین مورد نیاز به صورت خودکار و توسط مرورگر انجام می شود.

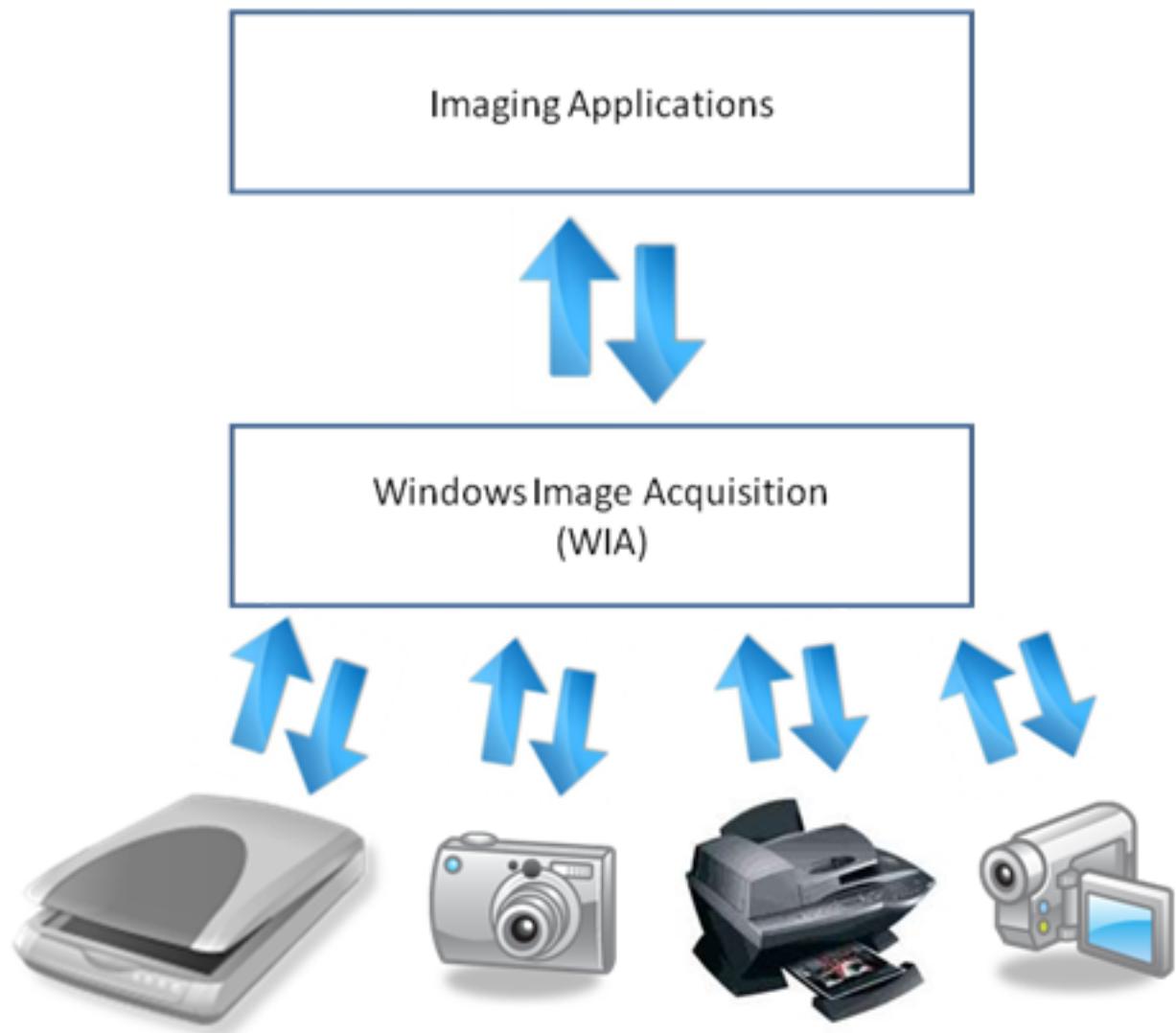
2- استفاده از یک کامپونت جانبی : مثل کامپونت های [LEADTOOLS](#) که ابزارهای متعدد و SDK های بسیار قدرتمندی را برای اینکار و کارهای دیگر، مانند کار با اسکن تصاویر مغزی، پردازش تصویر، بارکد خوانها، مدیریت استناد و ... فراهم کرده است. قیمت این ابزار بسیار زیاد است و در برخی موارد امکانات فراهم آورده بسیار بیشتر است از خواسته‌ی ما. این ابزار، هم در & HTML و هم در DotNet و Javascript قابل استفاده است و مستندات نسبتاً خوبی هم در این زمینه ارائه کرده است. همچنین کامپونت [Dynamsoft](#) که باز هم غیر رایگان هست و قیمت بالایی نیز دارد.

اگر روال کار کامپونت های بالا را مورد بررسی قرار دهید (از طریق اجرای [Demo](#) ها، [اینجا](#) و [اینجا](#)) متوجه خواهید شد که هر دو نیاز به نصب یک سرویس یا App سمت کلاینت جهت اجرای دستورات خود دارند. پس می شود اینطور نتیجه گرفت که انجام اینکار بدون اینکه چیزی سمت کاربر نصب شود، ممکن نیست و در هر دو، لینک نصب فایل exe سرویس برای دانلود قرار داده شده است. بر این اساس به راه حل سومی خواهیم رسید که خودمان این سرویس را جهت تعامل با اسکنر سمت کاربر طراحی و پیاده سازی نماییم.

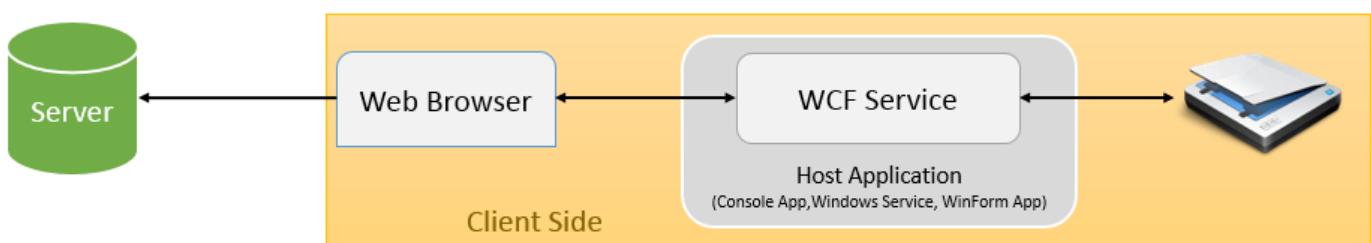
اما چطور ممکن است که با اجرای یک فایل exe سمت کاربر (با این فرض که کاربر در یک دامین مطمئن قرار دارد و می شود درخواست نصب سرویس را نمود) این امکان را برای کاربر فراهم نمود که با یک کلیک در مرورگر، اسکنر به صورت خودکار اسکن را آغاز کرده و سپس تصویر حاصل را به یکی از کنترلرهای ما در سمت سرور ارسال نماید؟

برای اینکار ما با دو صورت مساله مواجه هستیم؛ اول اینکه چطور تصویر را سمت کاربر اسکن کنیم و دوم اینکه چطور این تصویر را به سرور ارسال نماییم!

برای مساله اول از کتابخانه [Windows Image Acquisition \(WIA\)](#) استفاده خواهیم نمود که این کتابخانه به ما این امکان را میدهد تا با سخت افزارهایی که از [TWAIN](#) پشتیبانی می کنند، بتوانیم ارتباط برقرار نماییم.

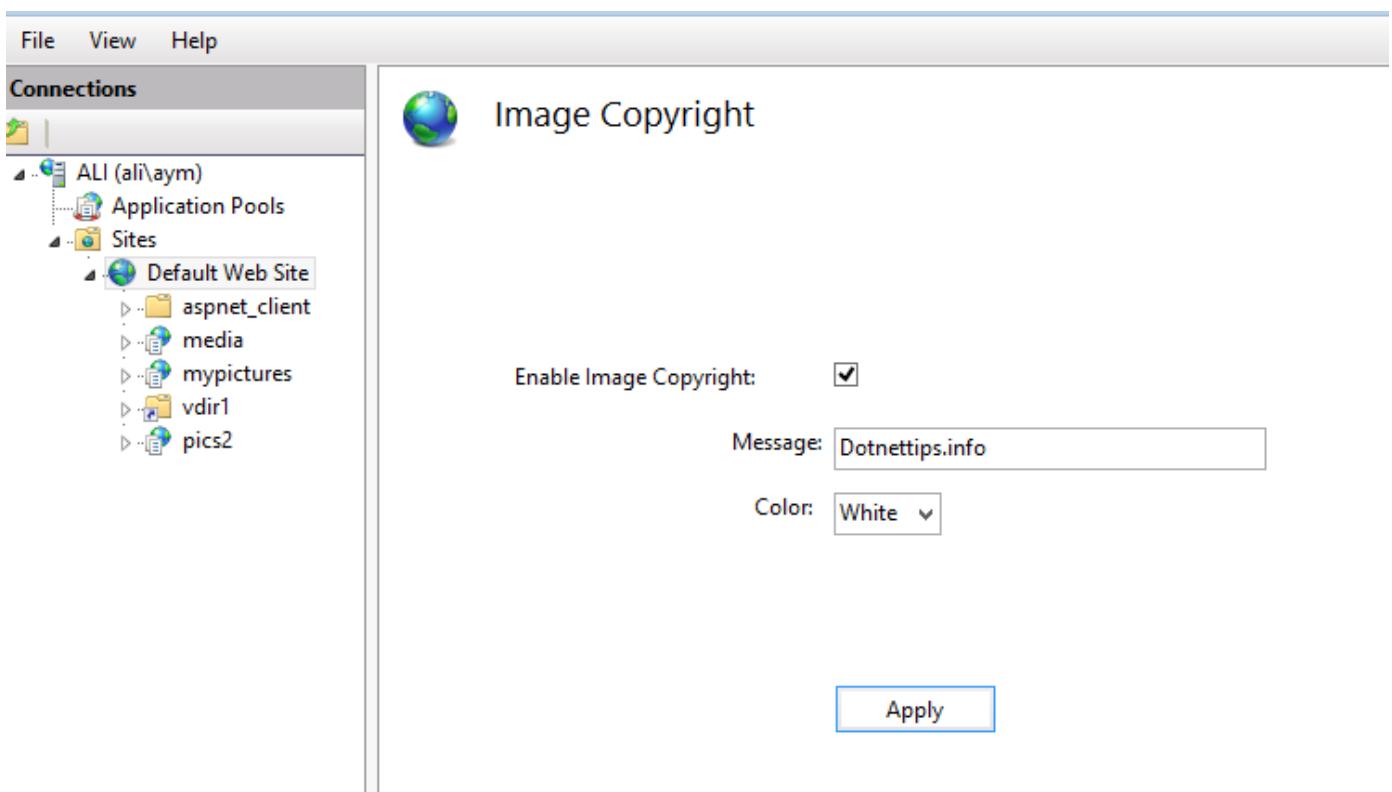


برای مساله‌ی دوم هم نیاز به پیاده سازی یک WCF Service و اجرای آن (هاست کردن) در سمت کلاینت داریم. در واقع با صد از زدن متدهای این سرویس، از کتابخانه‌ی بالا استفاده کرده و اسکن را انجام می‌دهیم.



ادامه دارد...

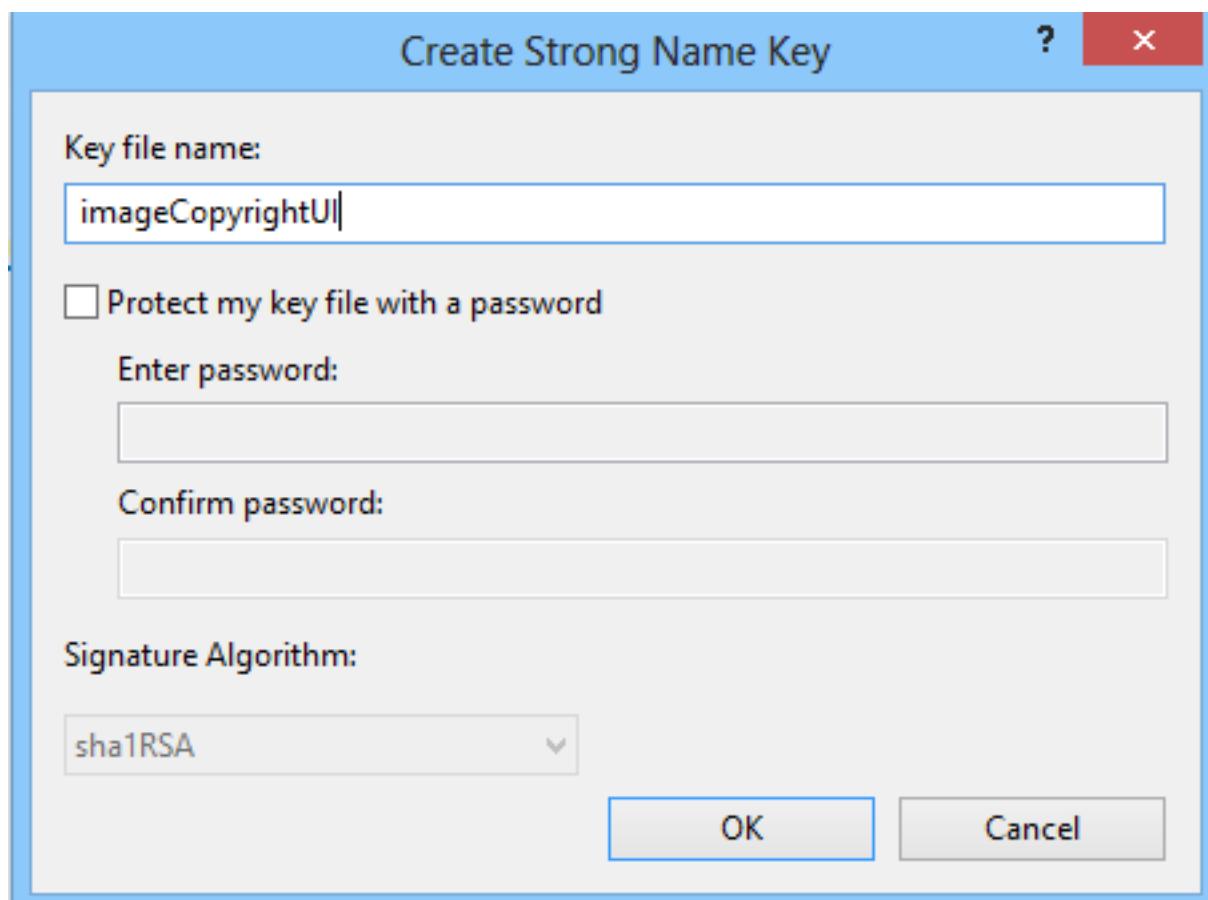
در قسمت [قبلی](#) ما یک هندر ایجاد کردیم و درخواست‌هایی را که برای فایل jpg و به صورت GET ارسال می‌شد، هندر می‌کردیم و تگی را در گوششی تصویر درج و آن را در خروجی نمایش میدادیم. در این مقاله قصد داریم که کمی هندر مورد نظر را توسعه دهیم و برای آن یک UI یا یک رابط کاربری ایجاد نماییم. برای توسعه دادن مازولها و هندرها ما یک dll نوشته و باید آن را در GAC که مخفف عبارت [Global Assembly Cache](#) ریجستر کنیم.



جهت اینکار یک پروژه از نوع `class library` ایجاد کنید. فایل `class1.cs` را که به طور پیش فرض ایجاد می‌شود، حذف کنید و رفرنس‌های `Microsoft.Web.Administration.dll` و `Microsoft.Web.Management.dll` را از مسیر زیر اضافه کنید:

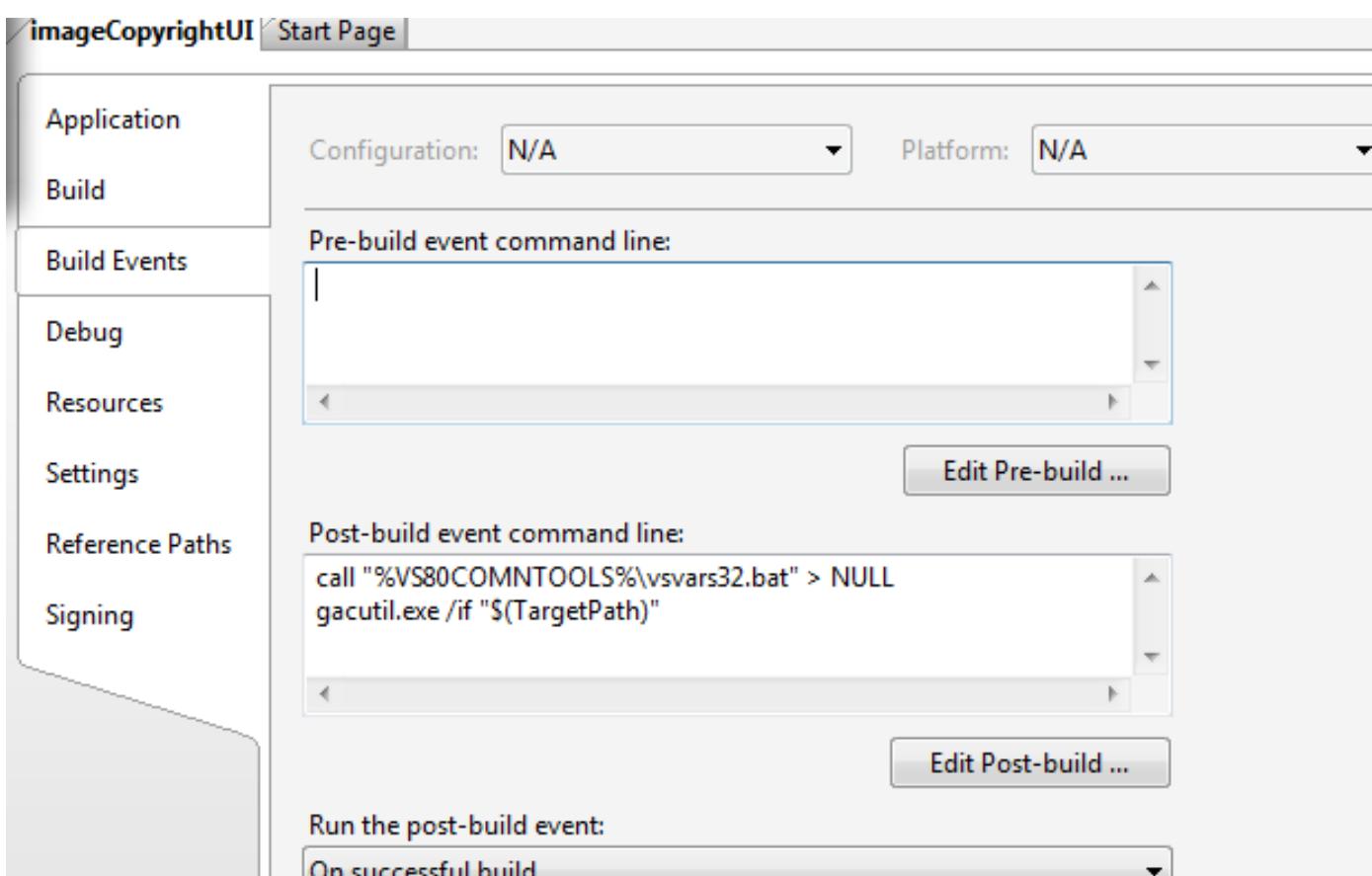
`\Windows\system32\inetsrv`

اولین رفرنس شامل کلاس‌هایی است که جهت ساخت مازولها برای کنسول IIS مورد نیاز است و دومی هم برای خواندن پیکربندی‌های نوشته شده مورد استفاده قرار می‌گیرد. برای طراحی UI بر پایه `winform` باید رفرنس‌های `System.dll` و `System.Windows.Forms.dll` و `System.Web.dll` را از سری اسمبلی‌های دات نت نیز اضافه کنیم و در مرحله‌ی بعدی جهت ایجاد امضاء یا `strong name` (`^` و `^`) به خاطر ثبت در GAC پروژه را انتخاب و وارد `Properties` پروژه شوید. در تب `sign the assembly` گزینه `new` گزینه `signing` را انتخاب نمایید و نام `imageCopyrightUI` را به آن نسبت داده و گزینه `تیک` زده و در لیست باز شده گزینه `new` را انتخاب نمایید `snk` و نام `imageCopyrightUI` را به آن نسبت داده و گزینه `تیک` زده و در لیست باز شده گزینه `new` را انتخاب نمایید `snk`. این باید یک فایل `key` باشد تا بعداً با استفاده از این کلید `dll` ایجاد شده را در GAC ریجستر کنیم.



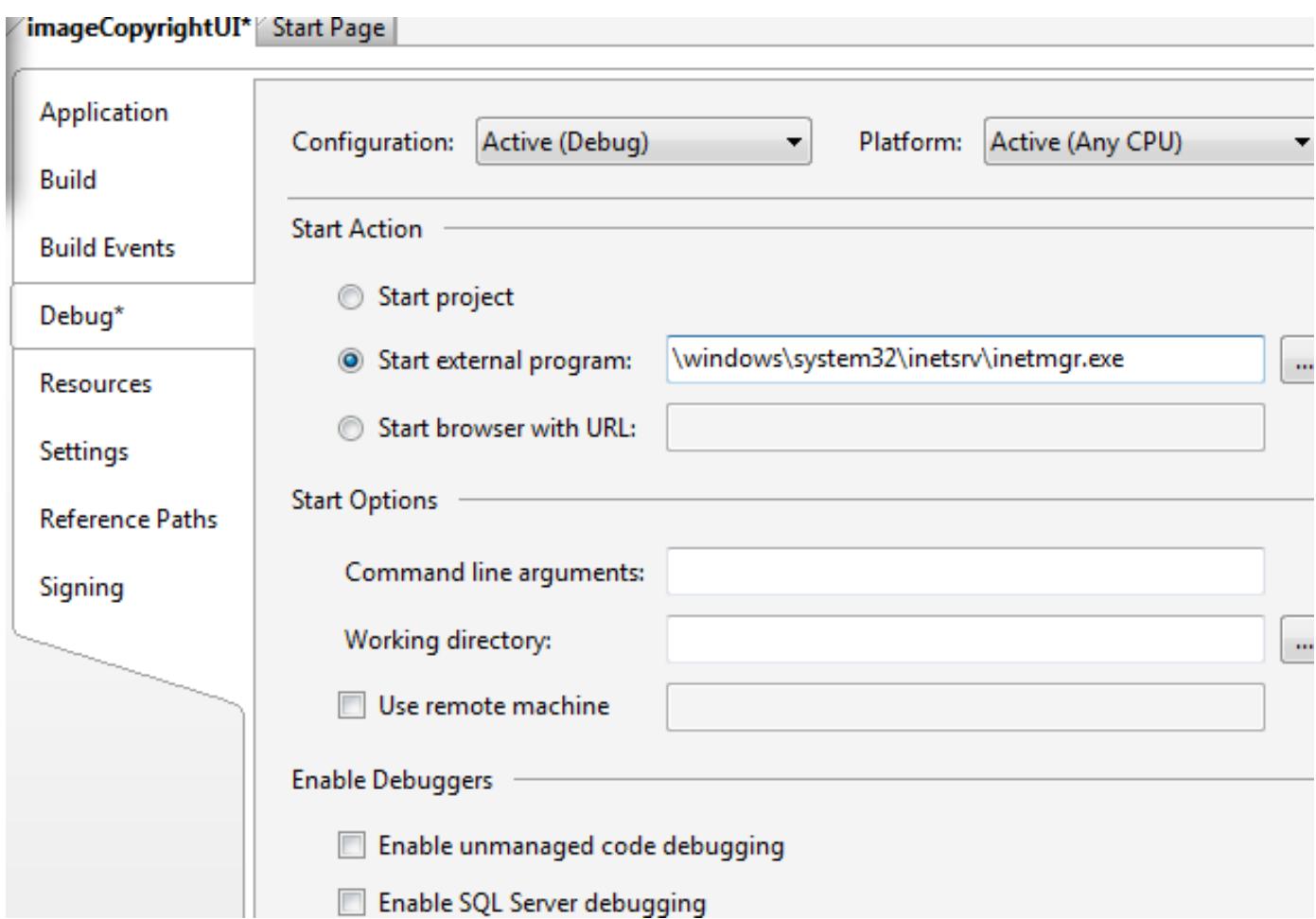
در مرحله بعدی در تب [Build Events](#) کد زیر را در بخش Post-build event command line اضافه کنید. این کد باعث می‌شود بعد از هر بار کامپایل پروژه، به طور خودکار در GAC ثبت شود:

```
call "%VS80COMNTOOLS%\vsvars32.bat" > NULL  
gacutil.exe /if "$(TargetPath)"
```



نکته: در صورتی که از vs2005 استفاده می‌کنید در تب Debug در قسمت Start External Program مسیر زیر را قرار بدهید.
اینکار برای تست و دیباگینگ پروژه به شما کمک خواهد کرد. این تنظیم شامل نسخه‌های اکسپرس نمی‌شود.

\windows\system32\inetsrv\inetmgr.exe



بعد از پایان اینکار پروژه را Rebuild کنید. با اینکار dll در GAC ثبت می‌شود. استفاده از سوییچ‌های if به طور همزمان در درستور gacutil به معنی این هست که اگر اولین بار است نصب می‌شود، پس با سوییچ i نصب کن. ولی اگر قبل نصب شده است نسخه جدید را به هر صورتی هست جایگزین قبلى کن یا همان reinstall کن.

ساخت یک Module Provider

رابطه‌های کاربری IIS همانند هسته و کل سیستم، ماژول‌وار و قابل خصوصی سازی است. رابط کاربری، مجموعه‌ای از ماژول‌هایی است که میتوان آن‌ها را حذف یا جایگزین کرد. تگ ورودی یا معرفی برای هر UI یک module provider است. خیلی خودمانی، تگ ماژول پروایدر به معرفی یک UI در IIS می‌پردازد. لیستی از module provider‌ها را می‌توان در فایل زیر در تگ بخش <modules> پیدا کرد.

```
%windir%\system32\inetsrv\Administration.config
```

در اولین گام یک کلاس را به اسم imageCopyrightUIModuleProvider.cs ایجاد کرده و سپس آنرا به کد زیر، تغییر می‌دهیم. کد زیر با استفاده از ModuleDefinition داده و کلاس Module Provider را که بعداً تعریف می‌کنیم، به عنوان مدخل entry رابط کاربری معرفی کرده:

```
using System;
using System.Security;
using Microsoft.Web.Management.Server;

namespace IIS7Demos
{
    class imageCopyrightUIProvider : ModuleProvider
    {
        public override Type ServiceType
```

```

    {
        get { return null; }
    }

    public override ModuleDefinition GetModuleDefinition(IManagementContext context)
    {
        return new ModuleDefinition(Name, typeof(imageCopyrightUI).AssemblyQualifiedName);
    }

    public override bool SupportsScope(ManagementScope scope)
    {
        return true;
    }
}
}

```

با ارث بری از کلاس `module provider`, سه متد بازنویسی می‌شوند که یکی از آن‌ها `SupportsScope` هست که میدان عمل پروایدر را مشخص می‌کند، مانند اینکه این پرواید در چه میدانی باید کار کند که می‌تواند سه گزینه‌ی `server,site,application` باشد. در کد زیر مثلاً میدان عمل `application` انتخاب شده است ولی در کد بالا با برگشت مستقیم `true`, همه‌ی میدان را جهت پشتیبانی از این پروایدر اعلام کردیم.

```

public override bool SupportsScope(ManagementScope scope)
{
    return (scope == ManagementScope.Application) ;
}

```

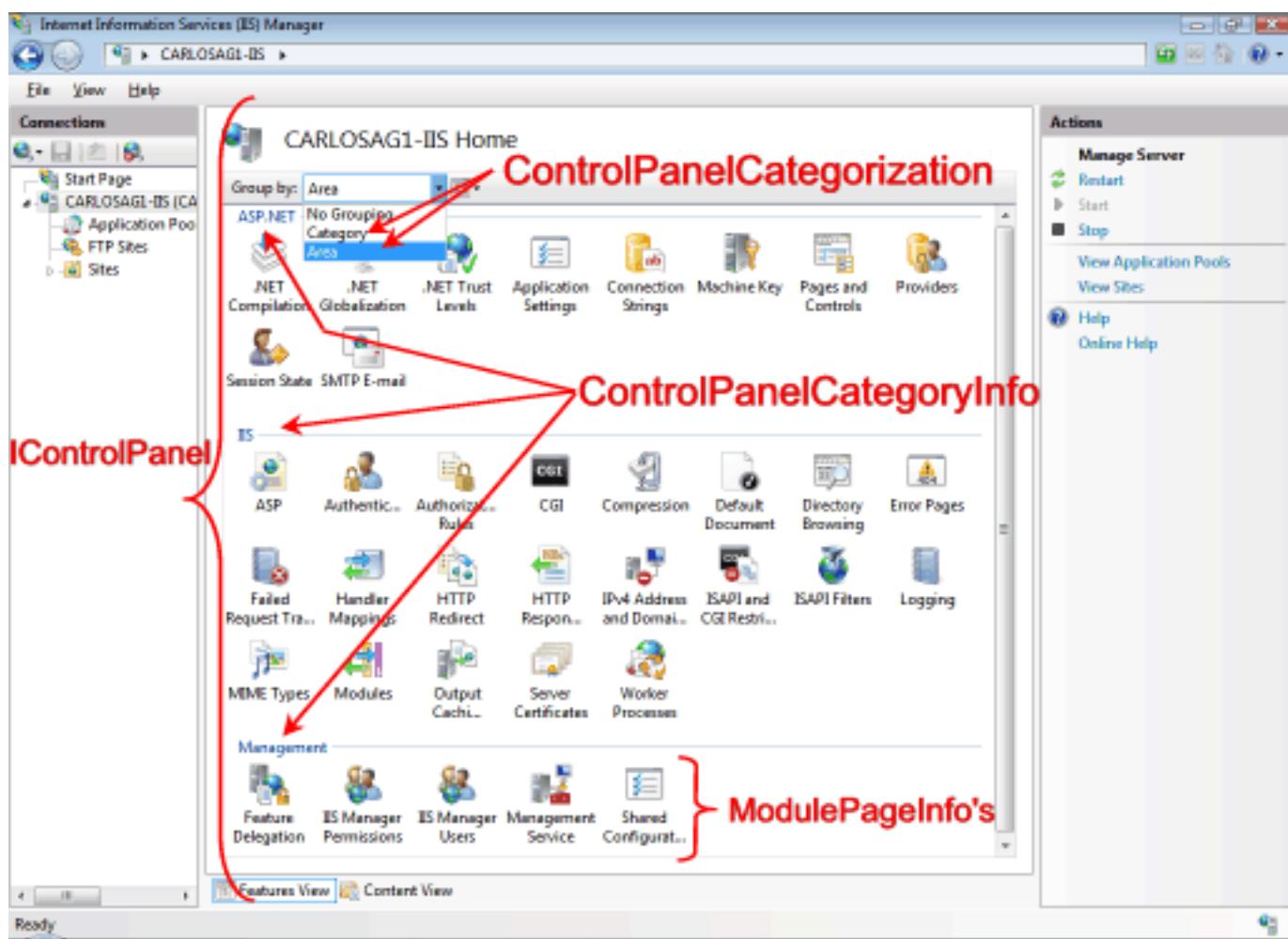
حالا که پروایدر (معرف رابط کاربری به IIS) تامین شده، نیاز است قلب کار یعنی ماژول معرفی گردد. اصلی‌ترین متدهی که باید از اینترفیس ماژول پیاده سازی شود متد `initialize` است. این متد جایی است که تمام عملیات در آن رخ می‌دهد. در کلاس زیر `imageCopyrightUI` ما به معرفی مدخل `entry` رابط کاربری می‌پردازیم. در سازنده‌های این متد، پارامترهای نام، صفحه رابط کاربری و توضیحی در مورد آن است. تصویر کوچک و بزرگ جهت آیکن سازی (در صورت عدم تعریف آیکن، چرخ دنده نمایش داده می‌شود) و توصیف‌های بلندتر را نیز شامل می‌شود.

```

internal class imageCopyrightUI : Module
{
    protected override void Initialize(IServiceProvider serviceProvider, ModuleInfo moduleInfo)
    {
        base.Initialize(serviceProvider, moduleInfo);
        IControlPanel controlPanel = (IControlPanel)GetService(typeof(IControlPanel));
        ModulePageInfo modulePageInfo = new ModulePageInfo(this, typeof(imageCopyrightUIPage),
        "Image Copyright", "Image Copyright", Resource1.Visual_Studio_2012, Resource1.Visual_Studio_2012);
        controlPanel.RegisterPage(modulePageInfo);
    }
}

```

شیء `ControlPanel` مکانی است که قرار است آیکن ماژول نمایش داده شود. شکل زیر به خوبی نام همه قسمت‌ها را بر اساس نام کلاس و اینترفیس آن‌ها دسته بندی کرده است:



پس با تعریف این کلاس جدید ما روی صفحه‌ی کنترل پنل IIS، یک آیکن ساخته و صفحه‌ی رابط کاربری را به نام `imageCopyrightUIPage` در آن ریجستر می‌کنیم. این کلاس را پایینتر شرح داده‌ایم. ولی قبل از آن اجازه بدهید تا انواع کلاس‌هایی را که برای ساخت صفحه کاربرد دارند، بررسی نماییم. در این مثال ما با استفاده از پایه‌ای‌ترین کلاس، ساده‌ترین نوع صفحه ممکن را خواهیم ساخت. ۴ کلاس برای ساخت یک صفحه وجود دارند که بسته به سناریوی کاری، شما یکی را انتخاب می‌کنید.

<p>شامل اساسی‌ترین متدها و سورس‌ها شده و هیچگونه رابط کاری ویژه‌ای را در اختیار شما قرار نمی‌دهد. تنها یک صفحه‌ی خام به شما می‌دهد که می‌توانید از آن استفاده کرده یا حتی با ارت بری از آن، کلاس‌های جدیدتری را برای ساخت صفحات مختلف و ویژه‌تر بسازید. در حال حاضر که هیچ کدام از ویژگی‌های IIS فعلی از این کلاس برای ساخت رابط کاربری استفاده نکرده‌اند.</p>	<p>ModulePage</p>
<p>یک صفحه شبیه به دیالوگ را ایجاد می‌کند و شامل دکمه‌های <code>Cancel</code> و <code>Apply</code> می‌شود به همراه یک سری متدهای اضافی‌تر که اجازه‌ی <code>override</code> کردن آنها را دارید. همچنین یک سری از کارهایی چون <code>refresh</code> و از این دست عملیات خودکار را نیز انجام می‌دهد. از نمونه رابطه‌ایی که از این صفحات استفاده می‌کنند می‌توان <code>management service</code> و <code>machine key</code> را اسم برد.</p>	<p>ModuleDialogPage</p>

<p>شامل اساسی‌ترین متد ها و سورس‌ها شده و هیچگونه رابط کاری ویژه‌ای را در اختیار شما قرار نمی‌دهد. تنها یک صفحه‌ی خام به شما می‌دهد که می‌توانید از آن استفاده کرده یا حتی با ارث بری از آن، کلاس‌های جدیدتری را برای ساخت صفحات مختلف و ویژه‌تر بسازید. در حال حاضر که هیچ کدام از ویژگی‌های IIS فعلی از این کلاس برای ساخت رابط کاربری استفاده نکرده‌اند.</p>	<p>ModulePage</p>
<p>این صفحه یک رابط کاربری را شبیه پنجره property که در ویژوال استادیو وجود دارد، در دسترس شما قرار میدهد. تمام عناصر آن در یک حالت گردید grid لیست می‌شوند. از نمونه‌های موجود می‌توان به CGI, ASP.NET Compilation اشاره کرد.</p>	<p>ModulePropertiesPage</p>
<p>این کلاس برای موقعي کاربرد دارد که شما قرار است لیستی از آیتم‌ها را نشان دهید. در این صفحه شما یک ListView دارید که می‌توانید عملیات جست و جو، گروه بندی و نحوه‌ی نمایش لیست را روی آن اعمال کنید.</p>	<p>ModuleListPage</p>

در این مثال ما از اولین کلاس نامبرده که پایه‌ی همه کلاس هاست استفاده می‌کنیم. کد زیر را در کلاسی به اسم **imageCopyrightUIPage**:

```
public sealed class imageCopyrightUIPage : ModulePage
{
    public string message;
    public bool featureenabled;
    public string color;

    ComboBox _colCombo = new ComboBox();
    TextBox _msgTB = new TextBox();
    CheckBox _enabledCB = new CheckBox();

    public imageCopyrightUIPage()
    {
        this.Initialize();
    }

    void Initialize()
    {
        Label crlabel = new Label();
        crlabel.Left = 50;
        crlabel.Top = 100;
        crlabel.AutoSize = true;
        crlabel.Text = "Enable Image Copyright:";

        _enabledCB.Text = "";
        _enabledCB.Left = 200;
        _enabledCB.Top = 100;
        _enabledCB.AutoSize = true;

        Label msglabel = new Label();
        msglabel.Left = 150;
        msglabel.Top = 130;
        msglabel.AutoSize = true;
        msglabel.Text = "Message:";

        _msgTB.Left = 200;
        _msgTB.Top = 130;
        _msgTB.Width = 200;
        _msgTB.Height = 50;

        Label collabel = new Label();
        collabel.Left = 160;
        collabel.Top = 160;
        collabel.AutoSize = true;
        collabel.Text = "Color:";

        _colCombo.Left = 200;
```

```

        _colCombo.Top = 160;
        _colCombo.Width = 50;
        _colCombo.Height = 90;
        _colCombo.Items.Add((object)"Yellow");
        _colCombo.Items.Add((object)"Blue");
        _colCombo.Items.Add((object)"Red");
        _colCombo.Items.Add((object)"White");

        Button apply = new Button();
        apply.Text = "Apply";
        apply.Click += new EventHandler(this.applyClick);
        apply.Left = 200;
        apply.AutoSize = true;
        apply.Top = 250;

        Controls.Add(crlabel);
        Controls.Add(_enabledCB);
        Controls.Add(collabel);
        Controls.Add(_colCombo);
        Controls.Add(msglabel);
        Controls.Add(_msgTB);
        Controls.Add(apply);
    }

    public void ReadConfig()
    {
        try
        {
            ServerManager mgr;
            ConfigurationSection section;
            mgr = new ServerManager();
            Configuration config =
                mgr.GetWebConfiguration(
                    Connection.ConfigurationPath.SiteName,
                    Connection.ConfigurationPath.ApplicationPath +
                    Connection.ConfigurationPath.FolderPath);

            section = config.GetSection("system.webServer/imageCopyright");
            color = (string)section.GetAttribute("color").Value;
            message = (string)section.GetAttribute("message").Value;
            featureenabled = (bool)section.GetAttribute("enabled").Value;
        }
        catch
        { }
    }

    void UpdateUI()
    {
        enabledCB.Checked = featureenabled;
        int n = _colCombo.FindString(color, 0);
        _colCombo.SelectedIndex = n;
        _msgTB.Text = message;
    }

    protected override void OnActivated(bool initialActivation)
    {
        base.OnActivated(initialActivation);
        if (initialActivation)
        {
            ReadConfig();
            UpdateUI();
        }
    }

    private void applyClick(Object sender, EventArgs e)
    {
        try
        {
            UpdateVariables();
            ServerManager mgr;
            ConfigurationSection section;
            mgr = new ServerManager();
            Configuration config =
                mgr.GetWebConfiguration(
(

```

```

        Connection.ConfigurationPath.SiteName,
        Connection.ConfigurationPath.ApplicationPath +
        Connection.ConfigurationPath.FolderPath
    );

    section = config.GetSection("system.webServer/imageCopyright");
    section.GetAttribute("color").Value = (object)color;
    section.GetAttribute("message").Value = (object)message;
    section.GetAttribute("enabled").Value = (object)featureenabled;

    mgr.CommitChanges();
}

catch
{ }

}

public void UpdateVariables()
{
    featureenabled = _enabledCB.Checked;
    color = _colCombo.Text;
    message = _msgTB.Text;
}
}

```

اولین چیزی که در کلاس بالا صدا زده می‌شود، سازنده‌ی کلاس هست که ما در آن یک تابع تعريف کردیم به اسم **initialize** که به آماده سازی اینترفیس یا رابط کاربری می‌پردازد و کنترل‌ها را روی صفحه می‌چیند. این سه کنترل، یکی *Combox* برای تعیین رنگ، یکی *Checkbox* برای فعال بودن مژول و دیگری هم یک *textbox* جهت نوشتمن متن است. ماقبی هم که سه *Label* برای نامگذاری اشیاست. بعد از اینکه کنترل‌ها روی صفحه درج شدند، لازم است که تنظیمات پیش فرض یا قبلی روی کنترل‌ها نمایش یابند که اینکار را به وسیله تابع **readConfig** انجام می‌دهیم و تنظیمات خوانده شده را در متغیرهای عمومی قرار داده و با استفاده از تابع **UpdateUI** این اطلاعات را روی کنترل‌هاست می‌کنیم و به این ترتیب *UI* به روز می‌شود. این دو تابع را به ترتیب پشت سر هم در یک متد به اسم **override** که **OnActivated** در واقع این متد یک جورایی همانند رویداد *Load* می‌باشد؛ اگر برگرداند اولین فعال سازی رابط کاربری بعد از باز شدن IIS است و در غیر این صورت *false* بر می‌گرداند.

در صورتی که کاربر مقادیر را تغییر دهد و روی گزینه *applyClick* کلیک کند تابع *applyClick* اجرا شده و ابتدا به تابع **UpdateVariables** ارجاع داده می‌شود که در آن مقادیر خوانده شده و در متغیرهای *Global* قرار می‌گیرند و سپس با استفاده از دو شیء از نوع *ConfigSection* و *serverManger* جایگذاری یا ذخیره می‌شوند.

استفاده از دو کلاس *Configsection* و *Servermanager* در دو قسمت خواندن و نوشتمن مقادیر به کار رفته‌اند. کلاس *servermanager* به ما اجازه دسترسی به تنظیمات IIS و قابلیت‌های آن را میدهد. در تابع *ReadConfig* مسیر وب سایتی را که در لیست IIS انتخاب شده است، دریافت کرده و به وب کانفیگ آن وب سایت رجوع نموده و تگ *imageCopyright* آن را که در تگ *system.webserver* قرار گرفته است، می‌خواند (در صورتی که این تگ در آن وب کانفیگ موجود نباشد، خواندن و سپس ذخیره مجدد آن روی تگ داخل فایل *applicationHost.config* اتفاق می‌فتد که نتیجتاً برای همه‌ی وب سایت هایی که این تگ را ندارند یا مقدارهای پیش فرض آن را تغییر نداده‌اند رخ میدهد) عملیات نوشتمن هم مشابه خواندن است. تنها باید خط زیر را در آخر برای اعمال تغییرات نوشت؛ مثل EF با گزینه *:Context.SaveChanges*:

```
mgr.CommitChanges();
```

وقت آن است که رابط کاربری را به IIS اضافه کنیم: پروژه را *Rebuild* کنید. بعد از آن با خطوطی که قبلاً در نوشتم باید *d11* ما در GAC ریجستر شود. برای همین آدرس زیر را در cmd تایپ کنید:

```
%vs110comntools%\vsvars32.bat
```

عبارت اول که مسیر ویژوال استودیو شماست و عدد 110 یعنی نسخه‌ی 11. هر نسخه‌ای را که استفاده می‌کنید، یک صفر جلویش بگذارد و جایگزین عدد بالا کنید. مثلاً نسخه 8 می‌شود و فایل بج‌بالا هم دستورات visual studio را برای شما آزاد می‌کند.

سپس دستور زیر را وارد کنید:

```
GACUTIL /l ClassLibrary1
```

کلمه classLibrary1 نام پروژه‌ی ما بود که در GAC رегистر شده است. با سوییچ ۱ تمامی اطلاعات اسمبلی‌هایی که در GAC رегистر شده‌اند، نمایش می‌یابند. ولی اگر اسم آن اسمبلی را جلویش بنویسید، فقط اطلاعات آن اسمبلی نمایش می‌بادد. با اجرای خط فوق میتوانیم کلید عمومی public key اسمبلی خود را بدانیم که در شکل زیر مشخص شده است:

```
C:\Windows\system32\cmd.exe
E:\Program Files (x86)\Microsoft Visual Studio 11.0\Common7\Tools>GACUTIL /l ClassLibrary1
Microsoft (R) .NET Global Assembly Cache Utility. Version 4.0.30319.17929
Copyright (c) Microsoft Corporation. All rights reserved.

The Global Assembly Cache contains the following assemblies:
  ClassLibrary1, Version=1.0.0.0, Culture=neutral, PublicKeyToken=d0b3b3b2aa8ea14b, processorArchitecture=MSIL

Number of items = 1

E:\Program Files (x86)\Microsoft Visual Studio 11.0\Common7\Tools>
```

پس اگر کلید را دریافت کرده‌اید، خط زیر را به فایل administration.config در تگ <ModuleProviders> اضافه کنید:

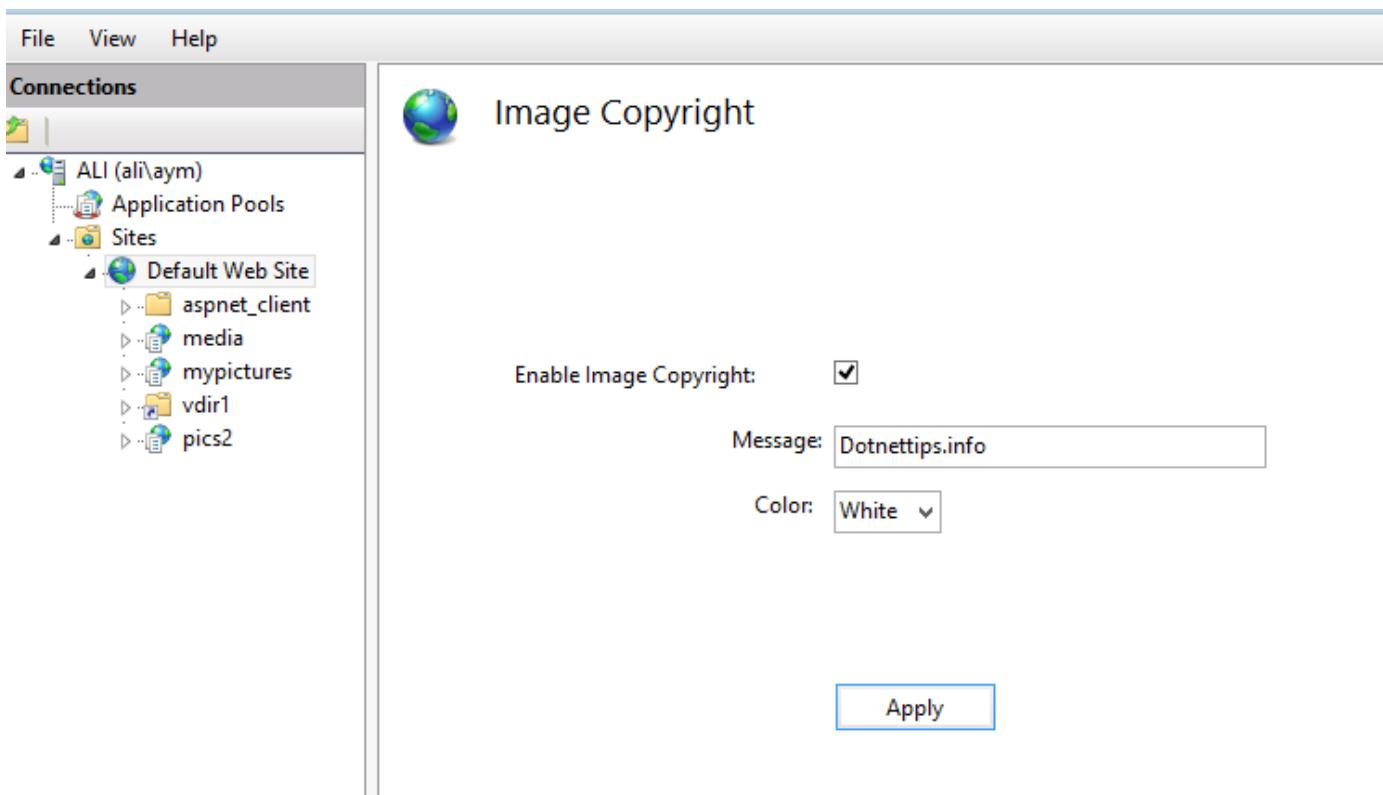
```
<add name="imageCopyrightUI" type="ClassLibrary1.imageCopyrightUIProvider, ClassLibrary1,
Version=1.0.0.0, Culture=neutral, PublicKeyToken=d0b3b3b2aa8ea14b"/>
```

عبارت UI imageCopyrightUIProvider به کلاس ClassLibrary1.imageCopyrightUIProvider معرفی می‌شود. مابقی عبارت هم کاملاً مشخص است و در لینک‌های بالا در مورد Strong name توضیح داده شده‌اند.

فایل administration.config در مسیر زیر قرار دارد:

```
%windir%\system32\inetsrv\config\administration.config
```

حالا تنها کاری که نیاز است، باز کردن IIS است. به بخش وب سایت‌ها رفته و اپلیکیشنی که قبلًا با نام mypictures را ایجاد کردیم، انتخاب کنید. در سمت راست، آخر لیست، بخش others باید مازول ما دیده شود. بازش کنید و تنظیمات آن را تغییر دهید و حالا یک تصویر را از اپلیکیشن mypictures روی مرورگر درخواست کنید تا تعییرات را روی تگ مشاهده کنید:

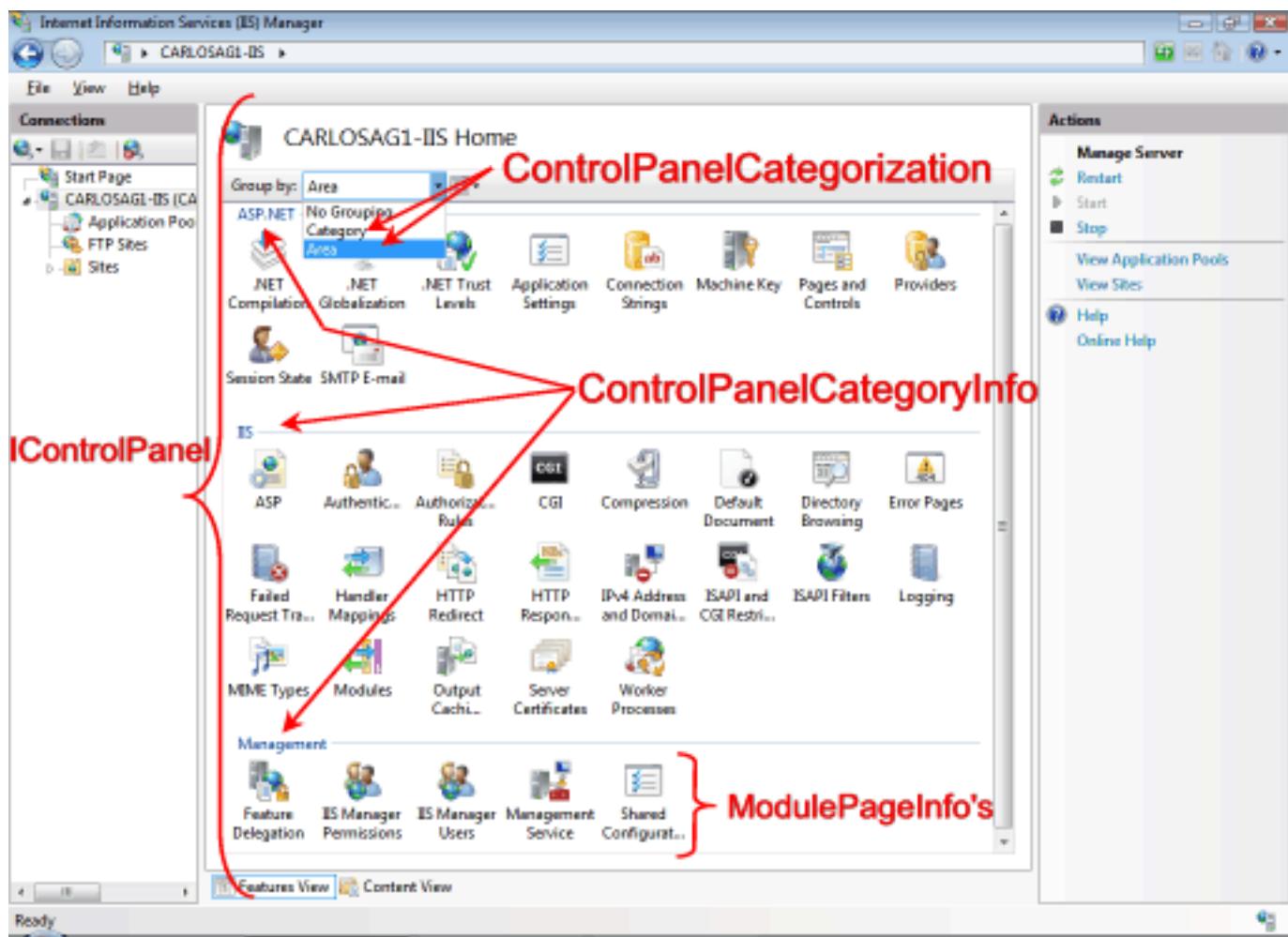


حالا دیگر باید مازول نویسی برای IIS را فراگرفته باشیم. این مازول‌ها می‌توانند از یک مورد ساده تا یک کلاس مهم و امنیتی باشند که روی سرور شما برای همه یا بعضی از وب سایتها در حال اجرا هستند و در صورت لزوم و اجازه شما، برنامه نویس‌ها می‌توانند مثل همه‌ی تگ‌های موجود در وب کانفیگ سایتی را که مینویسند، تگ مازول شما و تنظیمات آن را با استفاده از attribute یا خصوصیت‌های تعریف شده، بر اساس سلائق و نیازهایشان تغییر دهند و روی سرور شما آپلود کنند. الان شما یک سرور خصوصی سازی شده دارید. از آنجا که این مقاله طولانی شده است، باقی موارد ویرایشی روی این UI را در مقاله بعدی بررسی خواهیم کرد.

در دو مقاله پیشین ([^](#) و [_](#)) در مورد اینکه چگونه یک ماژول جدید به IIS اضافه کنیم صحبت کردیم و الان قصد داریم که در این بخش این مبحث را بیندیم. آخرین بار توانستیم که یک UI را به IIS نسبت داده و از آن استفاده کنیم و الان قصد داریم آن را تکمیل‌تر کرده و کمی آن را شکلی‌تر کنیم. اولین نکته‌ای که توجه ما را جلب می‌کند این است که ماژول ما یک آیکن پیش فرض (چرخ دنده) دارد که بهتر است به آیکن دلخواه ما تغییر کند. به این منظور در پروژه، یک فایل `Resource` ایجاد کنید و یک تصویر را به این فایل `Resource` اضافه کنید و در کلاس `imageCopyrightUI` کد را به صورت زیر تغییر دهید:

```
internal class imageCopyrightUI : Module
{
    protected override void Initialize(IServiceProvider serviceProvider, ModuleInfo moduleInfo)
    {
        base.Initialize(serviceProvider, moduleInfo);
        IControlPanel controlPanel = (IControlPanel)GetService(typeof(IControlPanel));
        ModulePageInfo modulePageInfo = new ModulePageInfo(this, typeof(imageCopyrightUIPage),
        "Image Copyright", "Image Copyright", Resource1.Visual_Studio_2012, Resource1.Visual_Studio_2012);
        controlPanel.RegisterPage(modulePageInfo);
    }
}
```

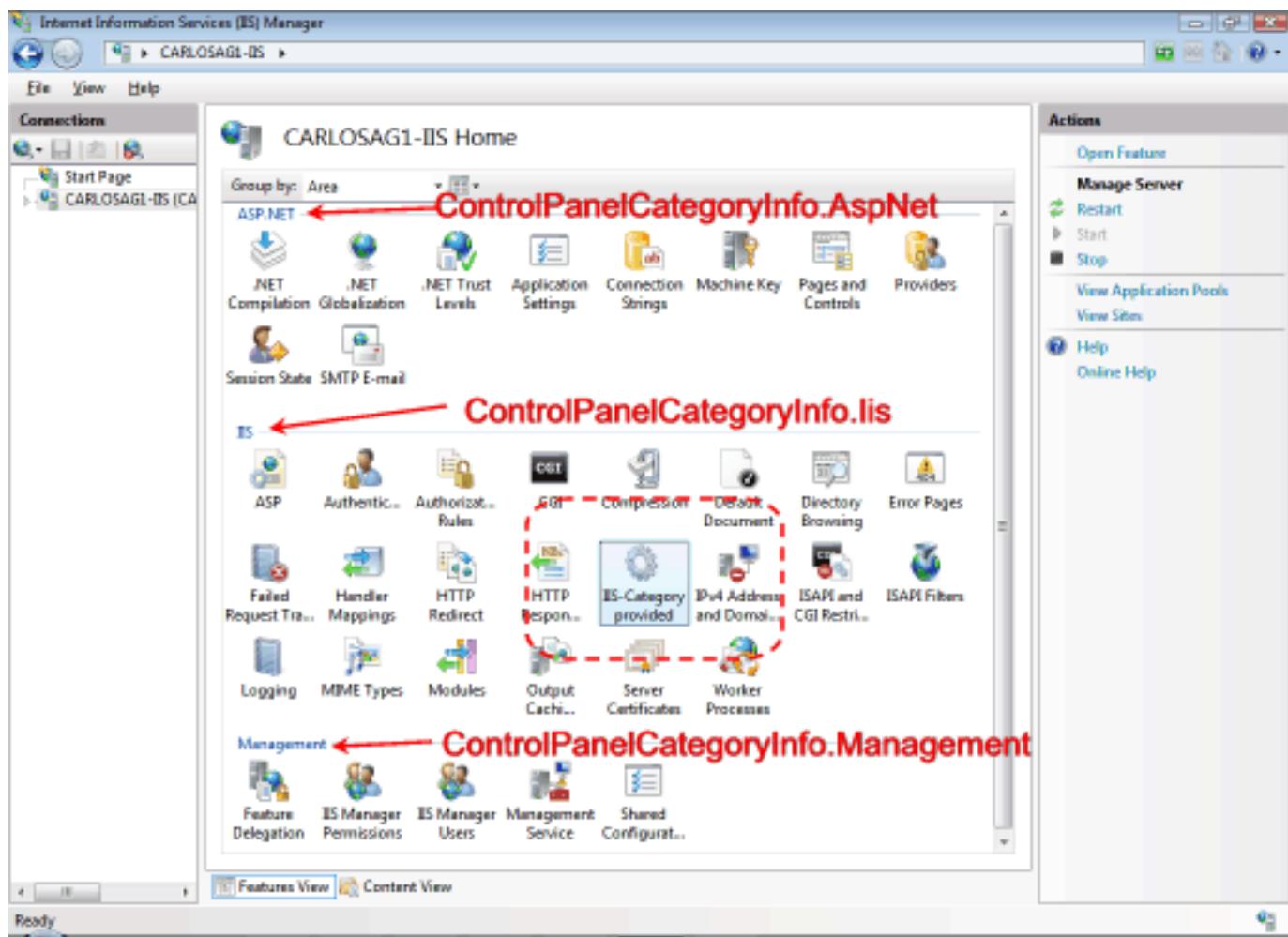
حال پروژه را `Rebuild` کنید و IIS را مجدداً باز کنید تا ببینید که کامپوننت جدید شما آیکن جدید را دارد. دومین نکته‌ای که به چشم می‌آید این است که اگر دقت کنید سایر ماژول‌ها یا کامپوننت‌ها هر کدام در گروهی جداگانه جای گرفته‌اند و افزونه‌ی شما در یک گروه به اسم `others`، که زیاد جالب نیست و شاید دوست داشته باشید که ماژول شما هم در یکی از همین دسته‌ها جای بگیرد یا حتی اینکه خودتان یک گروه جدید بسازید. برای اینکه خوب قسمت‌ها و نام کلاس‌های آن‌ها را یاد بگیرید به شکلی که در قسمت قبلی هم گذاشته بودیم خوب دقت کنید.



نکته ای که باید توجه داشته باشید این است که گروه بندی در IIS به سه شیوه صورت می‌گیرید : یکی به شیوه ناحیه **Area** مثل **Application Development** , **ServerFeatures Category** و **IIS management** و **ASP.Net** دسته بندی **Group By** در بالای پنجره IIS یکی از این سه حالت را انتخاب نمایید برای اینکه مازول در یک ناحیه یا دسته‌ای مشخص قرار بگیرد کد زیر را بنویسید:

```
internal class imageCopyrightUI : Module
{
    protected override void Initialize(IServiceProvider serviceProvider, ModuleInfo moduleInfo)
    {
        base.Initialize(serviceProvider, moduleInfo);
        IControlPanel controlPanel = (IControlPanel)GetService(typeof(IControlPanel));
        ModulePageInfo modulePageInfo = new ModulePageInfo(this, typeof(imageCopyrightUIPage),
        "Image Copyright", "Image Copyright", Resource1.Visual_Studio_2012, Resource1.Visual_Studio_2012);
        controlPanel.RegisterPage(ControlPanelCategoryInfo.AspNet, modulePageInfo);
    }
}
```

اگر خوب دقت کنید می‌بینید که تنها تغییر کدها در متد registerpage بوده است که با استفاده از کلاس **ControlPanelCategoryInfo** ناحیه مورد نظر را مشخص کردیم و اگر حالا پروژه را Rebuild کنید می‌بینید که در ناحیه **ASP.Net** قرار گرفته است. گروه‌های مختلف دیگری چون **Application Development**, **Server**, **Performance** و ... هم در این کلاس وجود دارند که جملاً 9 گروه می‌شوند.



حال این سوال پیش می‌آید اگر بخواهیم گروه اختصاصی ایجاد کنم، چه کاری باید انجام شود. پس کد را به شکل زیر تغییر دهید:

```
internal class imageCopyrightUI : Module
{
    protected override void Initialize(IServiceProvider serviceProvider, ModuleInfo moduleInfo)
    {
        base.Initialize(serviceProvider, moduleInfo);
        IControlPanel controlPanel = (IControlPanel)GetService(typeof(IControlPanel));
        ModulePageInfo modulePageInfo = new ModulePageInfo(this, typeof(imageCopyrightUIPage),
        "Image Copyright", "Image Copyright", Resource1.VisualStudio_2012, Resource1.Visual_Studio_2012);

        ControlPanelCategorization areaCategorization = null;
        foreach (ControlPanelCategorization categorization in controlPanel.Categorizations) {
            if (categorization.Key == ControlPanelCategorization.AreaCategorization) {
                areaCategorization = categorization;
                break;
            }
        }
        ControlPanelCategoryInfo cat=new ControlPanelCategoryInfo("dotnettips","Dot Net Tips","This
is a Tutorial",areaCategorization);
        controlPanel.RegisterCategory(cat);
        controlPanel.RegisterPage(cat.Name,modulePageInfo);
    }
}
```

در خطوط `foreach` ما به دنبال نوع گروهی که قرار است مازوں ما در آن قرار بگیرد می‌گردیم و علاوه‌نماییم که گروه بندی ما در نوع Area باشد. برای همین این نوع را یافته و در متغیری از نوع `ControlPanelCategorization` قرار می‌دهیم. سپس توسط کلاس `ControlPanelCategoryInfo` یک نوع گروه جدید ایجاد کرده ایم که به ترتیب نام، عنوان، توضیح و نهایتاً نوع گروه بندی را در آن لحاظ می‌کنیم و سپس با دستور `controlPanel.RegisterCategory` گروه جدید خود را که در area قرار دارد، در IIS ثبت می‌کنیم و موقع افزودن صفحه مستقیماً نام گروه را در آن ذکر می‌کنیم. پروژه را Rebuild و IIS را مجدداً اجرا کنید. اگر Group by

شما روی Area باشد که به طور پیش فرض چنین است شما باید گروه Dot Net Tips را ببینید؛ حال از طریق لیست Group By گزینه‌ی Category را انتخاب نمایید. همانطور که مشاهده می‌کنید دوباره مازول شما در دسته‌ی Others قرار می‌گیرد؛ چرا که ما برای Area گروه بندی را لحاظ کرده بودیم. برای اینکه بتوانیم در دو حالت، دسته بندی داشته باشیم، کد را به شکل زیر تغییر می‌دهیم:

```
ControlPanelCategorization areaCategorization = null;
    ControlPanelCategorization CategoryCategorization = null;
foreach (ControlPanelCategorization categorization in controlPanel.Categorizations) {
    if (categorization.Key == ControlPanelCategorization.AreaCategorization) {
        areaCategorization = categorization;
    }
    if (categorization.Key == ControlPanelCategorization.CategoryCategorization)
    {
        CategoryCategorization = categorization;
    }
}
ControlPanelCategoryInfo cat1=new ControlPanelCategoryInfo("dotnettipsarea","Dot Net Tips
Area","This is a Tutorial",areaCategorization);
controlPanel.RegisterCategory(cat1);
controlPanel.RegisterPage(cat1.Name,modulePageInfo);

ControlPanelCategoryInfo cat2 = new ControlPanelCategoryInfo("dotnettipscat", "Dot Net Tips
Category", "This is a Tutorial", CategoryCategorization);
controlPanel.RegisterCategory(cat2);
controlPanel.RegisterPage(cat2.Name, modulePageInfo);
```

فکر کنم اتفاق بالا با توجه به مواردی که قبلایا در گرفتید با وضوح کامل مشخص باشد که اینبار دو حالت گروه بندی را ذخیره و هر کدام را جداگانه در کنترل پنل IIS ثبت کردیم.

بین سایر گزینه‌های کنترل پنل گزینه controlpanel.registerhomepage هم به چشم می‌خورد که اگر رابط کاربری را با این گزینه رجیستر کنیم صفحه اصلی IIS پریده و رابط کاربری ما جایگزینش می‌شود که البته این قسمت را باید با احتیاط با آن برخورد کرد و گرنه اگر بخواهیم همین رابط کاربری را به عنوان صفحه‌ی خانگی رجیستر کنیم، دسترسی ما به دیگر مازول‌ها قطع خواهد شد.

سورس پروژه

تا به اینجا این مبحث از سری آموزشی ما بسته می‌شود. در مقالات آینده موارد دیگری از IIS را مورد بررسی قرار خواهیم داد.

در این مطلب و همینطور مطلب بعدی قرار است به مبحث لاغ فایل‌ها Logfile بپردازیم. همانطور که می‌دانید سیستم IIS مثل هر سیستم دیگری لاغ‌هایی دارد که به مرور زمان این لاغ‌ها میتوانند مقدار زیادی از ظرفیت دیسک سخت را به خود اختصاص بدهند و این عمل میتواند موجب بروز مشکلاتی در سرور شود. به خوبی به یاد دارم که برای یکی از مشتریانم VPS تهیه نموده بودیم و بعد از یک سال با من تماس گرفت که سایت بالا نمی‌آید و وقتی بررسی شد، دیدم که از فضای دیسک سخت چند گیگابایتی، تنها چند مگابایت به طور ناچیز فضا برایش باقی مانده است و باعث شده است سرور از کار بیفتد. پس از بررسی متوجه شدم تمام این فضاهای توسط لاغ فایل‌ها پر شده است و از آنجا که سرویس دهنده تا مبلغی را به عنوان مدیریت سرور، ماهانه دریافت نکند، مدیریت این سرور مجازی را به عهده نداشته اند که البته بعد از انتقال به یک سرور دیگر از یک سرویس دهنده دیگر مشکلات ما در مورد سرور برای همیشه حل شد.

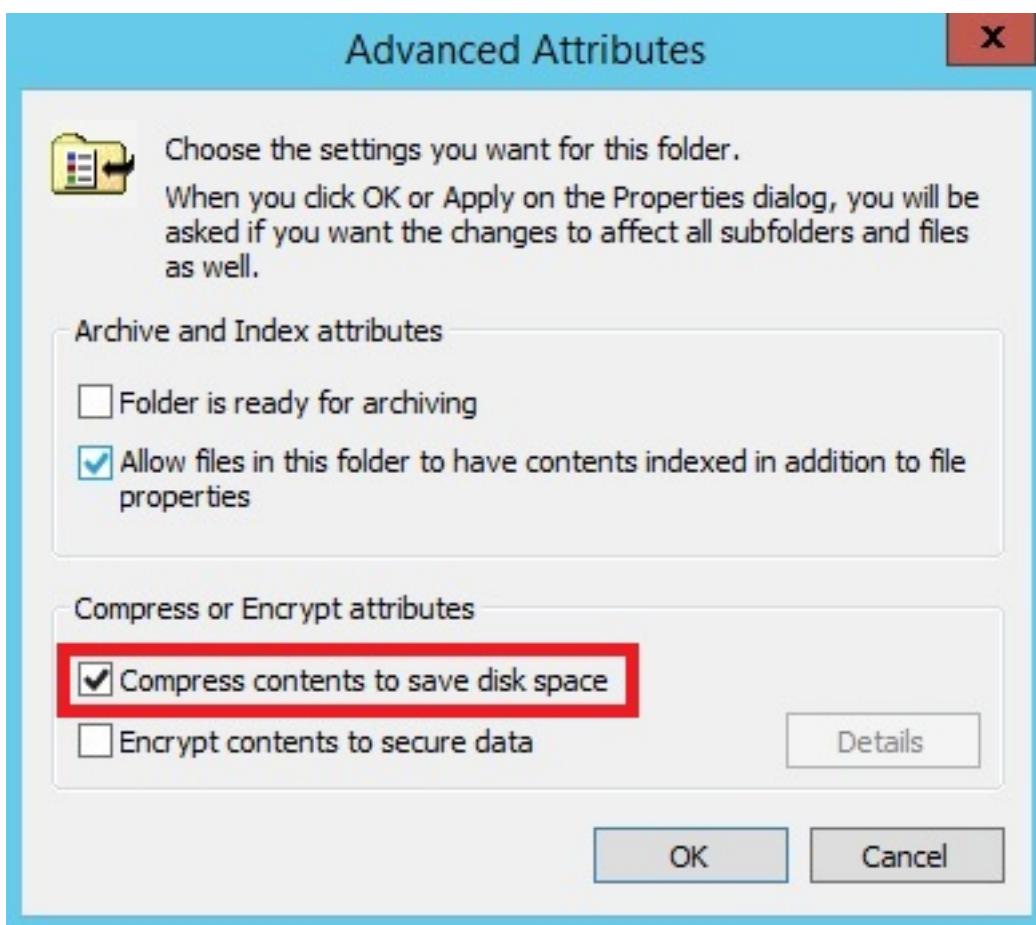
پس این داستان به خوبی روشن می‌کند که مدیریت این لاغ‌ها چقدر میتواند مهم و حیاتی باشد. آقای تیم وَن از تیم تحریریه مایکروسافت در بخش IIS موارد زیر را برای مدیریت لاغ‌ها بر می‌شمارد:

فعال سازی فشرده سازی
انتقال لاغ‌ها به یک سیستم راه دور
حذف لاغ فایل‌های قدیمی از طریق اسکریپت نویسی
حذف لاغ فایل‌های قدیمی توسط IIS Log File Cleaner

فشرده سازی دایرکتوری لاغ فایل‌ها

مسیر ذخیره لاغ فایل‌ها در آدرس زیر می‌باشد. به این آدرس رفته و Properties دایرکتوری مورد نظر را باز کنید و در برگه‌ی General آن، گزینه‌ی Advanced را زده تا در کادر جدیدی که باز می‌شود، گزینه‌ی Compress contents to save disk space را انتخاب کنید تا محتویات در هنگام ذخیره روی دیسک سخت فشرده شوند.

```
%SystemDrive%\inetpub\logs\LogFiles
```



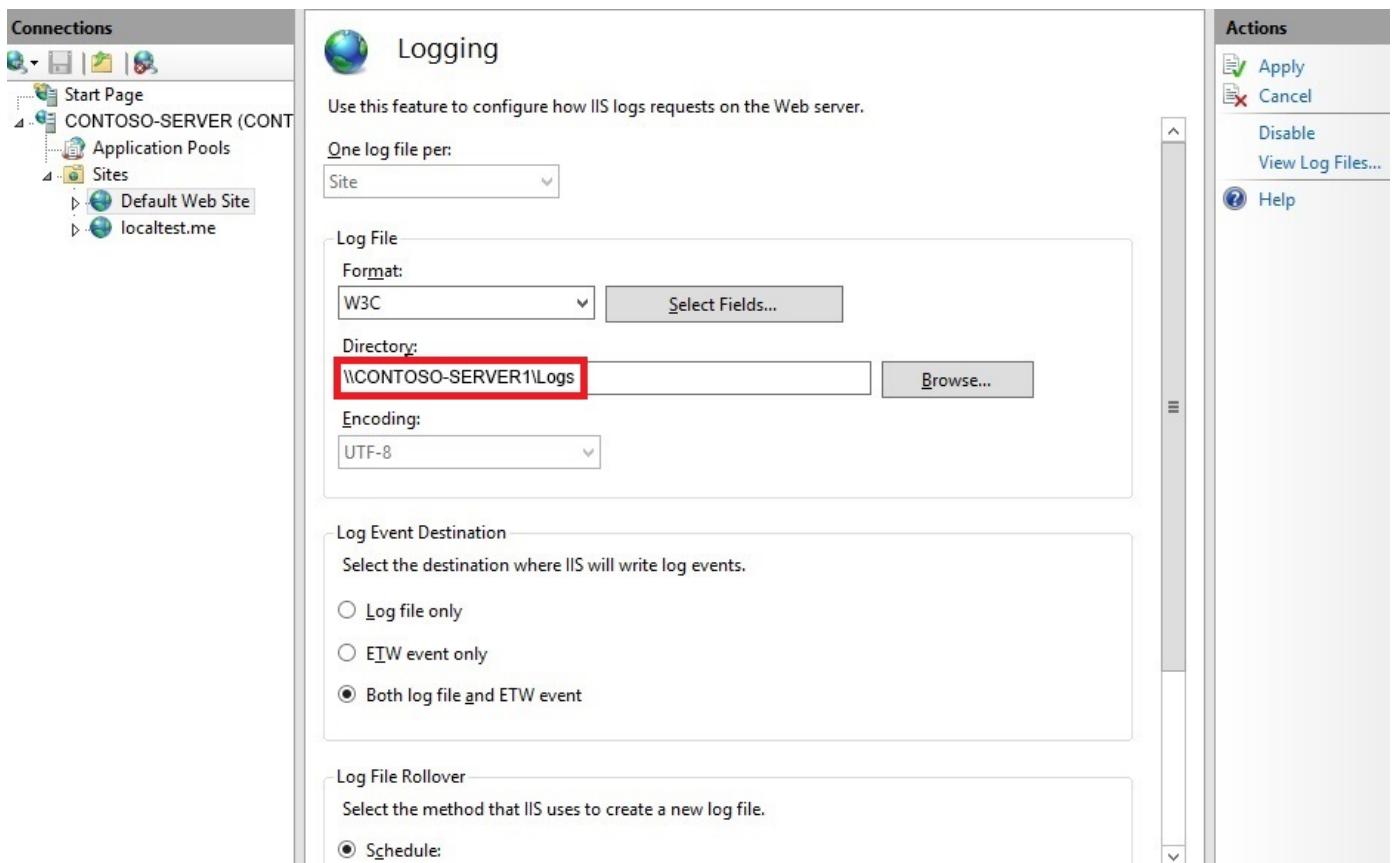
این روش ساده‌ترین روش موجود برای مدیریت لاغ هاست ولی روش نهایی نیست و باز به مرور زمان این روش هم کارایی خودش را از دست خواهد داد. این روش بیشتر شبیه خرید زمان می‌باشد تا اینکه یک راه حل نهایی برای حل مشکل باشد. البته این را هم باید مدنظر داشت که موقع تیک زدن گزینه بالا عملیات فشرده سازی باعث کند شدن سرعت کامپیوتر در حین آغاز عمل ذخیره سازی لاغ فایل‌ها هم خواهد شد. پس اگر قصد چنین کاری ذا دارید در ساعتی که سرور کمترین فشار از طرف کاربران را دارد یا اصطلاحاً پیک کاری آن پایین است انجامش دهید.

انتقال لاغ فایل‌ها به یک سیستم راه دور

همانطور که در بالا اشاره کردیم محل پیش فرض ذخیره سازی لاغ‌ها درمسیر %SystemDrive%\inetpub\logs\LogFiles

قرار دارد و این محل ذخیره سازی برای هر سرور یا حتی یک وب سایت خاص در صفحه تنظیمات Logging مشخص شده است و شما در میتوانید این لاغ‌ها را حتی برای کل سرور یا مربوط به یک سایت خاص، به سروری دیگر انتقال دهید. این امکان می‌تواند به امنیت سیستم هم کمک فراوانی کند تا اگر دیسک محلی Local Disk هم دچار مشکل شد، باز خواندن لاغ فایل‌ها میسر باشد و با استفاده از [ابزارهای تحلیل لاغ فایل‌ها](#)، آن‌ها را مورد بررسی قرار دهیم. برای تغییر محل ذخیره سازی لاغ‌ها به یک سیستم راه دور، راه حل زیر را طی کنید.

در IIS وب سایتی را که میخواهید لاغ آن انتقال یابد، انتخاب کنید؛ یا اگر لاغ کل سیستم IIS را میخواهید انتقال بدھید نام سرور را در لیست درختی انتخاب کنید و از مژویل‌های سمت راست، مژول Logging را انتخاب کنید و در قسمت Directory که محل ذخیره سازی فعلی لاغ‌ها را نوشته شده است، به صورت [UNC](#) آدرس دهی کنید. در آدرس زیر اولی نام سرور است - Contoso- server1\Logs نام پوشه‌ای که به اشتراک گذاشته شده است.



حذف لاغ فایل‌های قدیمی با استفاده از اسکریپت

با این روش میتوانید لاغ فایل‌هایی را که بعد از مدتی معین که دلخواه شما هست، از سیستم حذف نمایید و اگر این اسکریپت را زمان بندی خودکار نمایید، میتوانید از مراقبت مداوم و ثابت این کار نیز رها شوید.

با ستفاده از VBScript می‌کنیم که اگر مثلاً عمر لاغ فایل به 30 روز رسیده است، باید حذف شوند. خط دوم کد زیر نهایت عمر یک لاغ فایل را مشخص می‌کند:

```
sLogFolder = "c:\inetpub\logs\LogFiles"
iMaxAge = 30 'in days
Set objFSO = CreateObject("Scripting.FileSystemObject")
set colFolder = objFSO.GetFolder(sLogFolder)
For Each colSubfolder in colFolder.SubFolders
    Set objFolder = objFSO.GetFolder(colSubfolder.Path)
    Set colFiles = objFolder.Files
    For Each objFile in colFiles
        iFileAge = now - objFile.DateCreated
        if iFileAge > (iMaxAge+1) then
            objFSO.deletefile objFile, True
        end if
    Next
Next
```

اسکریپت بالا تمامی subfolders را برای همه سایتها بررسی کرده و لاغ‌های آنان را حذف می‌کند. ولی اگر دوست دارید این عملیات را تنها به یک وب سایت محدود کنید، باید مسیر را در خط اول دقیق‌تر مشخص کنید.

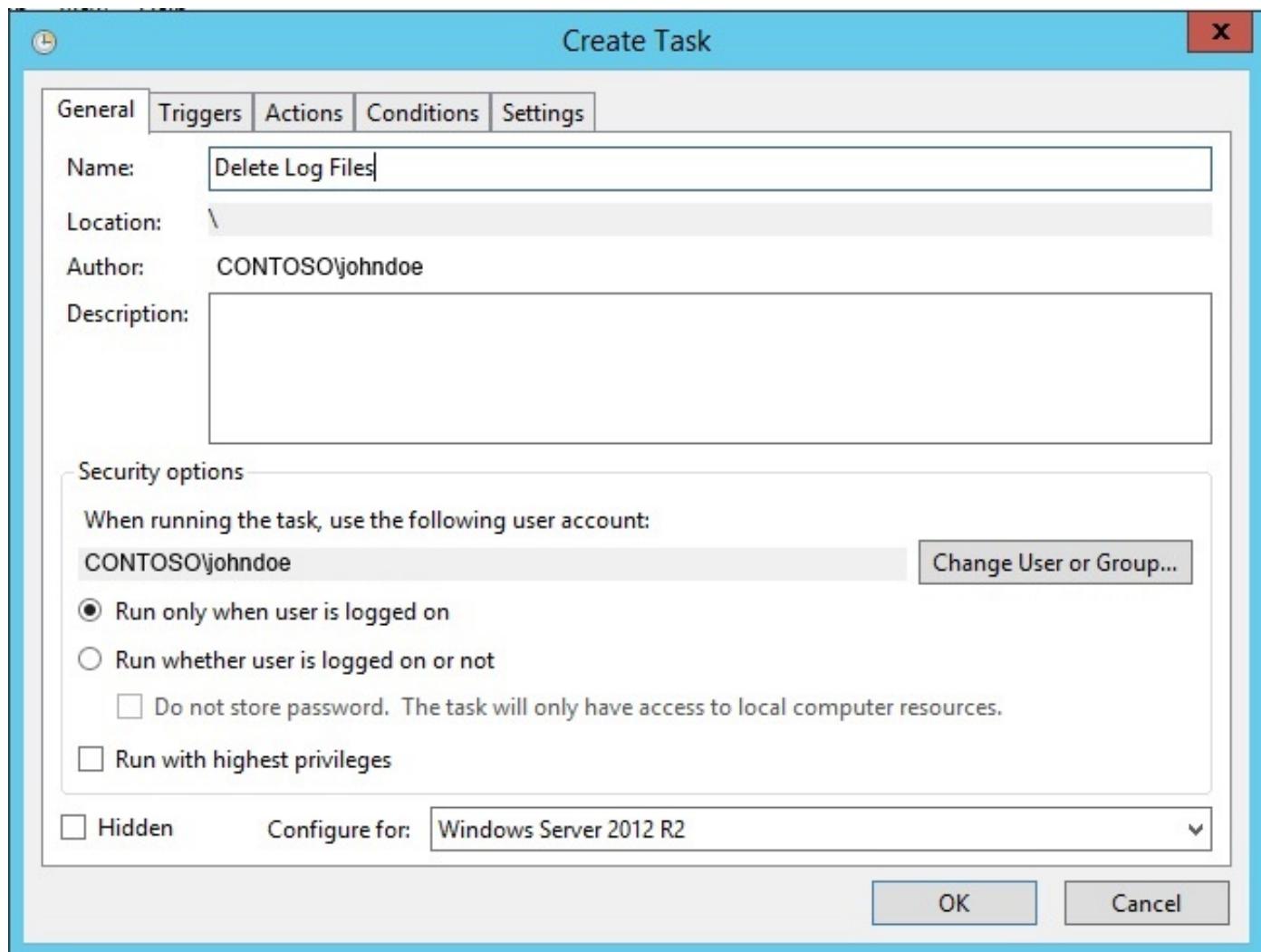
برای اجرای دستی اسکریپت در cmd تایپ کنید:

```
cscript.exe c:\scripts\retentionscript.vbs
```

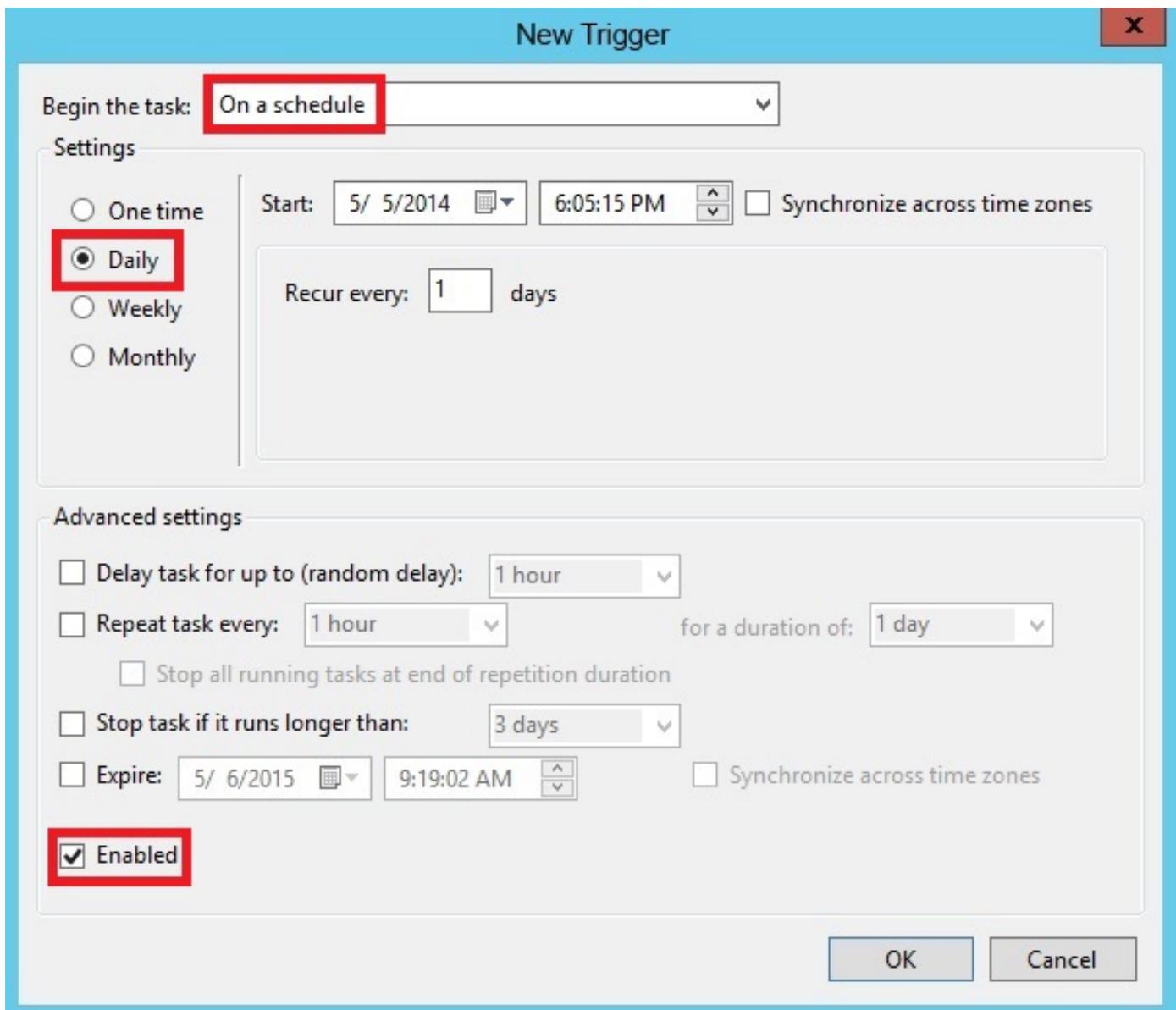
ولی اگر میخواهید این اسکریپت در هر دوره‌ی زمانی خاص اجرا شود، یا زمان بندی Scheduling گردد، دیگر مجبور نیستید هر بار به فکر نگهداری از لاغ‌ها باشید.

زمان بندی اجرای اسکریپت

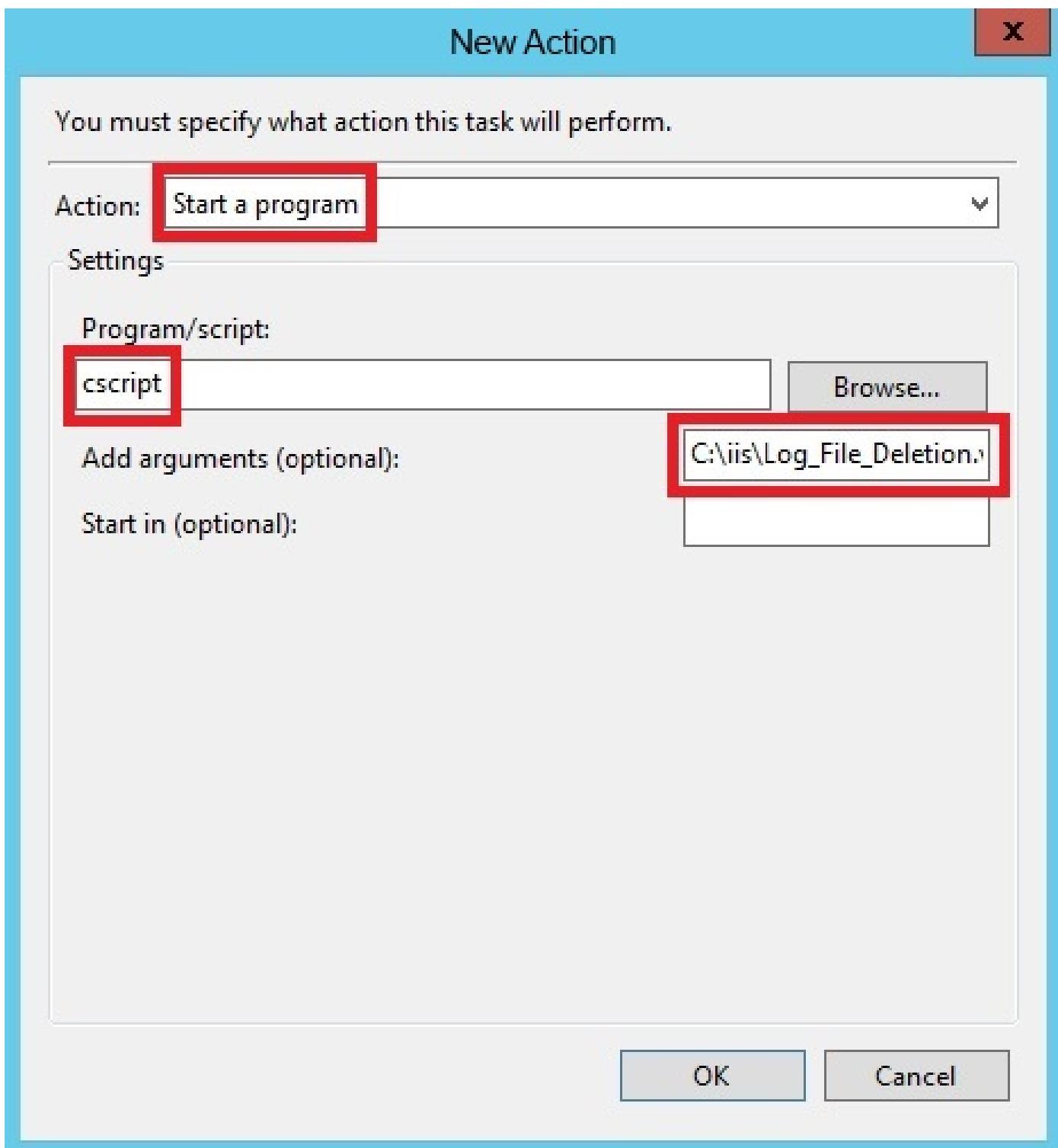
(قابل تست در ویندوز‌های سرور) را باز کرده و از منوی Tools گزینه Task Scheduler را انتخاب کنید و در قسمت گزینه Create Task Actions را انتخاب نمایید. در کادر باز شده نام "Delete Log Files" را برای مثال برگزینید و در قسمت Security هم کاربری که اجازه اجرای اسکریپت را دارد مشخص کنید.



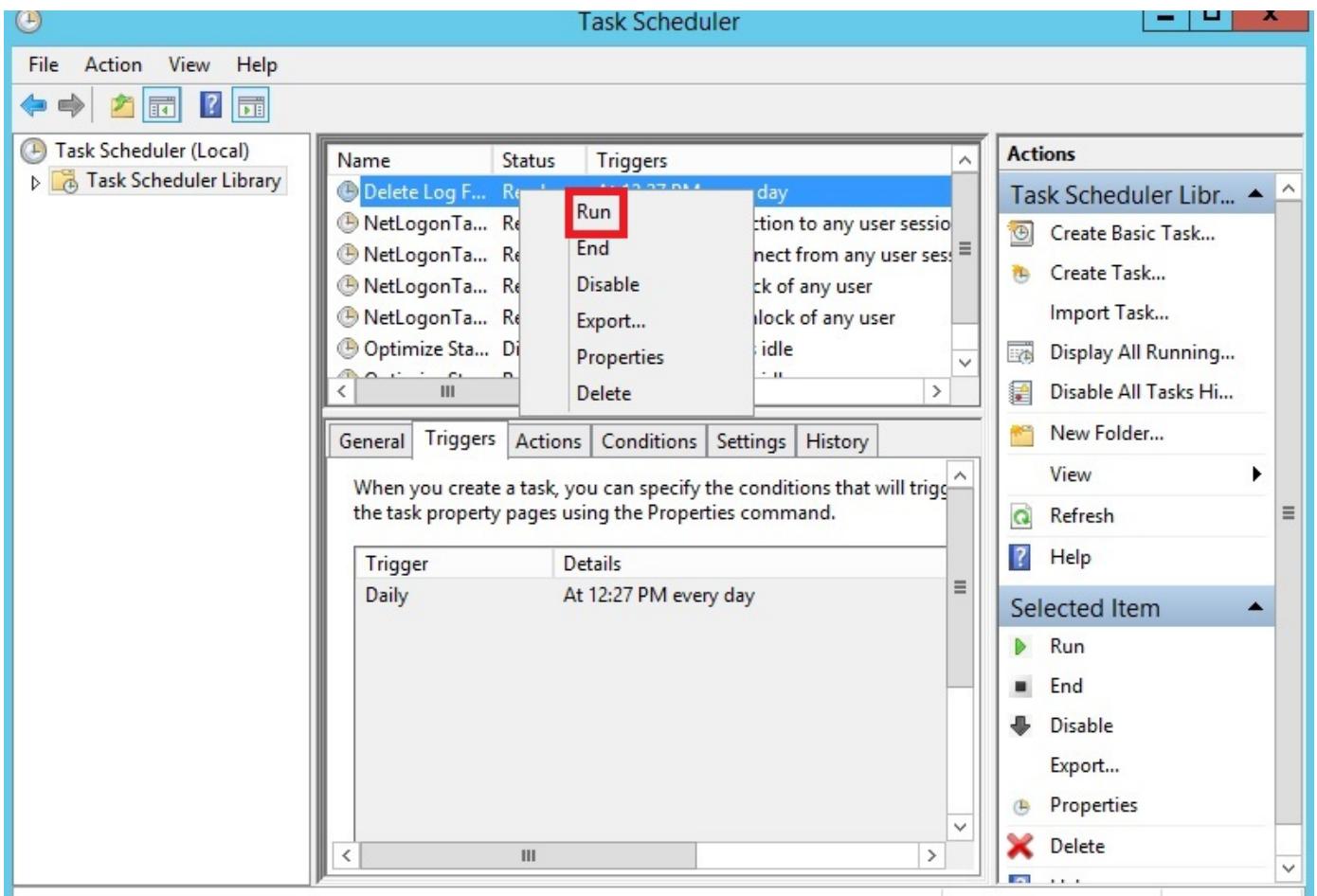
برگه Triggers را انتخاب کرده و گزینه New را انتخاب کنید و عملیات زمان بندی را تنظیم کنید و حتما بعد از زمان بندی مطمئن باشید که تیک Enabled فعال است.



در برگه Actions هم گزینه New را انتخاب کنید؛ در کادر باز شده از لیست Start a program را انتخاب کرده و در قسمت دستور cscript را ذکر نمایید و به عنوان آرگومان ورودی Program\script Add arguments هم مسیر اسکریپت خود را ذکر نمایید و کادر را تایید کنید.



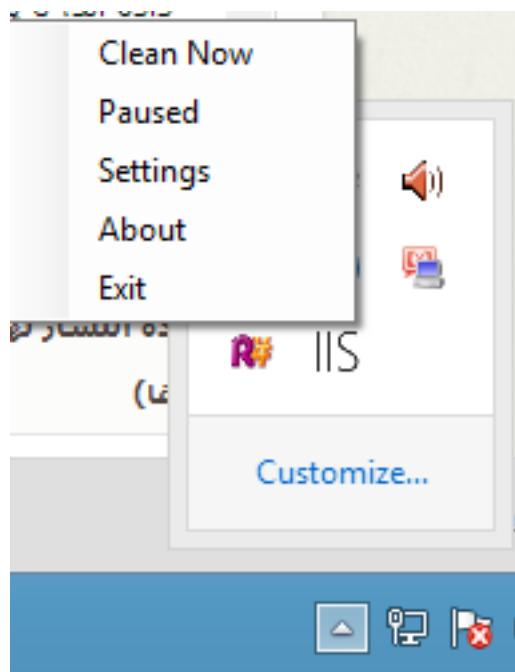
برای آغاز زمان بندی در لیست وظیفه های فعال active task pane، وظیفه ای که الان ساخته اید را اجرا کرده و به مسیر ذخیره لاغ ها رفته و می بینید که لاغ های مورد نظر حذف شده اند؛ پس از صحبت اجرای اسکریپت مطمئن می شویم. دوباره به لیست وظایف رفته و گزینه End را بزنید تا وظیفه، در حالت Ready قرار گیرد تا از همین الان فرایند زمان بندی اجرای اسکریپت آغاز شود.



حذف لاغ فایل‌ها با استفاده از IIS Log Cleaner Tools

ساده‌ترین ابزار برای مدیریت حذف لاغ فایل‌هاست که هر یک ساعت یکبار اجرا شده و لاغ فایل‌های تاریخ گذشته را که زمانش را شما تعیین می‌کنید، به سمت سطل زباله که البته درستش بازیافت است Recycle Bin انتقال میدهد تا از ضرر از دست دادن لاغ‌ها جلوگیری کند که بعداً شما میتوانید آن‌ها را به صورت دستی حذف کنید. همچنین عملیات خودکار حذف را نیز می‌توان متوقف نمود.

ابتدا برنامه را از [اینجا](#) دانلود کنید. موقعیکه برنامه را اجرا کنید، در نوتابلکیشن taskbar می‌نشینید و برنامه با یک پیغام به شما اعلام می‌کند، این اولین بار است که برنامه را باز کرده‌اید. پس یک سر به setting آن بزنید؛ با انتخاب گزینه‌ی settings برنامه بسته شده و فایل Settings.txt برای شما باز می‌شود که مدت زمان عمر لاغ فایل و مسیر ذخیره آن‌ها، از شما پرسیده می‌شود که مقدار عمر هر لاغ فایل به طور پیش فرض 30 روز و مسیر ذخیره‌ی لاغ‌ها همان مسیر پیش فرض IIS است که اگر شما دستی آن را تغییر داده اید، با پرسیدن آن، از محل لاغ‌ها اطمینان کسب می‌کند. در صورتی که قصد تغییری را در فایل، دارید آن را تغییر داده و ذخیره کنید و برنامه را مجدداً اجرا کنید.



نکات نهایی در مورد این برنامه :

اگر از ابزار IIS Cleaner Tool استفاده می‌کنید باید دستی سطل بازیافت را هم پاک کنید و هم اینکه میتوانید یک محدودیت حجمی برای Recycle Bin قرار دهید که اگر به یک حد رسید، خودکار پاک کند تا مشکلی برای سیستم عامل ایجاد نشود که البته به طور پیش فرض چنین است.

برنامه بالا به طور پیش فرض ریشه‌ی لاغ‌ها را حذف می‌کند. پس اگر میخواهید فقط سایت خاصی را مد نظر داشته باشد، آدرس دایرکتوری آن را اضافه کنید. البته چون این برنامه فقط روی یک دایرکتوری کار می‌کند و شما چند وب سایت دارید و مثلاً میخواهید سه تای آن‌ها را پاکسازی کنید، چاره‌ی جز استفاده از اسکریپت‌های با زمان بندی ندارید.

برنامه‌ی بالا فقط فایل‌هایی با پسوند `log` را به سطل بازیافت انتقال می‌دهد.

برنامه‌ی بالا یک سرویس نیست و باید به طور دستی توسط کاربر اجرا گردد. پس اگر ریست هم شد باید دستی اجرا شود یا آن را به داخل پوشه `startup` بکشید.

برنامه برای اجرایش نیاز به لاجین کاربر و مجوز نوشتن در آن پوشه را دارد تا به درستی کار کند.

در قسمت بعدی مبحث لاغ‌ها را ادامه خواهیم داد و با ماژول Logging در IIS و تنظیماتش آشنا خواهیم شد.

بیکربندی قسمت لاغ‌ها، میتواند برای یک سرور و یا وب سایت خاص از طریق فایل کانفیگ یا از طریق خود IIS انجام گیرد. برای اینکه به بیشتر این قابلیت‌ها در IIS دسترسی داشت، باید یکی از نسخه‌های ویندوز سرور 2012 و ویندوز 8 را نصب کرده باشد. لاغ‌ها به ثبت خطاهای HTTP می‌پردازند و با تحلیل آن‌ها میتوان عملیات بهینه سازی را بر روی سرو اجرا کرد. تمامی ثبت لاغ‌ها توسط Http.sys انجام می‌گیرد.

نحوه ذخیره سازی لاغها

در این بخش نحوه ذخیره سازی و فرمت ذخیره لاغ‌ها را در دو سطح سایت و سرور به طور جداگانه بررسی می‌کنیم. در IIS ماژول Logging را باز کنید و در لیست One log file per server می‌توانید مشخص کنید که لاغ‌ها در چه سطحی اجرا شوند. اگر گزینه باشد، تمامی خطاهای درخواست‌های رسیده به سرور در یک فایل لاغ ثبت می‌شوند. ولی اگر سطح سایت باشد، برای هر سایت بر روی IIS لاغ‌ها، جداگانه بررسی می‌شوند. به طور پیش فرض سطح سایت انتخاب شده است.

سطح سایت

موقعی که در لیست، سایت را انتخاب کنید، در لیست format می‌توانید تعیین کنید که لاغ‌ها به چه صورتی باید ذخیره شوند. مواردی که در این حالت لیست می‌شوند گزینه‌های W3C, IIS, NCSA, Custom می‌باشند که در زیر یکایک آن‌ها را بررسی می‌کنیم:

فرمت IIS: این فرمت توسط مایکروسافت ارائه شده و در این حالت لاغ‌های همه‌ی وب سایت‌ها ذخیره می‌شوند. به این فرم特 Fixed ASCII Based Text نیز می‌گویند؛ چرا که اجازه‌ی خصوصی سازی ندارد و نمی‌توانید بگویید چه فیلد‌هایی در لاغ قرار داشته باشند. لاغ فایل‌های این فرمت با، (کاما) از هم جدا می‌شوند و مقدار زمانی که برای هر فیلد ثبت می‌شود، به صورت محلی local Time می‌باشد.

فیلد‌هایی که در لاغ این نوع فرمت خواهند آمد، به شرح زیر است:

Client IP address

User name

Date

Time

Service and instance

Server name

Server IP address

Time taken

Client bytes sent

Server bytes sent

Service status code (A value of 200 indicates that the request was fulfilled successfully)

.Windows status code (A value of 0 indicates that the request was fulfilled successfully)

Request type

Target of operation

Parameters (the parameters that are passed to a script)

احتمال این وجود دارد که بعضی از فیلدها در بعضی رکوردها، شامل اطلاعاتی نباشند که به جای مقدار آن علامت - ثبت می‌گردد و برای کاراکترهایی که قابل نمایش نیستند یا کاراکتر نمایشی ندارند، از علامت + استفاده می‌شود. دلیل اینکار هم این است که ممکن است یک کاربر مهاجم، به ارسال اطلاعات کلیدهای کنترلی چون [Line Feed](#) یا [Carriage return](#) به اختصار CR به انتشار LF کند، که باعث شکسته شدن خط لاغ فایل می‌شود و در نتیجه از استاندارد خارج خواهد شد و هنگام خواندن آن هم با خطاب روپرتو می‌شویم؛ در نتیجه با جایگزینی چنین کاراکترهایی با + از این اتفاق جلوگیری می‌شود.

شكل زیر نمونه ای از یک خط لاغ در این فرمات است:

192.168.114.201, -, 03/20/01, 7:55:20, W3SVC2, SERVER, 172.21.13.45, 4502, 163, 3223, 200, 0, GET,
/DeptLogo.gif, -,

نام فیلد	نوع حالت مقداردهی	توضیح اتفاقات افتاده
Client IP address	192.168.114.201	آی پی کلاینت
User name	-	کاربر ناشناس است
Date	03/20/01	تاریخ فعالیت
Time	7:55:20	ساعت فعالیت
Service and instance	W3SVC2	
Server name	SERVER	لاغی که مربوط به سایت خاصی می‌شود به صورت W3SVC# نمایش داده می‌شود که علامت # شماره سایت می‌باشد که در اینجا این لاغ مربوط به سایت شماره 2 است
Server IP	172.21.13.45	نام سرور
Time taken	4502	آی پی سرور
Client bytes sent	163	چقدر انجام عملیات این درخواست به طول انجامیده است که بر حسب میلی ثانیه است.
Server bytes sent	3223	تعداد بایت هایی که از طرف کلاینت به سرور ارسال شده است
Service status code	200	تعداد بایت هایی که از طرف سرور به سمت کلاینت ارسال شده است
Windows status code	0	درخواست کاملاً موفقیت آمیز بوده است
Request type	GET	درخواست کاملاً موفقیت آمیز بوده است
Target of operation	/DeptLogo.gif	نوع درخواست کاربر
Parameters	-	کاربر قصد دانلود یک فایل تصویری GIF

نام فیلد	داشته است که نامش DeptLogo است	نوع حالت مقداردهی	توضیح اتفاقات افتاده
----------	--------------------------------	-------------------	----------------------

فرمت NCSA: این فرمت توسط مرکز علمی کاربردهای ابرمحاسباتی National Center for Supercomputing Applications ایجاد شده و دقیقاً مانند قبلی نمیتوان در آن نوع فیلدها را مشخص کرد و برای جدا سازی، از فاصله space استفاده می‌کند و ثبت مقدار زمان در آن هم به صورت محلی و هم UTC می‌باشد.
این فیلدها در لاغ آن نمایش داده می‌شوند:

Remote host address

Remote log name (This value is always a hyphen)

User name

Date, time, and Greenwich mean time (GMT) offset

Request and protocol version

Service status code (A value of 200 indicates that the request was fulfilled successfully)

Bytes sent

نمونه ای از یک لاغ ثبت شده:

```
172.21.13.45 - Microsoft\JohnDoe [08/Apr/2001:17:39:04 -0800] "GET /scripts/iisadmin/ism.dll?http/serv
HTTP/1.0" 200 3401
```

نام فیلد	مقدار ثبت شده	توضیح اتفاق افتاده
Remote host address	172.21.13.45	آی پی کلاینت
Remote log name	-	نام وجود ندارد
User name	Microsoft\JohnDoe	نام کاربری
Date, time, and GMT offset	[08/Apr/2001:17:39:04 -0800]	تاریخ و ساعت فعالیت به صورت محلی که 8 ساعت از مبدأ گرینویچ بیشتر است
Request and protocol version	GET /scripts/iisadmin/ism.dll?http/serv HTTP/1.0	کاربر با متد GET و Http نسخه‌ی یک، درخواست فایل ism.dll را کرده است.
Service status code	200	عملیات کاملاً موفقیت آمیز بود.
Bytes sent	3401	تعداد بایت‌های ارسال شده به سمت کاربر

امنیت در برابر کاربران مهاجم مانند همان فرمت قبلی صورت گرفته است.

فرمت W3C: توسط کنسرسیو姆 جهانی وب ارائه شده است و یک فرمت customizable ASCII text-based است. به این معنی که میتوان فیلدهایی که در گزارش نهایی می‌آید را خودتان مشخص کنید، که برای اینکار در کنار لیست، دکمه‌ی Select

وجود دارد که میتوانید هر کدام از فیلدهای را که خواستید، انتخاب کنید تا به ترتیب در خط لایگ ظاهر شوند. تاریخ ثبت به صورت UTC است.

نام فیلد	توضیح	به طور پیش فرض انتخاب شده است
Date	تاریخ رخدادن فعالیت	بله
Time	ساعت زمانی رخدادن فعالیت بر اساس UTC	بله
Client IP Addresses	آی پی کلاینت	بله
User Name	نام کاربری که هویت آن تایید شده و در صورتی که هویت تایید شده نباشد و کاربر ناشناس باشد، جای آن - قرار می‌گیرد	بله
Service Name and Instance Number	نام و شماره سایتی که درخواست در آن صورت گرفته است	خیر
Server Name	نام سروری که لایگ روی آن ثبت می‌شود	خیر
Server IP Address	آی پی سرور که لایگ روی آن ثبت می‌شود	بله
Server Port	شمره پورتی که سرویس مورد نظر را آن پورت اعمال می‌شود.	بله
Method	متد درخواست مثل GET	بله
URI Stem	هدف درخواست یا Target مثل index.htm	بله
URI Query	کوئری ارسال شده برای صفحات داینامیک	بله
HTTP Status	کد وضعیتی HTTP	بله
Win32 Status	کد وضعیتی ویندوز	خیر
Bytes Sent	تعداد بایت‌های ارسال شده به سمت کلاینت	خیر
Bytes Received	تعداد بایت‌های دریافت شده از سمت کلاینت	خیر
Time Taken	زمان به طول انجامیدن درخواست بر حسب میلی ثانیه	خیر
Protocol Version	درخواست با چه نسخه‌ای از پروتکل http یا ftp ارسال شده است	خیر
Host	اگر در هدر درخواست ارسالی این گزینه بوده باشد، نوشته خواهد شد.	خیر
User Agent	اطلاعات را از هدر درخواست می‌گیرد.	بله
Cookie	اگر کوکی رد و بدل شده باشد، محتويات کوکی ارسالی یا دریافت شده	خیر
Referrer	کاربر از چه سایتی به سمت سایت ما آمده است.	خیر
Protocol Substatus		بله

نام فیلد	توضیح
به طور پیش فرض انتخاب شده است	<p>در صورت رخدادن خطا در IIS ، کد خطا بازگردانده میشود. در IIS به منظور امنیت بیشتر و کاهش حملات، محتوای خطاهای رخداده در IIS به صورت متنی نمایش داده نمیشوند و شامل کد خطایی به اسم Substatus Code هستند تا مدیران شبکه با ردیابی لاغها پی به دلیل خطا و درخواست‌های ناموفق ببرند. برای مثال Error 404.2 به این معنی است که فایل درخواستی به دلیل قوانین محدود کنده، قفل شده و قابل ارائه نیست. ولی هکر تنها با خطای 404 یعنی وجود نداشتن فایل روبرو میشود. در حالت substatus کد شماره 2 را هم خواهد داشت که در لاغ ثبت میشود.</p> <p>هر شخصی که در سرور توانایی دسترسی به لاغها را داشته باشد، میتواند کد دوم خطا را نیز مشاهده کند. برای مثال مدیر سرور متوجه میشود که یکی از فایل‌های مورد نظر به کاربران، خطای 404 نمایش میدهد و با بررسی لاغها متوجه میشود که کد خطا 404.9 هست. از آنجا که ما همه‌ی کدها را حفظ نیستیم به این صفحه رجوع می‌کنیم و متوجه میشویم تعداد کاربرانی که برای این فایل، اتصال connection ایجاد کرده‌اند بیش از مقدار مجاز است و مدیر میتواند این وضع را کنترل کند. برای مثال تعداد اتصالات مجاز را نامحدود unlimited تعیین کند.</p>

حروف - و + برای موارد بالا هم صدق می‌کند. در ضمن گزینه‌های زیر در حالتی که درخواست از پروتکل FTP باشد مقداری خواهد گرفت:

uri-query

host

(User-Agent)

Cookie

Referrer

substatus

گزینه Custom : موقعی که شما این گزینه را انتخاب کنید مژول logging غیرفعال خواهد شد. زیرا این امکان در IIS قابل پیکربندی نیست و نوشتن مژول آن بر عهده شما خواهد بود؛ با استفاده از اینترفیس های ILogPlugin ، ILogPluginEx و ILogUIPlugin آن را پیاده سازی کنید.

ذخیره اطلاعات به انکدینگ UTF-8 و موضوع امنیت

در صورتی که شما از سایتی با زبانی غیر از انگلیسی و لاتین و فراتر از ANSI استفاده می کنید، این گزینه حتما باید انتخاب شده باشد تا درخواست را بهتر لگ کند. حتی برای وب سایت های انگلیسی زبان هم انتخاب این گزینه بسیار خوب است؛ چرا که اگر به سمت سرور کاراکتر های خاصی در URL ارسال شوند، نمی تواند با کدیج موجود آنها را درست تبدیل کند.

ادامه تنظیمات

موارد بعدی که در تنظیمات لاغ ها کاملا مشخص و واضح است، عملیات زمان بندی است که برای ساخت یک فایل لاغ جدید به کار می رود؛ برای مثال هر ساعت یک لاغ فایل جدید بسازد و فعالیت های موجود در هر ساعت در یک لاغ ذخیره می شوند. گزینه بعده حداکثر حجم هر فایل لاغ است که به صورت بایت مشخص می شود. اگر مقداری که تعیین می کنید کمتر از 1048576 بایت باشد، خودش به طور پیش فرض همان 1048576 بایت را در نظر خواهد گرفت. گزینه بعدی do not create a new logfile بدين معناست که همه لاغ ها در یک فایل ذخیره می شوند و فایل جدیدی برای لاغ ها ایجاد نمی شود.

گزینه آخری به اسم use local time for filenames and rollover است که اگر انتخاب شود، نامگذاری هر فایل لاغ بر اساس زمان محلی ساخت فایل لاغ خواهد بود. در صورتیکه انتخاب نشود، نامگذاری با زمان UTC درج خواهد شد.

سطح سرور

lag ها فقط در سمت سرور انجام می گیرد و لاغ هر سایت در یک فایل لاغ ثبت می شود. اگر بخواهید لاغ ها را در سطح سرور انجام دهید، گزینه binary هم اضافه خواهد شد.

Binary: در این گزینه دیگر از قالب بندی یا فرمت بندی لاغ ها خبری نیست و لاغ هر وب سایت به صورت اختصاصی صورت نمی گیرد. عملیات ذخیره سازی و ثبت هر لاغ می تواند از منابع یک سرور از قبیل حافظه و CPU و ... استفاده کند و اگر تعداد این وب سایت ها بالا باشد، باقی روش ها باعث فشار به سرور می شوند. برای همین ایجاد یک فایل خام از لاغ ها در این موقع می تواند راهگشا باشد. برای همه یک فایل لاغ ایجاد شده و بدون قالب بندی ذخیره می کند. پسوند این نوع لاغ ها .ibz است که مخفف Internet Binary Log می باشد. دلیل این تغییر پسوند این است که اطمینان کسب شود کاربر، با برنامه های متنی چون notepad یا امثال آن که به Text Utilities معروفند فایل را باز نمی کند. برای خواندن این فایل های میتوان از برنامه های [Log parser](#) استفاده کرد. پروتکل های FTP,NNTP و SMTP در این حالت لاگشان ثبت نمی شود.

در قسمت قبل «[کار با اسکنر در برنامه های تحت وب \(قسمت اول\)](#)» دیدی از کاری که قرار است انجام دهیم، رسیدیم. حالا سراغ یک پژوهشی عملی و پیاده سازی مطالب مطرح شده می رویم.

ابتدا پژوهشی WCF را شروع می کنیم. ویژوال استودیو را باز کرده و از قسمت New Project > Visual C# > WCF یک پژوهشی Service Application جدید را مثلًا با نام "WcfServiceScanner" ایجاد نمایید. پس از ایجاد، دو فایل IService1.cs و ScannerService موجود را به IService1.cs و IScannerService تغییر نام دهید. سپس ابتدا محتویات کلاس اینترفیس Service1.svc را به صورت زیر تعریف نمایید :

```
[ServiceContract]
public interface IScannerService
{
    [OperationContract]
    [WebInvoke(Method = "GET",
        BodyStyle = WebMessageBodyStyle.Wrapped,
        RequestFormat = WebMessageFormat.Json,
        ResponseFormat = WebMessageFormat.Json,
        UriTemplate = "GetScan")]
    string GetScan();
}
```

در اینجا ما فقط اعلان متدهای مورد نیاز خود را ایجاد کردیم. علت استفاده از Attribute ای با نام [WebInvoke](#)، مشخص نمودن نوع خروجی به صورت Json است و همچنین عنوان آدرس مناسبی برای صدا زدن متدهای [ScannerService](#) را مطابق کدهای زیر تغییر دهید:

```
public class ScannerService : IScannerService
{
    public string GetScan()
    {
        // TODO Add code here
    }
}
```

تا اینجا فقط یک WCF Service معمولی ساخته ایم. در ادامه به سراغ کلاس WIA برای ارتباط با اسکنر می رویم. بر روی پژوهشی خود راست کلیک کرده و Add Reference را انتخاب نموده و سپس در قسمت COM، گزینه [Image Acquisition Library v2.0](#) را به پژوهشی خود اضافه نمایید. با اضافه شدن این ارجاع به پژوهش، دسترسی به فضای نام WIA برای ما امکان پذیر می شود که ارجاعی از آن را در کلاس ScannerService قرار می دهیم.

```
using WIA;
```

اکنون متدهای GetScan را مطابق زیر اصلاح می نماییم:

```
public string GetScan()
{
    var imgResult = String.Empty;
    var dialog = new CommonDialogClass();
    try
    {
        نمایش فرم پیشفرض اسکنر //
        var image = dialog.ShowAcquireImage(WiaDeviceType.ScannerDeviceType);

        ذخیره تصویر در یک فایل موقت //
        var filename = Path.GetTempFileName();
```

```
image.SaveFile(filename);
var img = Image.FromFile(filename);

// img جهت ارسال سمت کاربر و نمایش در تگ
imgResult = ImageHelper.ImageToBase64(img, ImageFormat.Jpeg);
}

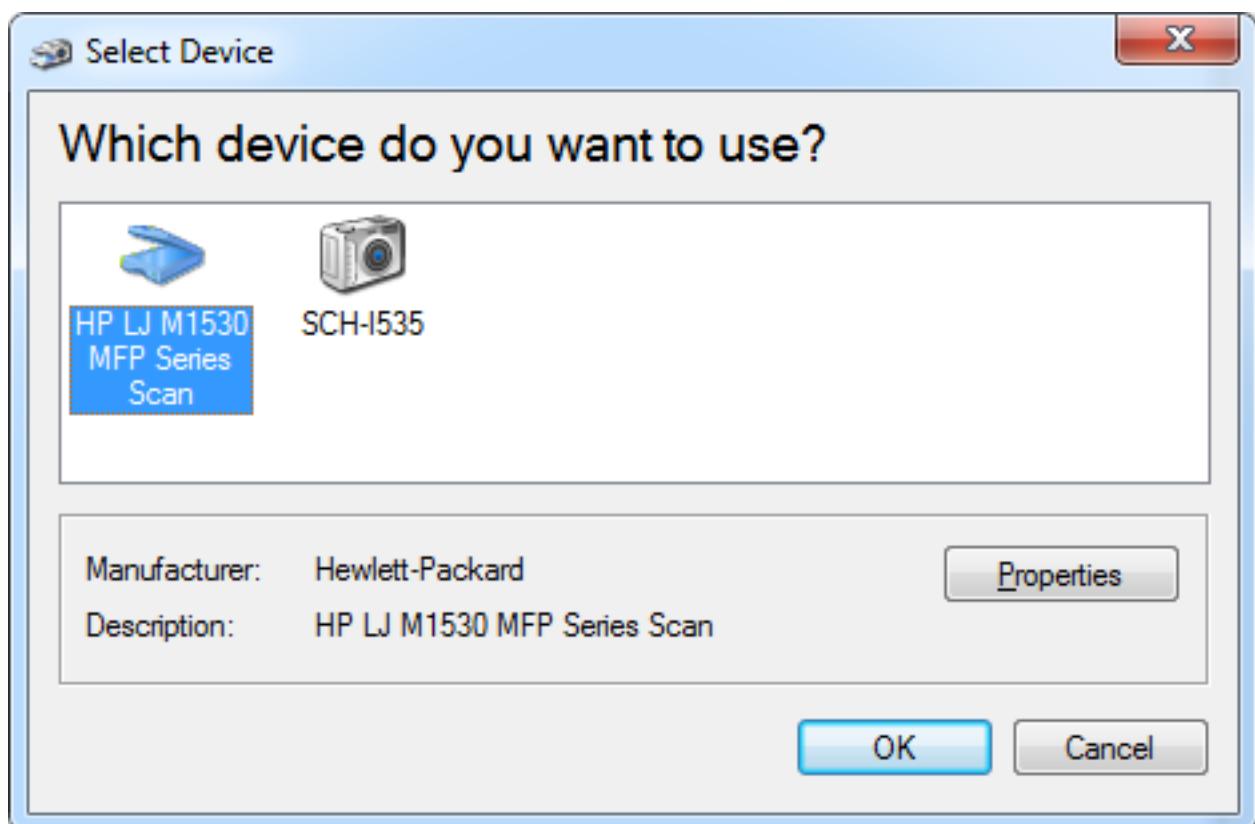
catch
{
    // از آنجاییه که امکان نمایش خطای وجود ندارد در صورت بروز خطای رشته خالی بازگردانده می شود که به معنای نبود تصویر می باشد //
}

return imgResult;
}
```

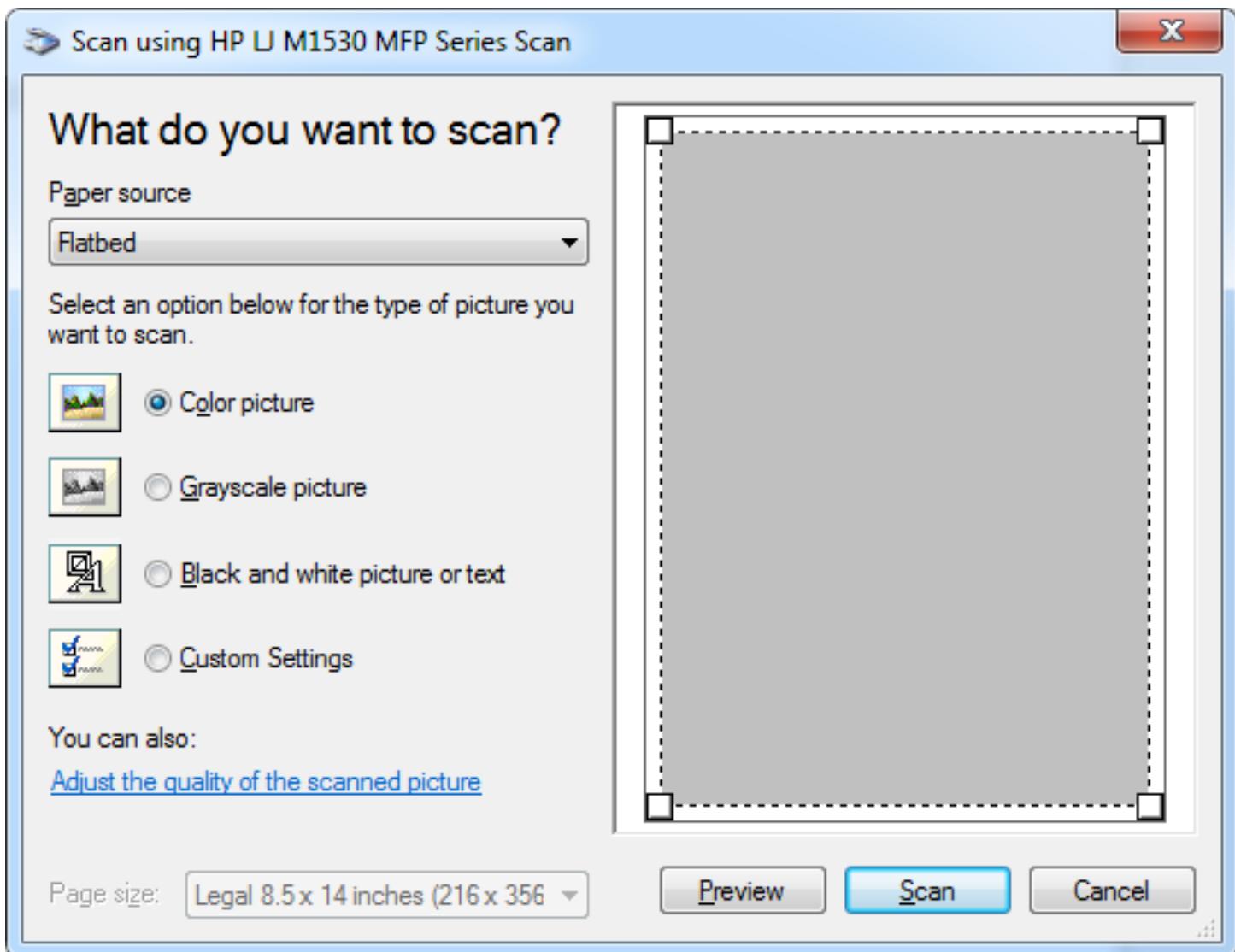
دقت داشته باشید که کدها را در زمان توسعه بین Try..Catch قرار ندهید چون ممکن است در این زمان به خطاهایی برخورد کنید که نیاز باشد در مرورگر آنها را دیده و رفع خطای نمایید.

اختیار ما قرار می دهد که بسته به نیاز خود می توانید از آنها استفاده کنید. برای نمونه در مثال ما نیز متد اصلی که مورد استفاده قرار گرفته، [ShowAcquireImage](#) می باشد که این متد، فرم پیش فرض دریافت اسکنر را به کاربر نمایش می دهد و کاربر از طریق آن می تواند قبل از شروع اسکن، یکسری تنظیمات را انجام دهد.

این متد ابتدا به صورت خودکار فرم تعیین دستگاه اسکنر ورودی را نمایش داده:



و سپس فرم پیش فرض اسکنرهای TWAIN را جهت تعیین تنظیمات اسکن نمایش می دهد که این امکان نیز در این فرم فراهم است تا دستگاههای Flated Feeder یا انتخاب گردنده.



خروجی این متدهمان عکس اسکن شده است که از نوع [WIA.ImageFile](#) میباشد و ما پس از دریافتش، ابتدا آن را در یک فایل موقت ذخیره نموده و سپس با استفاده از یک متدهمکی آن را به فرمت Base64 برای درخواست کننده اسکن ارسال مینماییم.

کدهای کلاس کمکی [ImageHelper](#):

```
public static string ImageToBase64(Image image, System.Drawing.Imaging.ImageFormat format)
{
    if (image != null)
    {
        using (MemoryStream ms = new MemoryStream())
        {
            // Convert Image to byte[]
            image.Save(ms, format);
            byte[] imageBytes = ms.ToArray();

            // Convert byte[] to Base64 String
            string base64String = Convert.ToBase64String(imageBytes);
            return base64String;
        }
    }
    return String.Empty;
}
```

توجه داشته باشید که خروجی این متد قرار است توسط `callBack` یک متدها اسکریپتی مورد استفاده قرار گرفته و احیاناً عکس مورد نظر در صفحه نمایش داده شود. پس بهتر است که از قالب تصویر به شکل Base64 استفاده گردد. ضمن اینکه پلاگین های Jquery مرتبط با ویرایش تصویر هم از این قالب پشتیبانی می کنند. ([اینجا](#)) این مثال به ساده ترین شکل نوشته شد. کلاس دیگری هم در [اینجا](#) وجود دارد و در صورتیکه از اسکنر نوع Feeder استفاده می کنید، می توانید از کدهای آن استفاده کنید.

کار ما تا اینجا در پروژه WCF Service تقریباً تمام است. اگر پروژه را یکبار Build نمایید برای اولین بار احتمالاً پیغام خطاهای زیر ظاهر خواهد شد:



جهت رفع این خطا، در قسمت Reference های پروژه خود، WIA را انتخاب نموده و از Properties های آن خصوصیت Embed را به False تغییر دهید؛ مشکل حل می شود.

به سراغ پروژه ویندوز فرم جهت هاست کردن این WCF سرویس می رویم. می توانید این سرویس را بر روی یک [Console App](#) یا [Windows Service](#) هم هاست کنید که در اینجا برای سادگی مثال، از WinForm استفاده می کنیم. یک پروژه WinForm جدید را ایجاد کنید و سپس از قسمت Solution > Add Reference به مسیر پروژه قبلی رفته و `WcfServiceScanner` را به پروژه خود اضافه نمایید. `Form1.cs` را باز کرده و ابتدا دو متغیر زیر را در آن به صورت عمومی تعریف نمایید:

```
private readonly Uri _baseAddress = new Uri("http://localhost:6019");
private ServiceHost _host;
```

برای استفاده از کلاس [ServiceHost](#) لازم است تا ارجاعی به فضای نام `System.ServiceModel` داده شود. متغیر `_baseAddress` نگه دارندهی آدرس ثابت سرویس اسکنر در سمت کلاینت می باشد و به این ترتیب ما دقیقاً می دانیم باید سرویس را با کدام آدرس در کدهای جاوا اسکریپتی خود فراخوانی نماییم.

حال در رویداد `Form_Load` برنامه، کدهای زیر را جهت هاست کردن سرویس اضافه می نماییم:

```
private void Form1_Load(object sender, EventArgs e)
{
    _host = new ServiceHost(typeof(WcfServiceScanner.ScannerService), _baseAddress);
    _host.Open();
}

private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    _host.Close();
}
```

همین چند خط برای هاست کردن سرویس روی آدرس localhost و پورت 8010 کامپیوتر کلاینت کافی است. اما یکسری تنظیمات مربوط به خود سرویس هم وجود دارد که باید در زمان پیاده سازی سرویس، در خود پروژه سرویس، ایجاد می گردد. اما از آنجا که ما قرار است سرویس را در یک پروژه دیگر هاست کنیم، بنابراین این تنظیمات را باید در همین پروژه WinForm قرار دهیم.

فایل `App.Config` پروژه WinForm را باز کرده و کدهای آنرا مطابق زیر تغییر دهید:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
  </startup>

  <system.serviceModel>
    <behaviors>
      <serviceBehaviors>
        <behavior name="BehaviourMetaData">
          <serviceMetadata httpGetEnabled="true" />
        </behavior>
      </serviceBehaviors>
    </behaviors>
    <services>
      <service name="WcfServiceScanner.ScannerService"
              behaviorConfiguration="BehaviourMetaData">
        <endpoint address=""
                  binding="basicHttpBinding"
                  contract="WcfServiceScanner.IScannerService" />
      </service>
    </services>
  </system.serviceModel>

</configuration>
```

اکنون پروژه‌ی هاست آماده اجرا می‌باشد. اگر آنرا اجرا کنید و در مرورگر خود آدرس ذکر شده را وارد کنید، صفحه‌ی زیر را مشاهده خواهد کرد که به معنی صحت اجرای سرویس اسکنر می‌باشد.

ScannerService Service

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the following syntax:

```
svcutil.exe http://localhost:6019/?wsdl
```

You can also access the service description as a single file:

```
http://localhost:6019/?singleWsdl
```

This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client class to call the Service. For example:

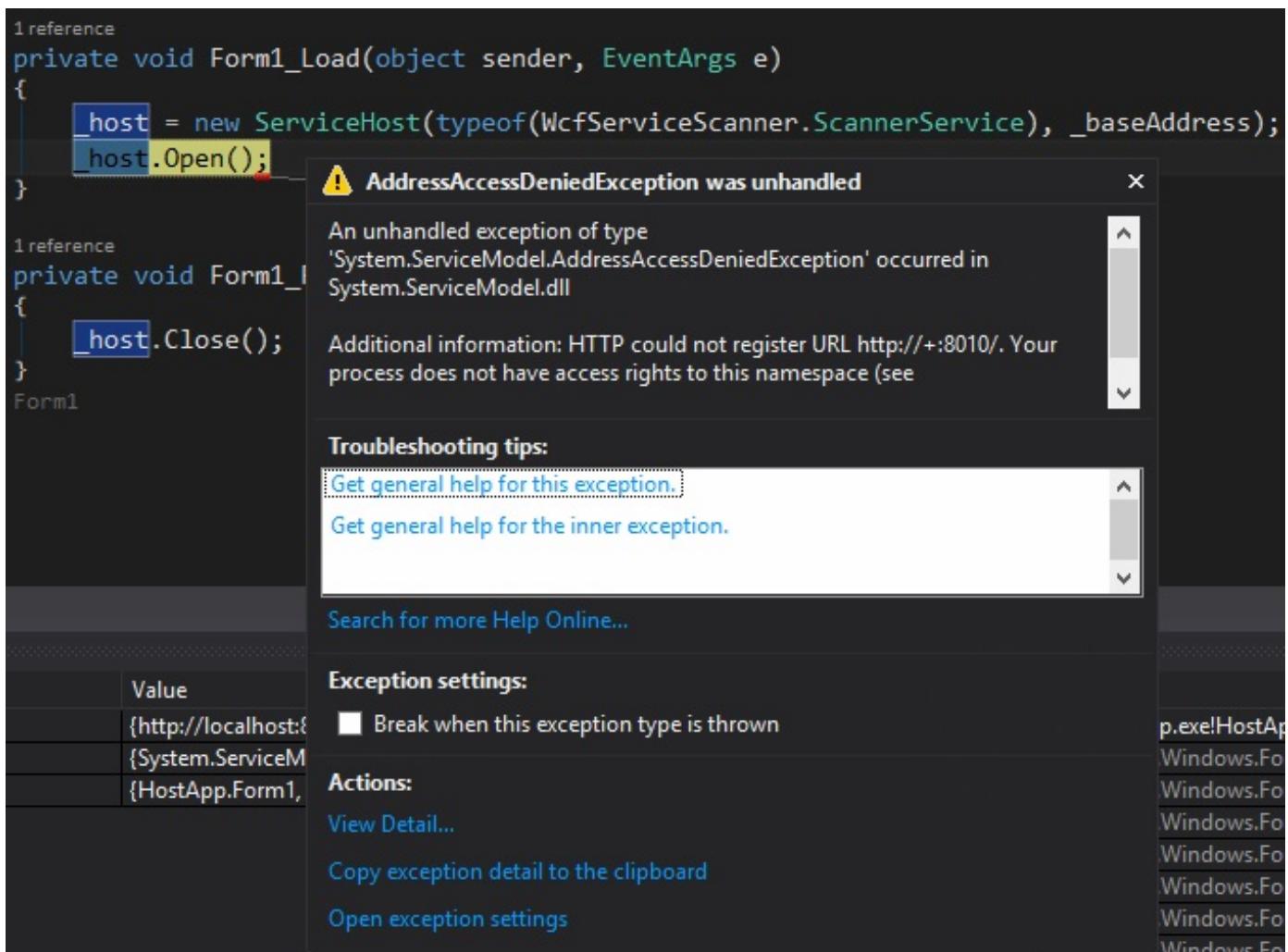
C#

```
class Test
{
    static void Main()
    {
        ScannerServiceClient client = new ScannerServiceClient();

        // Use the 'client' variable to call operations on the service.

        // Always close the client.
        client.Close();
    }
}
```

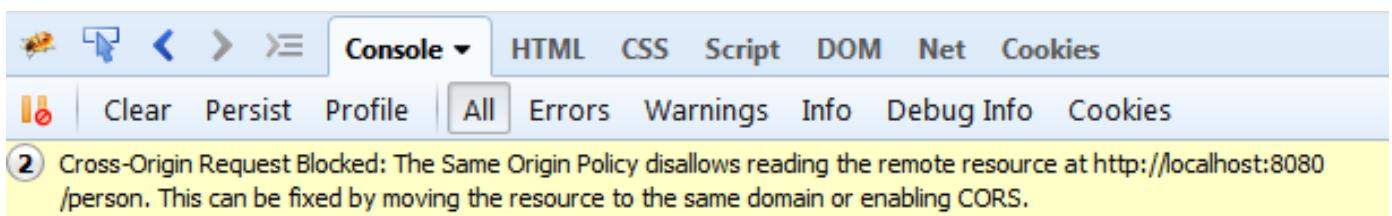
اگر موفق به اجرا نشدید و احیاناً با خطای زیر مواجه شدید، اطمینان حاصل کنید که ویژوال استودیو باشد. مشکل حل خواهد شد.



به سراغ پروژه‌ی بعدی، یعنی وب سایت خود می‌رویم. یک پروژه‌ی MVC جدید ایجاد نمایید و در View مورد نظر خود، کدهای زیر را جهت صدا زدن متند GetScan اضافه می‌کنیم.
از آنجا که کدها به صورت جاوا اسکریپت می‌باشد، پس مهم نیست که حتماً پروژه‌ی MVC باشد؛ یک صفحه‌ی HTML ساده هم کافی است).

```
<a href="#" id="get-scan">Get Scan</a>
<img src="" id="img-scanned" />
<script>
    $("#get-scan").click(function () {
        var url = 'http://localhost:6019/';
        $.get(url, function (data) {
            $("#img-scanned").attr("src", "data:image/jpeg;base64, " + data.GetScanResult);
        });
    });
</script>
```

دقت کنید در هنگام دریافت اطلاعات از سرویس، نتیجه به شکل GetScanResult خواهد بود. الان اگر پروژه را اجرا نمایید و روی لینک کلیک کنید، اسکن شروع به دریافت اسکن خواهد کرد اما نتیجه‌ای بازگشت داده نخواهد شد و علت هم مشکل امنیتی CORS می‌باشد که به دلیل دریافت اطلاعات از یک دامین دیگر رخ می‌دهد و اگر با Firebug درخواست را بررسی کنید متوجه خطا به شکل زیر خواهید شد.



راه حل های زیادی برای این مشکل ارائه شده است، و متأسفانه بسیاری از آنها در شرایط پروژه های ما جوابگو نمی باشد (به دلیل هاست روی یک پروژه ویندوزی). تنها راه حل مطمئن (تست شده) استفاده از یک کلاس سفارشی در پروژه WCF Service می باشد که مثال آن در [اینجا](#) آورده شده است.

برای رفع مشکل به پروژه WcfServiceScanner بازگشته و کلاس جدیدی را به نام CORSSupport ایجاد کرده و کدهای زیر را به آن اضافه کنید:

```
public class CORSSupport : IDispatchMessageInspector
{
    Dictionary<string, string> requiredHeaders;
    public CORSSupport(Dictionary<string, string> headers)
    {
        requiredHeaders = headers ?? new Dictionary<string, string>();
    }

    public object AfterReceiveRequest(ref System.ServiceModel.Channels.Message request,
System.ServiceModel.IClientChannel channel, System.ServiceModel.InstanceContext instanceContext)
    {
        var httpRequest = request.Properties["httpRequest"] as HttpRequestMessageProperty;
        if (httpRequest.Method.ToLower() == "options")
            instanceContext.Abort();
        return httpRequest;
    }

    public void BeforeSendReply(ref System.ServiceModel.Channels.Message reply, object correlationState)
    {
        var httpResponse = reply.Properties["httpResponse"] as HttpResponseMessageProperty;
        var httpRequest = correlationState as HttpRequestMessageProperty;

        foreach (var item in requiredHeaders)
        {
            httpResponse.Headers.Add(item.Key, item.Value);
        }
        var origin = httpRequest.Headers["origin"];
        if (origin != null)
            httpResponse.Headers.Add("Access-Control-Allow-Origin", origin);

        var method = httpRequest.Method;
        if (method.ToLower() == "options")
            httpResponse.StatusCode = System.Net.HttpStatusCode.NoContent;
    }
}

// Simply apply this attribute to a DataService-derived class to get
// CORS support in that service
[AttributeUsage(AttributeTargets.Class)]
public class CORSSupportBehaviorAttribute : Attribute, IServiceBehavior
{
    #region IServiceBehavior Members

    void IServiceBehavior.AddBindingParameters(ServiceDescription serviceDescription,
ServiceHostBase serviceHostBase, System.Collections.ObjectModel<ServiceEndpoint> endpoints,
BindingParameterCollection bindingParameters)
    {
    }

    void IServiceBehavior.ApplyDispatchBehavior(ServiceDescription serviceDescription,
ServiceHostBase serviceHostBase)
    {
        var requiredHeaders = new Dictionary<string, string>();
        //Chrome doesn't accept wildcards when authorization flag is true
    }
}
```

کار با اسکنر در برنامه های تحت وب (قسمت دوم و آخر)

```
//requiredHeaders.Add("Access-Control-Allow-Origin", "*");
requiredHeaders.Add("Access-Control-Request-Method", "POST,GET,PUT,DELETE,OPTIONS");
requiredHeaders.Add("Access-Control-Allow-Headers", "Accept, Origin, Authorization, X-
Requested-With,Content-Type");
requiredHeaders.Add("Access-Control-Allow-Credentials", "true");
foreach (ChannelDispatcher cd in serviceHostBase.ChannelDispatchers)
{
    foreach (EndpointDispatcher ed in cd.Endpoints)
    {
        ed.DispatchRuntime.MessageInspectors.Add(new CORSSupport(requiredHeaders));
    }
}
void IServiceBehavior.Validate(ServiceDescription serviceDescription, ServiceHostBase
serviceHostBase)
{
}
#endregion
}
```

فضاهای نام لازم برای این کلاس به شرح زیر می‌باشد:

```
using System.ServiceModel;
using System.ServiceModel.Channels;
using System.ServiceModel.Description;
using System.ServiceModel.Dispatcher;
```

کلاس ScannerService را باز کرده و آنرا به ویژگی

```
[CORSSupportBehavior]
public class ScannerService : IScannerService
{
```

مزین نمایید.

کار تمام است، یکبار دیگر ابتدا پروژه‌ی WcfServiceScanner و سپس پروژه‌ی هاست را Build کرده و برنامه‌ی هاست را اجرا کنید. اکنون مشاهده می‌کنید که با زدن دکمه‌ی اسکن، اسکنر فرم تنظیمات اسکن را نمایش می‌دهد که پس از زدن دکمه‌ی Scan، پروسه آغاز شده و پس از اتمام، تصویر اسکن شده در صفحه‌ی وب سایت نمایش داده می‌شود.

نظرات خوانندگان

نویسنده: امیر بختیاری
تاریخ: ۱۳۹۳/۱۱/۲۱ ۱۴:۴۸

با سلام و تشکر خدمت شما دوست گرامی

من تمام مراحل را به دقت ایجاد کردم ولی بازم همون خطایی که خودتون هم عنوان کردید را میده

Cross-Origin Request Blocked: The Same Origin Policy disallows reading the remote resource at <http://localhost:13748/>. This can be fixed by moving the resource to the same domain or enabling CORS.

فایل پروژه را به طور کامل میزارم ببینید علت چیه [WcfServiceScanner.zip](#)

با تشکر مجدد

نویسنده: شروین ایرانی
تاریخ: ۱۳۹۳/۱۱/۲۱ ۹:۵

با سلام؛ آیا امکانش هست که سورس پروژه‌ی مثال رو هم قرار بدید؟ تشکر

نویسنده: امیر بختیاری
تاریخ: ۱۳۹۳/۱۱/۲۱ ۱۰:۳۱

سورس پروژه را از نظر قبلی می‌تونید دریافت کنید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۱/۲۲ ۱۶:۱۵

این مثال را بررسی کنید: [SelfHostWcfScannerService.zip](#)

- ابتدا سرویس self host آن را با دسترسی ادمین اجرا کنید.
- بعد برنامه‌ی وب را اجرا کنید.

نویسنده: امیر بختیاری
تاریخ: ۱۳۹۳/۱۱/۲۲ ۱۷:۳۰

با تشکر

این مثال به قسمت Catch رفته و عکس پیشفرض رو نشان می‌دهد در حالی که بر روی کامپیوتر من اسکنر نصب است و با دموی DynamSoft هم به خوبی جواب گرفتم و مشکلی نداشتم به نظرتون چرا پنجره اسکنر رو باز نمیکنه، اسکنر مجازی میشه نصب کرد و تست گرفت

نویسنده: محمد رعیت پیشه
تاریخ: ۱۳۹۳/۱۱/۲۲ ۱۷:۵۲

اینجا رو بررسی کنید و ببینید خطایی که دریافت می‌کنید توضیحش چیست. موردی که خودم تست کردم این بود که وقتی روی سیستمی که اسکنر متصل نبود امتحان می‌کردم، به خطای خودم و به محض اتصال اسکنر خطای بطرف می‌شد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۱/۲۲ ۱۸:۱۷

در همان فایل ScannerService اگر catch را به این صورت تغییر دهید، استثناءها در کنسول سرویس نمایش داده می‌شوند.

```
catch(Exception ex)
{
    Console.WriteLine(ex);
```

برای مثال این مورد را من دریافت می کنم:

```
System.Runtime.InteropServices.COMException (0x80210015):
Exception from HRESULT: 0x80210015 at WIA.CommonDialogClass.ShowAcquireImage
```

کد [0x80210015](#) به معنای عدم یافتن اسکنر است (که صحیح است).

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۱/۲۲ ۱۸:۵۱

یک نکته‌ی تكميلی

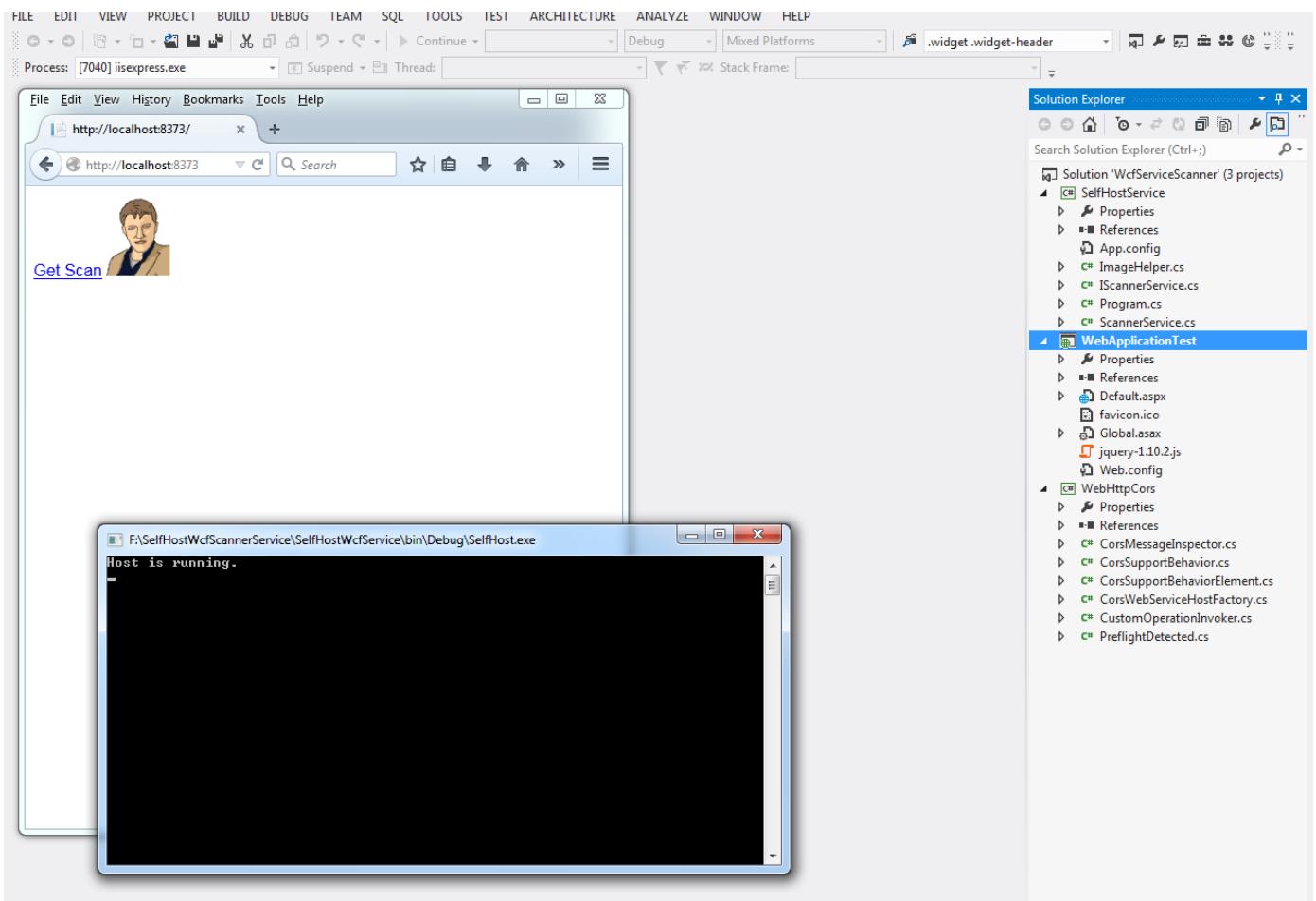
در متده `GetScan` در پایان کار، نیاز است منابع COM مورد استفاده رها شوند تا نشی حافظه رخ ندهد:

```
finally
{
    if(image!=null) Marshal.ReleaseComObject(image);
    if(dialog!=null) Marshal.ReleaseComObject(dialog);
}
```

نویسنده: شروین ایرانی
تاریخ: ۱۳۹۳/۱۱/۲۸ ۱۰:۹

با سلام؛ من یه عکسی تهیه کردم از مراحل اجرای برنامه. ابتدا پروژه `SelfHostService` را `Build` کدم و سپس فایل `exe` را `Run` کردم و کنسول باز شد. سپس پروژه `WebApplicationTest` را بوسیله مرورگر فایرفاکس اجرا کردم. ولی تصویر پیش فرض را لود کرد. ضمنا در `FireBug` هم خطایی نداد. مشکل کجاست. اسکنر هم نصب هست. مرسى.

کار با اسکنر در برنامه های تحت وب (قسمت دوم و آخر)



نویسنده: وحید نصیری

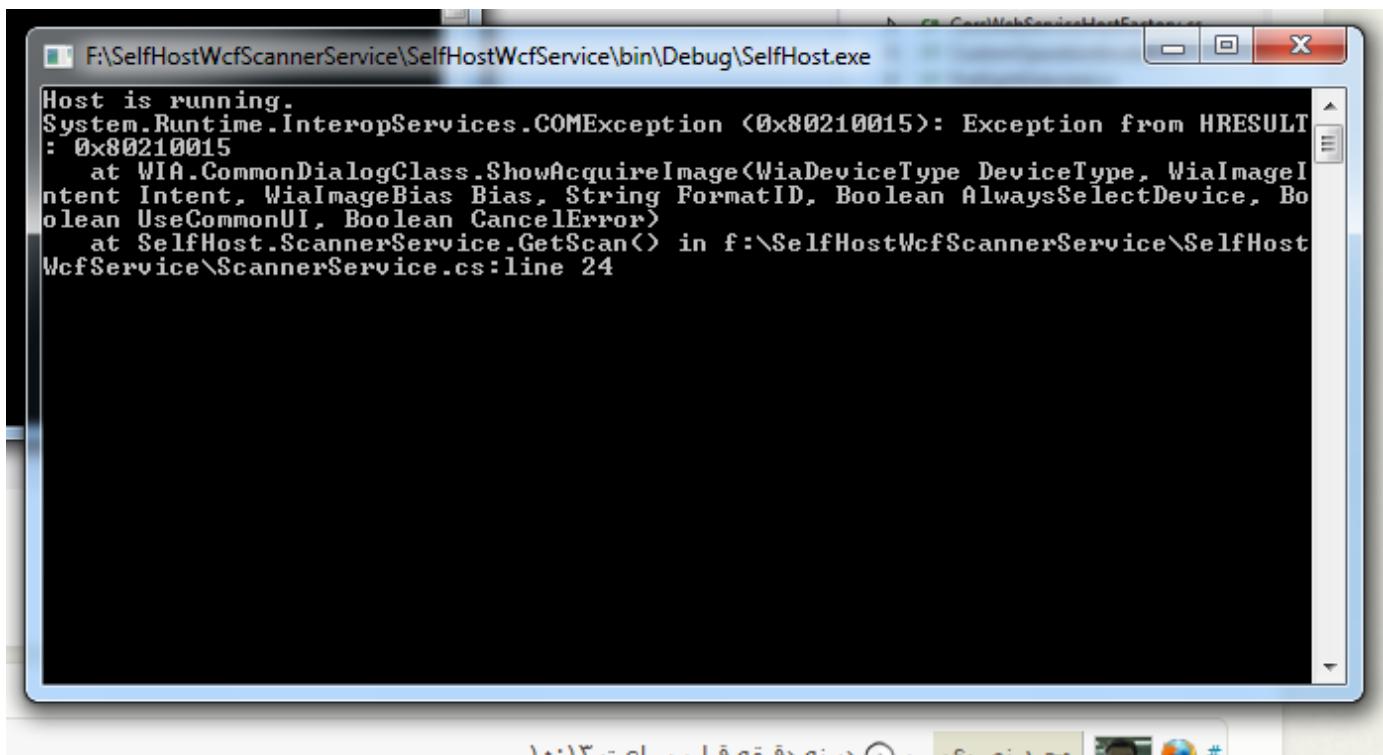
تاریخ: ۱۰:۱۳ ۱۳۹۳/۱۱/۲۸

در نظرات قبلی عنوان شد: «...در همان فایل catch اگر ScannerService را به این صورت تغییر دهید ...»

نویسنده: شروین ایرانی

تاریخ: ۱۰:۲۴ ۱۳۹۳/۱۱/۲۸

بله انجام شد. خطای 0x80210015 میده. ولی اسکنر کاملا نصب و کار میکنه در محیطهای دیگه.



نویسنده: وحید نصیری
تاریخ: ۱۱:۸ ۱۳۹۳/۱۱/۲۸

- معنای دوم خطای 0x80210015 عدم سازگاری درایورهای اسکنر شما با WIA است (Windows Image Acquisition). اسکنر شما احتمالاً فقط WIA or TWAIN compatible است و TWAIN compatible نیست.
- [لیستی از این نوع اسکنرهای](#)

نویسنده: محمد رضا برخورداری
تاریخ: ۱۲:۱۱ ۱۳۹۳/۱۲/۱۴

با سلام؛ من تمامی مراحل رو طی کردم اما با خطای زیر مواجه شدم :

```
Host is running.
System.Runtime.InteropServices.COMException (0x80070050): The file exists. (Exception from HRESULT: 0x80070050)
at WIA.ImageFileClass.SaveFile(String Filename)
at SelfHost.ScannerService.GetScan() in g:\project\asnad\scan\SelfHostWcfScannerService\SelfHostWcfService\ScannerService.cs:line 25
```

نویسنده: وحید نصیری
تاریخ: ۱۲:۳۴ ۱۳۹۳/۱۲/۱۴

سطر Path.GetFileName را تبدیل کنید به مسیری که به آن دسترسی دارید. همچنین بررسی کنید نام این فایل منحصر بفرد باشد.

نویسنده: محمد رضا برخورداری
تاریخ: ۱۹:۴۱ ۱۳۹۳/۱۲/۱۶

بسیار ممنون، مشکل حل شد. یه سوال دیگه : آیا در WIA امکان اسکن دو رو هم وجود داره؟ به چه شکلی

نویسنده: محمد رعیت پیشه
تاریخ: ۱۴:۴۲ ۱۳۹۴/۰۱/۱۴

پاسخ شما اسکن کردن در قالب تصاویر tiff هست، که می‌توانند چندین عکس در یک عکس باشند، مثل اسکن پشت و رو کارت ملی.

فقط برای نمایش در مرورگر به مشکل برخورد می‌کنید که یک راه استفاده از افزونه [AlteraTIFF](#) می‌باشد.

نویسنده: محمد رضابرخوردار
تاریخ: ۱۳:۱۲ ۱۳۹۴/۰۱/۱۵

باعث خطای The file exists Path.GetTempFileName می‌شود. زمانیکه با آدرس قابل دسترس و نام ثابت برای فایل جایگزین میکنم برای بار دوم همین خط را میدهد.

نویسنده: محمد رعیت پیشه
تاریخ: ۲۳:۳۴ ۱۳۹۴/۰۱/۱۶

به نظرم یکتا بودن آدرس فایل در سمت کلاینت ضروری نمی‌باشد، در اینصورت می‌توانید یک مسیر ثابت و مشخص رو در نظر بگیرید و قبل از ذخیره، وجود اون رو بررسی کنید و چنانچه از قبل وجود داشت ابتدا آن را پاک نموده سپس ذخیره کنید.

نویسنده: فطرسی
تاریخ: ۲۰:۴۶ ۱۳۹۴/۰۳/۲۳

با سلام و عرض خسته نباشد.

خیلی ممنون بابت زحمتی که کشیدید و بسیار عالی است. من می‌خواستم که پروژه SelfHostService رو به صورت Windows Service استفاده کنم البته می‌دونم که برای اینکار نمی‌شه چون پنجره اسکنر رو نمیشه نشون بده ولی شما فرض کنید همون رشته که در صورت خطا بر میگردونه رو بخواهم برگردانم به صفحه. حالا دوباره به مشکل Cors برخورد کردم در حالت قبل شما خیلی خوب اون رو حل کردید ولی برای ویندوز سرویس هم میشه قرار بدھید. با تشکر این خیلی به دردم می‌خوره.

نویسنده: محسن خان
تاریخ: ۲۳:۴۶ ۱۳۹۴/۰۳/۲۳

پروژه دومین نظر این بحث از بالا، سرویس را درون یک برنامه‌ی ویندوزی اجرا کرده.

نویسنده: محمد رعیت پیشه
تاریخ: ۱:۲ ۱۳۹۴/۰۳/۲۴

اتفاقاً من این پروژه رو روی ویندوز سرویس هاست کرده بودم و مشکلی هم نداشت!
در واقع این شما نیستید که بخواهید فرمی را نشان بدھید، این مورد بر عهده کتابخانه WIA می‌باشد.
برای اون مشکل یعنی با کلاس CORSSupportBehavior هم حل نشده؟
بحث‌های نصب یک سرویس که خارج از مطلب هست اما به سادگی می‌توانید پیدا کنید، پس از نصب سرویس شما در متد OnStart این کدها را خواهید داشت

```
_host = new ServiceHost(typeof(WcfServiceScanner.ScannerService), _baseAddress);
_host.Open();
```

و در متد OnStop هم

```
_host.Close();
```

بقیه موارد فرقی با هاست روی یک پروژه کنسول نمی‌کند.

در مبحث قبلی گفتیم که ویرایش تنظیمات لاغ‌ها از طریق IIS یا ویرایش مستقیم فایل‌های کانفیگ میسر است. در این مقاله که قسمت پایانی مبحث لاغ هاست، در مورد ویرایش فایل‌های کانفیگ صحبت می‌کنیم؛ همچنین استفاده از دستورات appcmd برای ویرایش و نهایتاً کد نویسی در زبان سی‌شارپ و جاوااسکریپت.

تنظیمات لاغ سایت‌ها در فایل applicationhost در آدرس زیر قرار دارد:

C:\Windows\System32\inetsrv\config\applicationHost.config

برای هر تگ سایت، یک تگ <logfile> وجود دارد که ویژگی‌های Attributes آن، نوع ثبت لاغ را مشخص می‌کند و می‌توانید مستقیماً در اینجا به ویرایش بپردازید. البته ویرایش فایل کانفیگ از طریق IIS به طور مستقیم هم امکان پذیر است. برای این منظور در IIS سرور را انتخاب و از بین مارژول‌های قسمت management گزینه‌ی Configuration Editor را انتخاب کنید. در قسمت گزینه‌ی System.applicationhost را باز کرده و از زیر مجموعه‌های آن گزینه‌ی Site را برگزینید. در تنظیمات باز شده، گزینه collection را انتخاب کنید تا در انتهای سطر، دکمه‌ی ... پدیدار گردد. روی آن کلیک کنید تا محیط ویرایشی باز گردد که به شما اجازه‌ی افزودن و ویرایش خصوصیت‌ها را می‌دهد. برای ویرایش لاغ‌ها باید خصوصیت logfile را باز کنید. اگر قسمت قبلی را مطالعه کرده باشید، باید بسیاری از این خصوصیت‌ها و مقادیر را بشناسید.

خصوصیات دیگری را هم مشاهده خواهید کرد که شاید قبل از نمایه شدن IIS شما دارد؛ مثلاً خصوصیت‌های flushByEntryCountW3Clog و maxLogLineLength از IIS8.5 اضافه شده‌اند.

جدول خصوصیت‌ها

خصوصیت	توضیح
customLogPluginClsid	یک پارامتر رشته‌ای اختیاری که در آن، آی دی کلاس یا کلاس‌هایی نوشته می‌شود که برای custom logging نوشته شده‌اند و این گزینه ترتیب اجرای آن‌ها را تعیین می‌کند.
directory	اختیاری است. محل ذخیره‌ی لاغ فایل‌ها را مشخص می‌کند و در صورتیکه ذکر نشود، همان مسیر پیش فرض است.
enabled	اختیاری است. فعال بودن سیستم لاغ برای آن سایت را مشخص می‌کند. مقدار پیش فرض آن true است.
flushByEntryCountW3CLog	این مقدار مشخص می‌کند چند رخداد باید اتفاق بیفتد تا عمل ذخیره سازی لاغ صورت گیرد. اگر بعد از هر رخداد عمل ثبت لاغ انجام شود، سرعت ثبت لاغ‌ها بالا می‌رود؛ ولی باعث استفاده‌ی مداوم از منابع و همچنین درخواست ثبت اطلاعات را روی دیسک خواهد داد و تاوان آن با زیاد شدن عملیات روی دیسک، پرداخته خواهد شد. ولی در حالتیکه چند رخداد را نگهداری سپس دسته‌ای ثبت کند، باعث افزایش کارآیی و راندمان سرور خواهد شد. در صورتیکه سرور به مشکلات لحظه‌ای برخورد می‌کند مقدار آن را کاهش دهید. مقدار پیش فرض 0 است. یعنی اینکه ثبت، بعد از 64000 لاغ خواهد بود.
localTimeRollover	نحوه‌ی نامگذاری فایل‌های لاغ را مشخص می‌کند که مقدار بولین گرفته و اختیاری است. به طور پیش فرض مقدار false دارد.

خصوصیت	توضیح
<code>logExtFileFlags</code>	این گزینه در حالتی به کارتان می‌آید که فرمات W3C را برای ثبت لاغ‌ها انتخاب کرده باشید و در اینجا مشخص می‌کنید که چه فیلدهایی باید در لاغ باشند و اگر بیش از یکی بود میتوان با، (کاما) از هم جداشان کرد.
<code>logFormat</code>	اختیاری است و مقدار پیش فرض آن <code>true</code> است. بدین معنا که کد یا شماره‌ی سایت هم در لاغ خواهد بود و این در حالتی است که گزارش در سطح سرور باشد. در غیر این صورت اگر هر سایت، جداگانه لاغی برای خود داشته باشد، ذکر نمی‌گردد.
<code>logSiteId</code>	اختیاری است و مقدار <code>file</code> و <code>* ETW</code> را می‌گیرد که به طور پیش فرض روی <code>File</code> تنظیم است. در این حالت فایل لاغ‌ها در یک فایل متنه توسط <code>http.sys</code> ذخیره می‌شود. ولی موقعیه از <code>iislogprovider</code> استفاده می‌شود، <code>http.sys</code> با استفاده از <code>ETW</code> داده‌ها را به سمت <code>ETW</code> ارسال می‌کند که منجر به اجرای سرویس <code>Logsvc</code> شده که از داده‌ها کوئری گرفته و آن‌ها را مستقیماً از پروسه‌های کارگر جمع آوری و به سمت فایل لاغ ارسال می‌کند. همچنین انتخاب این دو گزینه نیز ممکن است.
<code>logTargetW3C</code>	حداکثر تعداد خطی که یک لاغ میتواند داشته باشد تا اینکه بتوانید در مصرف دیسک سخت صرفه جویی کنید و بیشتر کاربرد آن برای لاغ‌های کاستوم است. این عدد باید از نوع <code>UInt</code> باشد و اختیاری است و از 2 تا 65536 مقدار می‌پذیرد که مقدار پیش فرض آن 65536 می‌باشد.
<code>maxLogLineLength</code>	همان مبحث زمان بندی در مورد ایجاد فایل‌های لاغ است که در مقاله‌ی پیشین بررسی کردیم و مقادیر <code>maxsize</code> و <code>Daily,Hourly,monthly,weekly</code> هم هست؛ موقعی که لاغ به نهایت حجمی که برای آن تعیین کردیم میرسد.
<code>period</code>	اختیاری است و مقدار آن از نوع <code>int64</code> است. حداکثر حجم یک فایل لاغ را مشخص می‌کند تا اگر <code>period</code> روی <code>maxsize</code> تنظیم شده بود، حداکثر حجم را میتوان از اینجا تعیین نمود. در مقاله پیشین در این باره صحبت کردیم؛ حداقل عدد برای آن 1,048,576 است و اگر کمتر از آن بنویسید، سیستم همین عدد 1,048,576 را در نظر خواهد گرفت. مقدار پیش فرض آن 20971520 می‌باشد.
<code>truncateSize</code>	

به غیر از خصوصیات بالا، خصوصیت `customFields` نیز از 8.5 IIS (به بعد) در دسترس است. اگر قصد دارید به غیر از فیلدهای W3C فیلدهای اختصاصی دیگری نیز داشته باشید، میتوان از این گزینه استفاده کرد. این فیلدهای کاستوم می‌توانند اطلاعاتشان را از `server variables` و `request header`، `response header` دریافت کنند. این ویژگی تنها در فرمات W3C و در سطح سایت قابل انجام است. موقعی که یک فایل لاغ شامل فیلدهای اختصاصی شود، به انتهای نام فایل `_X` اضافه می‌گردد تا نشان دهد شامل یک فیلد

اختصاصی یا کاستوم است. نحوه تعریف آن در فایل applicationhost به شکل زیر است:

```
<system.applicationHost>
  <sites>
    <siteDefaults>
      <logFile logFormat="W3C"
        directory="%SystemDrive%\inetpub\logs\LogFiles"
        enabled="true">
        <customFields>
          <clear/>
          <add logFieldName="ContosoField" sourceName="ContosoSource"
            sourceType="ServerVariable" />
        </customFields>
      </logFile>
    </siteDefaults>
  </sites>
</system.applicationHost>
```

تغییر تنظیمات لگ با Appcmd

```
appcmd.exe set config -section:system.applicationHost/sites /siteDefaults.logFile.enabled:"True"
/commit:apphost
appcmd.exe set config -section:system.applicationHost/sites /siteDefaults.logFile.logFormat:"W3C"
/commit:apphost
appcmd.exe set config -section:system.applicationHost/sites
/siteDefaults.logFile.directory:"%SystemDrive%\inetpub\logs\LogFiles" /commit:apphost
```

تنظیمات تگ لگ با برنامه نویسی و اسکریپت نویسی هچنین با رفرنس Microsoft.web.administration در پروژه‌های دات نتی خود میتوانید امکان ویرایش تنظیمات را در برنامه‌های خود نیز داشته باشید:

```
using System;
using System.Text;
using Microsoft.Web.Administration;

internal static class Sample
{
    private static void Main()
    {
        using (ServerManager serverManager = new ServerManager())
        {
            Configuration config = serverManager.GetApplicationHostConfiguration();
            ConfigurationSection sitesSection = config.GetSection("system.applicationHost/sites");
            ConfigurationElement siteDefaultsElement = sitesSection.GetChildElement("siteDefaults");

            ConfigurationElement logFileElement = siteDefaultsElement.GetChildElement("logFile");
            logFileElement["logFormat"] = @"W3C";
            logFileElement["directory"] = @"%SystemDrive%\inetpub\logs\LogFiles";
            logFileElement["enabled"] = true;

            serverManager.CommitChanges();
        }
    }
}
```

با استفاده از اسکریپت نویسی توسط جاوااسکریپت و وی بی اسکریپت هم نیز این امکان مهیا است:

```
var adminManager = new ActiveXObject('Microsoft.ApplicationHost.WritableAdminManager');
adminManager.CommitPath = "MACHINE/WEBROOT/APPHOST";
var sitesSection = adminManager.GetAdminSection("system.applicationHost/sites",
" MACHINE/WEBROOT/APPHOST");
var siteDefaultsElement = sitesSection.ChildElements.Item("siteDefaults");

var logFileElement = siteDefaultsElement.ChildElements.Item("logFile");
logFileElement.Properties.Item("logFormat").Value = "W3C";
logFileElement.Properties.Item("directory").Value = "%SystemDrive%\inetpub\logs\LogFiles";
logFileElement.Properties.Item("enabled").Value = true;

adminManager.CommitChanges();
```

FTP Logging

برای اطمینان از نصب Ftp logging موقع نصب، باید از مورد زیر مطمئن باشید: IIS را باز کنید و در لیست درختی، سرور را انتخاب کنید. در قسمت FTP میتوانید گزینه‌ی Ftp logging را ببینید. تنظیمات این قسمت هم دقیقاً همانند قسمت logging میباشد و همان موارد برای آن هم صدق می‌کند.

بررسی تگ آن در applicationhost

تگ این نوع لاغ در فایل applicationhost در زیر مجموعه‌ی تگ <site> به شکل زیر نوشته می‌شود:

```
<system.ftpServer>
  <log centralLogFileMode="Central">
    <centralLogFile enabled="true" />
  </log>
</system.ftpServer>
```

گزینه centralLogFileMode دو مقدار central و site را می‌پذیرد. اگر گزینه‌ی central انتخاب شود، یعنی همه‌ی لاغ‌ها را داخل یک فایل در سطح سرور ثبت کن ولی اگر گزینه‌ی site انتخاب شده باشد، لاغ‌های هر سایت در یک فایل ثبت خواهد شد.

گزینه‌ی logInUTF8 یک خصوصیت اختیاری است که مقدار پیش فرض آن true میباشد. در این حالت باید تمامی رشته‌ها به انکدینگ UTF-8 تبدیل شوند.

همانطور که می‌بینید تگ log در بالا یک تگ فرزند هم به اسم centralLogFile دارد که همان خصوصیات جدول بالا در آن مهیاست.

دسترسی به تنظیمات این قسمت توسط دستور Appcmd

```
appcmd.exe set config -section:system.ftpServer/log /centralLogFileMode:"Central" /commit:apphost
appcmd.exe set config -section:system.ftpServer/log /centralLogFile.enabled:"True" /commit:apphost
```

دسترسی به تنظیمات این قسمت توسط دات نت:

```
using System;
using System.Text;
using Microsoft.Web.Administration;

internal static class Sample
{
    private static void Main()
    {
        using (ServerManager serverManager = new ServerManager())
        {
            Configuration config = serverManager.GetApplicationHostConfiguration();

            ConfigurationSection logSection = config.GetSection("system.ftpServer/log");
            logSection["centralLogFileMode"] = @"Central";

            ConfigurationElement centralLogFileElement = logSection.GetChildElement("centralLogFile");
            centralLogFileElement["enabled"] = true;

            serverManager.CommitChanges();
        }
    }
}
```

دسترسی به تنظیمات این قسمت توسط Javascript:

```
var adminManager = new ActiveXObject('Microsoft.ApplicationHost.WritableAdminManager');
adminManager.CommitPath = "MACHINE/WEBROOT/APPHOST";

var logSection = adminManager.GetAdminSection("system.ftpServer/log", "MACHINE/WEBROOT/APPHOST");
logSection.Properties.Item("centralLogFileMode").Value = "Central";

var centralLogFileElement = logSection.ChildElements.Item("centralLogFile");
centralLogFileElement.Properties.Item("enabled").Value = true;

adminManager.CommitChanges();
```

سشن‌ها در برنامه‌های وب، یکی از وابستگی‌های استاتیکی هستند که می‌توان آن‌ها را از طریق تزریق وابستگی‌ها، جهت بالا بردن قابلیت آزمون پذیری برنامه، تامین کرد. همچنین اگر از سشن‌ها برای نمونه در برنامه‌های ASP.NET MVC استفاده کنید، مقدار آن‌ها در سازنده‌ی کنترلرها نال خواهد بود؛ از این جهت که در زمان نمونه سازی یک کنترلر توسط IoC Container، کار مدیریت سشن‌ها صورت نمی‌گیرد و اگر در این بین سرویسی نیاز به سشن داشته باشد، دیگر وله سازی نخواهد شد؛ به این دلیل که صرفنظر از مقدار دهی متغیر سشن در صفحه‌ای دیگر، این مقدار در سازنده‌ی کلاس، نال است. در ادامه این مشکل را از طریق غنی سازی تزریق وابستگی‌ها با اطلاعات سشن جاری، برطرف خواهیم کرد.

طراحی یک تامین کننده‌ی عمومی سشن

```
public interface IServiceProvider
{
    object Get(string key);
    T Get<T>(string key) where T : class;
    void Remove(string key);
    void RemoveAll();
    void Store(string key, object value);
}
```

در اینجا یک اینترفیس عمومی را مشاهده می‌کنید که کار آن کپسوله سازی اعمال متداول کار با سشن‌ها است؛ برای مثال دریافت اطلاعات یک سشن، بر اساس کلیدی مشخص و یا ذخیره سازی اطلاعات اشیاء در سشن‌ها. یک نمونه پیاده سازی عمومی آن نیز برای کار با سشن‌ها در برنامه‌های وب ASP.NET و فرم و MVC، می‌تواند به صورت زیر باشد:

```
public class DefaultWebSessionProvider : IServiceProvider
{
    private readonly HttpSessionStateBase _session;

    public DefaultWebSessionProvider(HttpSessionStateBase session)
    {
        _session = session;
    }

    public object Get(string key)
    {
        return _session[key];
    }

    public T Get<T>(string key) where T : class
    {
        return _session[key] as T;
    }

    public void Remove(string key)
    {
        _session.Remove(key);
    }

    public void RemoveAll()
    {
        _session.RemoveAll();
    }

    public void Store(string key, object value)
    {
        _session[key] = value;
    }
}
```

در کلاس DefaultWebSessionProvider مستقیماً از HttpContext.Current.Session برای دسترسی به سشن جاری استفاده نشده است. این مقدار را از سازنده‌ی خود که توسط کلاس پایه HttpSessionStateBase تأمین می‌شود، دریافت خواهد کرد. این سازنده را توسط تنظیمات ابتدایی IoC Container خود و هله سازی و مقدار دهی می‌کنیم؛ زیرا HttpContext.Current.Session برای مقدار دهی، نیاز به راه اندازی یک وب سرور دارد و عملاً استفاده و شبیه سازی از آن در بسیاری از آزمون‌های واحد، بسیار مشکل خواهد بود.

```
private static Container defaultContainer()
{
    return new Container(ioc =>
    {
        // session manager setup
        ioc.For<ISessionProvider>().Use<DefaultWebSessionProvider>();
        ioc.For<HttpSessionStateBase>()
            .Use(ctx => new HttpSessionStateWrapper(HttpContext.Current.Session));

        ioc.Policies.SetAllProperties(properties =>
        {
            properties.OfType<ISessionProvider>();
        });
    });
}
```

در مثال فوق یک نمونه از تنظیمات ابتدایی HttpContext.Current.Session را برای استفاده از مقدار StructureMap مشاهده می‌کنید. سازنده‌ی کلاس DefaultWebSessionProvider جهت و هله سازی سشن در برنامه است.

استفاده از تامین کننده‌ی سفارشی سشن در برنامه

پس از طراحی تامین کننده‌ی سفارشی سشن و همچنین معرفی آن به IoC Container خود، اکنون استفاده‌ی از آن به سادگی ذیل است:

```
public class HomeController : Controller
{
    private readonly ISessionProvider _sessionProvider;
    public HomeController(ISessionProvider sessionProvider)
    {
        _sessionProvider = sessionProvider;
    }
}
```

بنابراین اگر در کلاسی، کنترلری و یا سرویسی نیاز به سشن وجود داشت، بهتر است از ISessionProvider بجای مقدار دهی و یا دسترسی مستقیم به شیء استاتیک Session استفاده کرد.

نظرات خوانندگان

نویسنده: **وحید نصیری**
تاریخ: **۱۵:۲۵ ۱۳۹۳/۱۲/۱۴**

یک نکته‌ی تکمیلی

شبیه به سشن، چند مورد دیگر را نیز می‌توان به همین روش تامین کرد:

```
ioc.For<IIdentity>().Use(() => HttpContext.Current.User.Identity);
ioc.For<HttpContextBase>().Use(() => new HttpContextWrapper(HttpContext.Current));
ioc.For<HttpServerUtilityBase>().Use(() => new HttpServerUtilityWrapper(HttpContext.Current.Server));
ioc.For<HttpRequestBase>().Use(ctx => ctx.GetInstance<HttpContextBase>().Request);
```

عنوان: پیاده سازی یک متد الحاقی برای تبدیل آدرس فیزیکی به آدرس مجازی (آدرس سرور)
 نویسنده: احمد نواصری
 تاریخ: ۱۳۹۴/۰۲/۲۹
 آدرس: www.dotnettips.info
 گروهها: ASP.Net, MVC

سناربوی زیر را در نظر بگیرید:

در حال تهیه‌ی یک CMS هستید و طبق سفارش مشتری قسمتی را برای نمایشگاه محصولات در نظر گرفته‌اید. مشتری در نظر دارد در وب سایت خود، محصولات عرضه شده را به صورت یک گالری نمایش دهد و برای اینکار شما از یک فایل آپلودر مثل [Kendo](#) استفاده کرده‌اید. در این حالت برای ذخیره‌ی فایلها بر روی دیسک، از متد [MapPath](#) به صورت زیر استفاده می‌کنید:

```
var physicalPath = Path.Combine(Server.MapPath("~/Content/Images"), fileName);
```

خروجی متد بالا چیزی شبیه زیر است :

```
C:\\YourProject\\Content\\Images\\1.jpg
```

اما برای نمایش عکس‌ها باید بتوانیم مسیر عکس‌های ذخیره شده‌ی در فایل فیزیکی را به آدرس سرور، یا همان آدرس مجازی تبدیل کنیم. برای اینکار می‌توان از یک متد الحاقی به صورت زیر استفاده نمود :

```
public static class PathConverter
{
  public static string PhysicalToVirtualPathConverter(this HttpServerUtilityBase utility, string path, HttpRequestBase context)
  {
    return path.Replace(context.ServerVariables["APPL_PHYSICAL_PATH"], "/").Replace(@"\", "/");
  }
}
```

و برای استفاده از متد بالا به صورت زیر عمل می‌کنیم :

```
image.ImagePath = Server.PhysicalToVirtualPathConverter(PhysicalPath, Request);
```

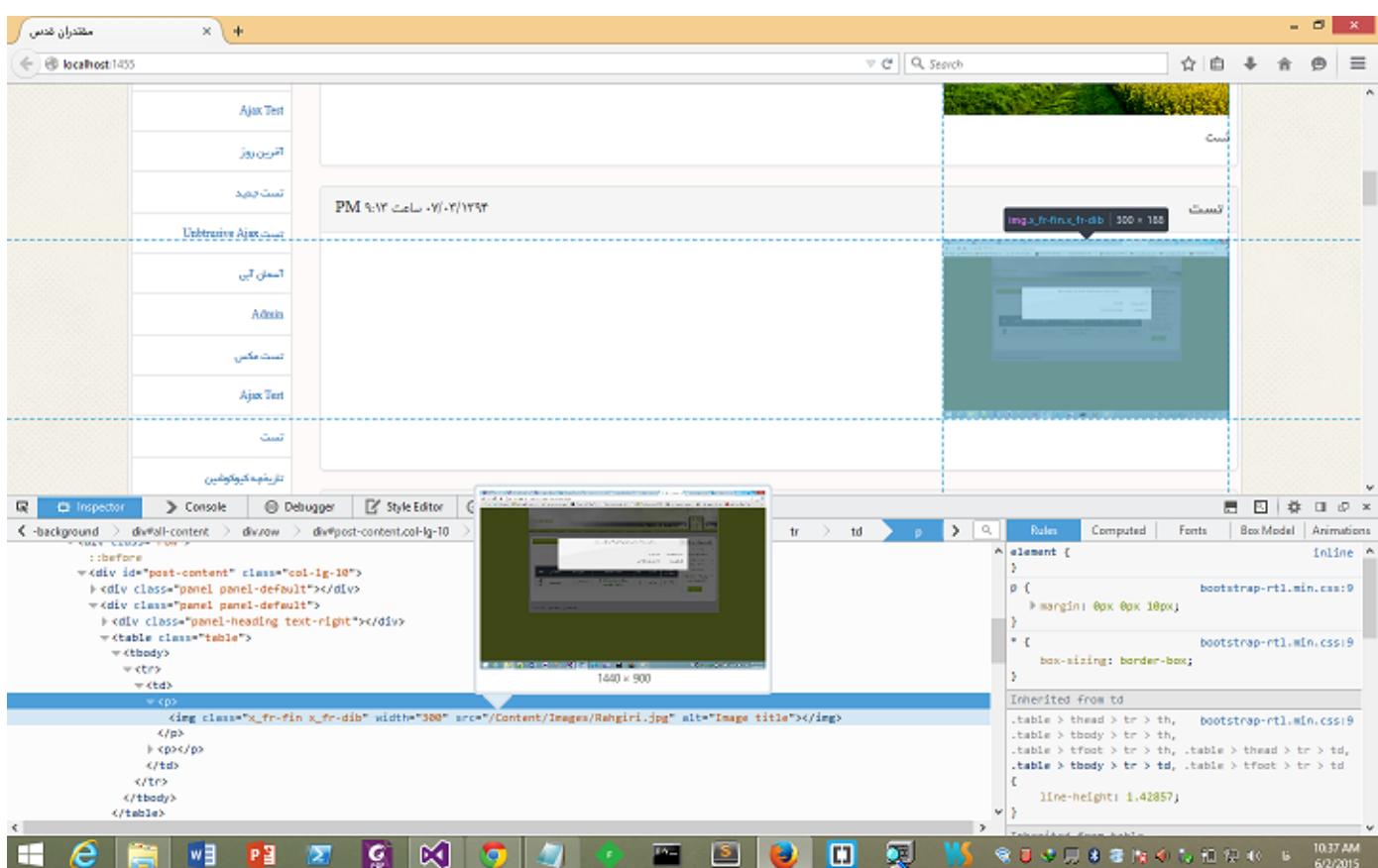
نظرات خوانندگان

نویسنده: معصومه فیضیان
تاریخ: ۷:۵۱ ۱۳۹۴/۰۳/۱۲

بسیار عالی. می خواستم بدانم در صورت استفاده از این روش وقتی روی عکس inspect میگیریم مسیر واقعی را نشان نمی دهد و مسیر مجازی که ساخته ایم نشان داده میشود؟ در این صورت یک روش امنیتی ایجاد شده

نویسنده: احمد نواصری
تاریخ: ۱۰:۴۷ ۱۳۹۴/۰۳/۱۲

بله مسیر مجازی رو نشون میده. نتیجه : Inspect



برای بسیاری از تازه کاران که پا به عرصه‌ی برنامه‌های تحت وب می‌گذارند، اینکه چگونه، از کجا و چطور باید هاستی را انتخاب کنند، دچار سردرگمی هستند. دیدن پلن‌های مختلف با قیمت‌های مختلف، باعث افزایش سردرگمی آن‌ها می‌شود. در این مقاله به بررسی اینکه چطور باید هاستی خریداری شود و اینکه اصلاً خود برنامه‌ی نوشته شده نیازش چقدر هست، صحبت می‌کنیم.

قبل از اینکه صحبت را آغاز کنیم باید این نکته اشاره کنیم که انواع هاست چیست؟

انواع هاست

هاست‌ها بر سه نوع تقسیم می‌شوند:

هاست اشتراکی

سرورهای مجازی

سرور اختصاصی

هاست اشتراکی

در این حالت شرکت میزبان یک سرور را به چند سایت تقسیم کرده و به هر وب سایت، با توجه به پلنی که مشتری انتخاب کرده، مقداری از منابع را اختصاص می‌دهد که عموماً این تخصیص منابع در زمان خرید توسط [WHMCS](#) به طور خودکار صورت می‌گیرد. در این حالت ممکن است بر روی یک سرور بیش از یکصد وب سایت در حال سرویس گرفتن باشند. مزیت این هاست‌ها قیمت ارزان آن‌ها می‌باشد. عموماً وب سایتها با بازدید کم و نیاز به منابع کمتر، از این دست هاست‌ها استفاده می‌کنند. در این نوع هاست‌ها در صورتیکه استفاده از منابع به نهایت مقداری که برای آن مشخص شده است برسد، از قبیل ترافیک (که بیشتر مسئله مربوط به آن است) یا دیسک سخت و ... به حد مشخص شده برسند، سایت را متوقف یا suspend می‌کنند. پرداخت این نوع هاست‌ها به خاطر قیمت پایین به صورت یکسانه دریافت می‌شود. قیمت سالیانه آن‌ها در بعضی جاها از 50 هزار تومان تا صدهزار تومان به عنوان مبلغ آغازین شروع می‌شود.

سرورهای مجازی Virtual Private Server یا VPS

این‌ها هم تقریباً مثل هاست‌های اشتراکی هستند با این تفاوت که منابع بیشتر و دسترسی بیشتری به شما داده می‌شوند؛ به طوری که احساس می‌کنید به شما یک سرور واقعی را داده‌اند و عموماً تعداد سایت‌هایی که روی آن سرویس می‌گیرند، به مراتب کمتر از هاست اشتراکی است. اکثر وب سایتها ای که توانایی استفاده از هاست‌های اشتراکی را ندارند، از این نوع هاستینگ بهره می‌برند. قیمتیش بالاتر از یک هاست اشتراکی است ولی به مراتب پایین‌تر از یک سرور اختصاصی است. پرداخت این نوع هاست عموماً به دو صورت ماهیانه و یا سالانه است که احتمال زیادی دارد پرداخت سالیانه تخفیف خوبی را شامل شود. در صورت رسیدن به حد نهایت منابع همانند هاست اشتراکی با شما رفتار خواهد شد. این سرورهای مجازی در ایران عموماً از مبلغ ماهیانه 20 هزار تومان به بالا آغاز می‌شوند.

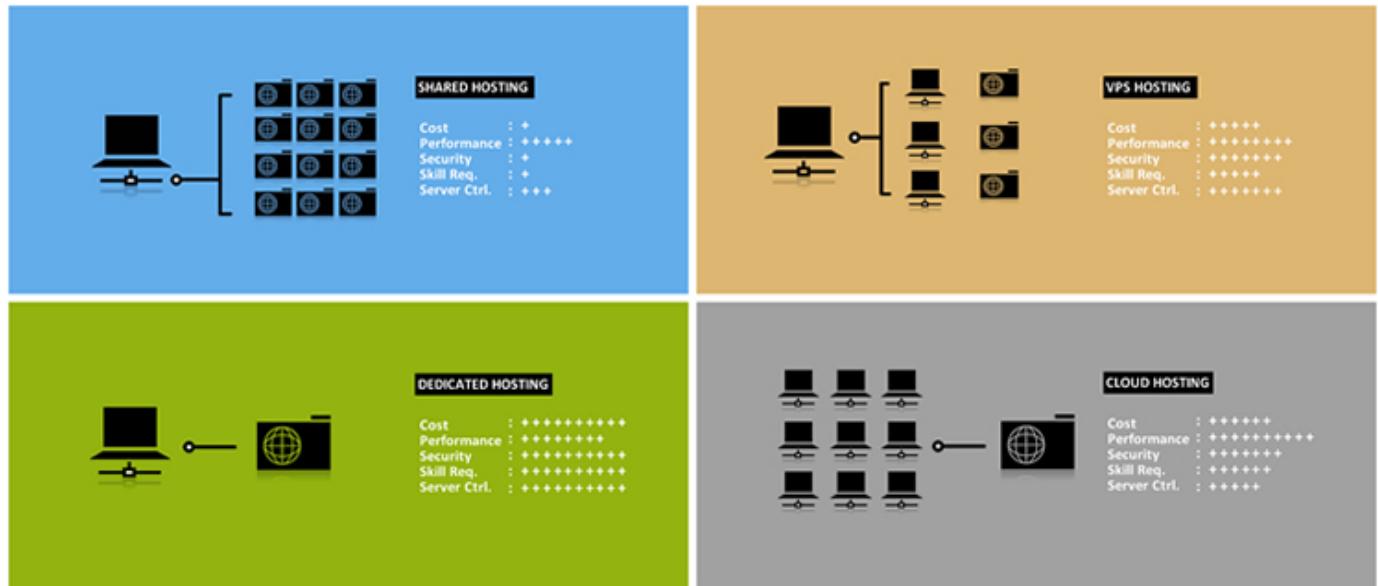
سروهای اختصاصی یا Dedicated

این‌ها دیگر سروهای واقعی هستند و صاحب اول و آخرشان شمایید و دسترسی کامل به همه اجزا و منابع آن را دارید. این سرورهای عموماً برای فعالیت‌های بزرگ تجاری و بازدیدهای همزمان به شدت بالا استفاده می‌شوند. قیمت، بسته به مشخصات آن متفاوت است و ممکن است یکی ماهیانه 200 هزار تومان، یکی ماهیانه 500 هزار تومان و .. آغاز شود.

نکته‌ای در مورد سروهای اختصاصی و حتی گاهها هاست اینکه عموماً پشتیبانی این‌ها توسط شما تامین می‌شود و یا باید یک مدیر سرور با حق ماهیانه اختیار کنید یا در صورت بروز مشکل به صورت ساعتی حق حل مشکل را بدھید. بهتر هست این مورد را از قبل توسط میزبان جویا شوید.

سرورهای ابری

اگر قصد انجام عملیات [رایانش برای](#) را دارید بهتر هست که دنبال چنین [سرورهای](#) باشید که مورد بحث این مقاله نیست و صرفاً جهت تکمیل معرفی انواع هاستها آمده است.



چک لیست

موقعی که شما نوع هاست خود را انتخاب می‌کنید به یک سری پلن یا پکیج‌های مختلفی می‌رسید که مشخصات مختلفی دارند و هر شرکت مبلغ و پیکربندی مختلفی را در نظر می‌گیرد. نگاهی به چک لیست پایین و بررسی گام به گام آن می‌تواند شما را در رسیدن به انتخاب بهتر یاری کند.

فضای دیسک و پهنای باند (ترافیک)

اولین نکاتی که همیشه توسط میزبان‌ها و خریداران مورد توجه قرار می‌گیرند، این دو مورد هست. در صورتیکه سایت شما اجازه آپلودی به کاربران نمی‌دهد، یا خودتان هم آپلود چندانی ندارید، می‌توانید فضای دیسک سخت پایین‌تری را انتخاب کنید. مثلاً اگر شما تعدادی تصویر دارید و مقدار زیادی فایل متند... به نظر یک گیگ کفایت می‌کند و در صورتیکه بعدها دیدید این فضا رو به اتمام است، می‌توانید به میزبان درخواست افزایش و ارتقاء آن را بدهید؛ اما برای بار اول یک گیگ به خوبی کفایت می‌کند. ولی اگر چنانچه با فایل‌های حجمی سرو کله میزند و یا تعداد فایل‌هایی چون تصاویر، صوت و ویدیو در آن زیاد دیده می‌شود، بهتر است به فکر فضایی بیشتر و حتی گاهها **unlimited** باشید. برای سایت‌هایی که حجم به شدت بالایی دارند و یا مثلثاً نیاز به راه اندازی بخش دانلود دارند، بهتر هست از یک هاستینگ به نام هاست دانلود استفاده کنند. این نوع هاست‌ها به شما اجازه میزبانی فایل‌های وب سایت‌هایی چون **.aspx, .mvc, .php** را نمی‌دهند و بیشتر پهنای باند و فضای دیسک سخت آن مدنظر است. در این حالت سایت خود را روی یک هاست معمولی به اشتراک می‌گذارید و فایل‌های خود را روی هاست دانلود قرار می‌دهید و ارتباط آن‌ها را لینک می‌کنید.

در مورد پهنای باند باید تعداد کاربر و همچنین نوع اطلاعاتی که جابجا می‌شوند را مورد بررسی قرار دهید. اگر تعداد کاربران پایین است عموماً این مقدار کم است و یا اگر اطلاعات متند فقط جابجا می‌شود این مقدار کمتر هم می‌شود.

در سناریوی اول یک انجمن **VB** را بررسی می‌کنیم: عموم انجمن‌ها در زمینه‌ی چند رسانه‌ای مثل تصاویر و ...، چیزی جز تصاویر

پوسته خود (که کش هم می‌شوند) را انتقال نمی‌دهند. به این دلیل که اجازه‌های آپلود را به کار نمی‌دهند و کاربرها بیشتر از سرویس‌های ثالث برای تصاویر بهره می‌برند و به غیر از پوسته‌ی، سایت متون انجمن هستند که متن هم ترافیک پایینی را مصرف می‌کند. پس در این حالت ترافیک ماهیانه‌ی 10 گیگ میتواند برای شروع کار تا مدتی که سایت بازدیدکنندگان خودش را پیدا کند، مناسب باشد. از نظر دیسک سخت هم به نظر من اگر شما نیاز به قرار دادن تصاویری چون اسلایدشوها و ... را دارید، به انضمام آواتار کاربران، 500 مگابایت می‌تواند کافی باشد. حجم آواتار کاربران در قسمت مدیریت، کنترل شده است و تنظیم دستی آن می‌تواند این مقدار حجم آواتارها را افزایش دهد.

در سناریوی دوم ما یک وب سرویس مثلاً برای یک برنامه‌ی اندرویدی را مثال می‌زنیم: عموماً وب سرویس‌ها چیزی جز یک فایل متنی را انتقال نمی‌دهند؛ مگر اینکه از تصاویر، صورت و ویدیو هم بهره بردۀ باشید. در صورتیکه بیشتر همان متن را بخواهید انتقال دهید، ترافیکی جز همان متن مصرف نمی‌شود و حتی از یک انجمن هم مصرف پایین‌تری دارد و می‌تواند ماهیانه 10 گیگ یا حتی کمتر هم مورد استفاده قرار بگیرد. در صورت اضافه شدن تصویر و دیگر موارد چندرسانه‌ای و کاربران در حال افزایش، این مقدار هم به همان تناسب بالا می‌رود.

سرعت Speed و توان سرویس دهنده (Uptime)

بعد از ارزیابی موارد بالا نوبت به این دو مورد می‌رسد. اینکه هاست شما به قولی چقدر دوام و ایستادگی دارد، بسیار مهم است و در صورتیکه این امر محقق نگردد، می‌تواند موجب از دست دادن و شنیدن اعتراض کاربرها باشد. درست شنیدید! حتماً پیش خودتان می‌گویید خوب، یک وب سایت همیشه در دسترس است. ولی واقعیت را بخواهید خیر، اینگونه نیست. اگر سرویس دهنده‌ی خوبی را انتخاب نکنید، ممکن است در روز یا هفته یا در هر مقطع زمانی، با در دسترس نبودن وب سایت روبرو شوید؛ تقریباً مشابه زمانیکه ads1 شما قطع می‌شود، چند لحظه‌ای (طولانی‌تر از زمان عادی) مرورگر شما سعی کرده و می‌بیند که زورش نمی‌رسد و یک پیام عدم دسترسی را به شما نمایش می‌دهد.

به خوبی به یاد دارم که یک میزبان، هاست‌های ارزان قیمتی را ارائه می‌کرد، ولی گاهها در روز دو الی سه بار پنج دقیقه‌ای سرور از دسترس خارج می‌شد و می‌گفتیم این هاست برای کسانی است که پول کافی ندارند و یا خیلی نمی‌خواهند خرج کنند و حالا پنج دقیقه‌ای هم در روز در دسترس نباشد اهمیتی ندارد. در سایت‌ها بیشتر این مورد را با اعدادی شبیه 99.9% مشخص می‌کنند و پیشنهاد می‌کنم هاستی را بخرید که این عدد را به شما نشان می‌دهد، یا حداقل دیگر 99.5% کمتر نباشد. البته تمامی هاست‌ها همین را می‌نویسند، ولی واقعیت چیز دیگری است و بهتر از از دوستان و آشنایان و جستجوی در اینترنت و [انجمنها](#) به این مورد بررسید.

فاکتور بعدی سرعت سرور است و کاربران به این مورد هم اهمیت ویژه‌ای می‌دهند. به خوبی به یاد دارم، اولین هاستی که در کار استفاده کردم و به پیشنهاد دوستم که در آن کار می‌کرد این هاست را خریداری کردم، اوایل خوب بود، ولی مشکل سرعت بعدها اضافه شد که حتی ورود به پنل مدیریتی چند دقیقه‌ای طول می‌کشید. یا اینکه چند وقت پیش وب سایتی را مطالعه می‌کردم که فقط متن جابجا میکرد، ولی به طوری که کندرتر از زمان اجرای یک سایت با گرافیک متوسط طول می‌کشید. (البته این نکته حائز اهمیت است خود وب سایت هم باید از این لحاظ مورد بررسی قرار گیرد و مشکل را سریعاً گردن سرویس دهنده نیندازیم).

برای اینکه بدانیم که سرعت یک هاست چگونه باید ارزیابی شود باید سه مورد را بررسی کنیم:

دیتابستر

سرور

نzdیکی سرور به محل زندگی کاربران هدف

دو مورد اول تا حدی فنی هستند و اصلاً ممکن هست به چنین اطلاعاتی دسترسی هم نداشته باشید؛ هر چند میزبان - در صورتی که قصد خرید سروری را دارید - در صورت پرسش شما باید این اطلاعات را در اختیار شما بگذارد. عموماً چیزی که پیشنهاد می‌شود دیتابسترها [SAS70 Type 2](#) هستند و سرورهای DELL، امروزه خوب مورد استقبال قرار گرفته‌اند. در رده‌های بعدی میتوان HP را هم مثال زد.

مورد سوم نzdیکی سرور به کاربران هدف است. هر چقدر traceroute و پینگ زمانی کمتری به سرور داشته باشید، دسترسی به اطلاعات آن سریعتر می‌شود. در ایران عموماً از کشورهایی چون آمریکا، کانادا، انگلیس، آلمان و استرالیا سرور خریده می‌شود و

به مدت چندسالی است که در ایران هم این امکان فراهم شده است ولی خب مسائلی که این سرورها در ایران دارند باعث می‌شوند عده‌ی زیادی دور آن‌ها را که هیچ روی آن‌ها را هم خط بکشند.

مشکلاتی که این سرورها دارند بیشتر به سه مورد زیر بر می‌گردد:

هزینه‌ی سنگین ترافیک : این مورد آن قدر به وضوح پیداست که شما را به کل برای هر نوع خریدی پشمیمان می‌کند؛ مگر اینکه پولتان از جایی تامین نمی‌شود. ادارات دولتی عموماً این نوع هاست را بر میدارند و یا سایت‌های اشتراکی که منابع زیادی را مصرف نمی‌کنند و یا شرکت‌ها یا اشخاصی که پول تامین را دارند.

عدم دسترسی به شبکه‌های اجتماعی یا هرسرویس دهنده‌ی مسدود شده : من این را تست نکردم ولی با دو دوتا چهارتا کردن این حساب دستم آمد که وقتی سایتی مثل فیس بوک و تؤیتیر در ایران مسدود باشند، باید روی این سرورها هم مسدود باشند. در نتیجه اگر سایت شما مطالبش را در این سایتها معرفی می‌کند و می‌خواهید پست و تؤیتی روی آن‌ها داشته باشید، احتمالاً توانایی این کار را نخواهید داشت. مگه اینکه خودتان دستی بروید در سایت مربوطه و اضافه کنید.

جایگاه پایین‌تر SEO : گفتیم که یکی از عوامل سرعت، نزدیک بودن سرور به کاربر هدف است. این مورد برای سرور موتورهای جستجویی مثل گوگل که در ایران سروری ندارند هم اتفاق می‌افتد و در نتیجه گوگل از لحاظ امتیاز بندی سرعت، امتیاز شما را کم می‌کند ولی این مورد همیشه چندان صدق نمی‌کند و مبحث ترافیک سایت بیشتر مورد ارزیابی قرار می‌گیرد.

امنیت

این مورد شامل دو بخش می‌شود یکی امنیت دسترسی به سرور و دیگر امنیت نگهداری اطلاعات. نصب فایروال‌ها روی سرور و نظارت 24 ساعته روی سرورهای شرکت (که شامل بخش پشتیبانی می‌شود) می‌تواند این موردها را پوشش دهد. هر چند این مورد را زیاد نمی‌توانید محک بزنید، ولی اگر اخباری چون «همکاری با یک شرکت امنیتی جهت تست سرور و همکاری با تعدادی هکر جهت تست سرور» و ... را شنیدید، می‌توانید این اطمینان را کسب کنید که آن‌ها به این مورد اهمیت میدهند.

امنیت اطلاعات چون پشتیبان‌های روزانه و نگهداری اطلاعات تا مدتی معین پس از اتمام قرارداد و موارد این چنینی، می‌تواند شما را مطمئن سازد. در مورد یکی از مشتریانم به خاطر دارم که فراموش کرده بود هاست را تمدید کند و یک روز بعد از انقضاء ما درخواست دیتابیس را داشتیم که برای ما ارسال کنند و به گفته‌ی خودشان ما مسئولیتی در قبال نگه داری اطلاعات نداریم، ولی تا 5 روز نگه میداریم که البته گفتند هر چه به دنبال دیتابیس گشتند، چیزی پیدا نکردند که البته این مشکل را از راهکار دیگری حل کردیم و ماجرا به خوشی تمام شد. ولی جالب این بود که اینقدر که من حرص خوردم، چطوری به مدیر این‌ها بگوئیم که اطلاعات تیمیان پریده و این‌ها حرص نخورند و بی خیال.

کنترل پنل

می‌گویند کنترل پنل هم چیز مهمی است ولی در اکثر اوقات اکثر هاستینگ‌ها از همان پنل‌های معروف استفاده می‌کنند. برای مثال در لینوکس "سی پنل" و در ویندوز "وب سایت پنل" و "پلسک" می‌باشد . تا آخرین باری که من با آن‌ها کار کردم، وب سایت پنل، یک پنل سبک بود که برای انجام عملیات ساده تا متوسط در زمینه کاری پنل‌ها می‌باشد و پلسک نسبت به آن امکانات خیلی زیادتری دارد. اکثراً آنها حاوی امکانی چون نصب یک کلیکی CMS‌هایی چون وردپرس و جوملا و ... هستند.

اگر از یک سرور اختصاصی بهره ببرید خودتان مختار نصب هر نوع چیزی روی آن هستید و در مورد کنترل پنل هم صدق می‌کند ولی عموماً شرکتها یک سری الگوهای پیش فرضی را برای سرویس‌های خود دارند که در صورت تغییر باید به آن‌ها، تغییرات را طلاع دهید.

پشتیبانی فنی

موردی را تصور کنید که ساعت 2 شب هست و سایت شما هم که در زمینه‌ی مهم و حساسی فعالیت می‌کند، یک دفعه سرویس آن قطع می‌شود و برای سرور مشکلی پیش می‌آید. در این موقع چکاری را انجام می‌دهید؟ صبر می‌کنید تا صبح شود و سرویس شما راه بیفت و کاربران هم از سایت شما نا امید شوند؟ اگر این مشکل به دفعات رخ بدهد چه؟ اگر چند روز پشت سرهم تعطیلی باشد چه؟

همهی این موارد مربوط به پشتیبانی می‌شود. این پشتیبانی‌ها عموماً به صورت چت و تلفنی (بیشتر مربوط به ساعت‌های اداری) و ارسال تیکت در سامانه پشتیبانی می‌شود. البته تجربه‌ی من میگوید در ایران زیاد به متون نوشته‌ی روی سایت میزبان توجهی نکنید و این را هم به موضوع تحقیق خود اضافه کنید. بعضی‌ها می‌گویند 24 ساعته پشتیبانی تلفنی دارند ولی هر بار زنگ میزني کسی پاسخی نمیدهد. یا تیکت می‌زنید و بارها و بارها پشت سر هم تیکت "چی شد؟" را برای جویا شدن ارسال می‌کنید. (البته انصاف هم داشته باشید و پنج دقیقه پنج دقیقه این تیکت را نفرستید).

پشتیبانی هر نوع سرور را مطلع شوید. بعضی‌ها واقعاً پشتیبانی دارند؟ ولی وقتی زنگ می‌زنید می‌گویند این پشتیبانی مربوط به لینوکس است و بچه‌های ویندوز فقط ساعت‌های اداری هستند. خلاصه حواستان را جمع کنید که بچه‌های آن بخش همیشه باشند. بنابراین مطمئن شوید که یک پشتیبانی 7/24 واقعی داشته باشد.

اماکنات اضافه

در کنار خود هاست یک سری ویژگی‌هایی چون FTP و تعدادی دامنه‌های پشتیبانی شده و زیردامنه‌ها و تعداد دیتابیس‌ها و ایمیل و ... هم هستند که در قدیم یادم هست محدودیت‌هایی در این رابطه ایجاد کرده بودند که امروزه از این نظر در اکثر هاست‌ها آن طور که دیدم این محدودیت‌ها رفع شده است که البته خیلی هم خوب هست و این محدودیت‌ها بیشتر شبیه سودجویی بود تا چیز دیگر.

هزینه

با همهی حرف‌های بالا به نظر میرسد که هزینه، همه این معادلات را به هم می‌ریزد. هاستینگ‌های مختلف و قیمت‌های مختلف، تصمیم گیرنده شما هستید که چقدر به عوامل بالا اهمیت می‌دهید و چگونه آن‌ها را در راستای قیمت اولویت بندی می‌کنید. اگر محدودیتی ندارید سعی کنید همهی عوامل را بررسی کنید. نکته اینکه اکثر هاست‌ها طرحی به نام ضمانت برگشت پول را در هفت روز یا گاهای یک ماه، دارند و در صورتیکه از سرویس خریداری شده راضی نبودید می‌توانید پول خود را پس گرفته و سرویس متعلق شود.

کل مطالب گفته شده در چک لیست به طور خلاصه: اول از همه بررسی فضای ذخیره سازی و پهنانی باند، بعد از آن بررسی توانایی سرویس دهی و سرعت سرورها، داشتن محیط امن و پشتیبانی مناسب بود.

هاستینگ در قرارداد

هنگام عقد قرارداد با مشتری، در قرارداد در مورد هاستینگ که خودش تهیه می‌کند، این نکته ذکر شود که شما در مورد مسائلی که مربوط به هاست و دامنه می‌شود هیچ مسئولیتی ندارید و باید مشکلات را با مسئولان آن در میان بگذارد.

همچنین این نکته هم ذکر شود در مورد مسائلی چون عدم پشتیبانی مناسب هاست، از محصول یا سرویس شما، شما هیچگونه مسئولیتی در قبال آن ندارید و یا باید این مورد توسط شما دنبال شود یا خرید ایشان با مشاوره و تایید شما از هاست مربوطه باشد.

در صورتی که مشتری نمیتواند شخصاً خرید هاستینگ را انجام دهد یا در شرکت مسئولین IT ندارند که این کار را انجام دهند و این مورد به شما محول می‌شود، در قرارداد ذکر شود که شما هیچ گونه مسئولیتی در قبال مشکلاتی که برای هاست و دامنه رخ میدهد ندارید و تنها میتوانید به عنوان یک واسطه یا وکیل در ازای مبلغ توافق شده‌ای مشکل را با مسئولین هاست و دامنه در میان گذاشته و ماجرا را برای حل مشکل ایجاد شده دنبال کنید یا این کار را به عنوان هدیه ای از طرف خدمات طلایی شما به مشتریان به صورت رایگان انجام دهید.