

همان طور که می‌دانید کاربرد پذیری در خیلی از پروژه‌ها حرف اول رو می‌زند و کاربر دوست دارد کارهایی که انجام می‌دهد خیلی راحت و با استفاده از موس باشد. یکی از کارهایی که در اکثر پروژه‌ها نیاز است، چیدمان ترتیب رکوردها است. ما می‌خواهیم در این پست ترتیبی اتخاذ کنیم که کاربر بتواند رکوردها را به هر ترتیبی که دوست دارد نمایش دهد.

ایجاد یک مورد جدید

عنوان	توضیحات	فعال	
مدل کلاسیک	طرح های قدیمی شیرآلات	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
مدل اهرمی	این مدل از طرح های جدید و نوین بهره می برد.	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
سینک شاور	توضیحاتی در مورد سینک های شاور	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
دسته ها و بوسته ها	توضیحاتی در مورد دسته ها و بوسته ها	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
علم های دوش	توضیحاتی در مورد علم های دوش	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
سایر محصولات		<input checked="" type="checkbox"/>	ویرایش مشاهده حذف

از توضیحاتی که قبلاً دادم مشخص است که این کار احتمالاً در ASP.NET WebForm کار سختی نیست ولی این کار باید در MVC از ابتدا طراحی شود.

طرح سوال : یک سری رکورد از یک Table داریم که می‌خواهیم به ترتیب وارد شدن رکوردها نباشد و ترتیبی که ما می‌خواهیم نمایش داده شود.

پاسخ کوتاه : خب باید ابتدا یک فیلد (برای اولویت بندی) به Table اضافه کنیم بعد اون فیلد رو بنا به ترتیبی که دوست داریم رکوردها نمایش داده شود پر کنیم (Sort می‌کنیم) و در آخر هم هنگام نمایش در View رکوردها را بر اساس این فیلد نمایش می‌دهیم.

(این پست هم در ادامه [پست قبلی](#) در همان پروژه است و از همان Table ها استفاده شده است)

اضافه کردن فیلد :

ابتدا یک فیلد به Table مورد نظر اضافه می‌کنیم. من اسم این فیلد رو Priority گذاشتم. Table من چنین وضعیتی دارد.

AMIRHOSSEIN-PC.Khazar - dbo.Types			
	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	Title	nvarchar(200)	<input checked="" type="checkbox"/>
	Description	text	<input checked="" type="checkbox"/>
	ImagePath	varchar(500)	<input checked="" type="checkbox"/>
	Priority	int	<input checked="" type="checkbox"/>
	IsActive	bit	<input type="checkbox"/>
	CreatedOn	datetime	<input type="checkbox"/>
	CreatedBy	uniqueidentifier	<input type="checkbox"/>
	ModifiedOn	datetime	<input checked="" type="checkbox"/>
	ModifiedBy	uniqueidentifier	<input checked="" type="checkbox"/>
	IsDeleted	bit	<input type="checkbox"/>
			<input type="checkbox"/>

افزودن فایل‌های jQuery UI :

در این مرحله شما نیاز دارید فایل‌های مورد نیاز برای Sort کردن رکوردها را اضافه کنید. شما می‌توانید فقط فایل‌های مربوط به Sortable را به صفحه خودتان اضافه کنید و یا مثل من فایل‌هایی که حاوی تمام قسمت‌های jQuery UI هست را اضافه کنید.

من برای این کار از Section استفاده کردم ، ابتدا در Head فایل Layout دو Section تعریف کردم برای CSS و JavaScript . و فایل‌های مربوط به Sort کردن را در صفحه ای که باید عمل Sort انجام بشود در این Section ها قرار دادم.

فایل Layout

```
<head>
<meta charset="utf-8" />
@RenderSection("meta", false)
<title>@ViewBag.Title</title>
<link href="@Url.Content("~/Content/~Site.css")" rel="stylesheet" type="text/css" />
<link href="@Url.Content("~/Content/redactor/css/redactor.css")" rel="stylesheet" type="text/css" />
<link href="@Url.Content("~/Content/css/bootstrap-rtl.min.css")" rel="stylesheet" type="text/css" />
@RenderSection("css", false)
<script src="@Url.Content("~/Scripts/jquery-1.8.2.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/modernizr-1.7.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Content/js/bootstrap-rtl.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Content/redactor/redactor.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.unobtrusive-ajax.min.js")" type="text/javascript"></script>
@RenderSection("js", false)
</head>
```

```
@model IEnumerable<KhazarCo.Models.Type>
@{
    ViewBag.Title = "Index";
    Layout = "~/Areas/Administrator/Views/Shared/_Layout.cshtml";
}
@section css
{<link href="@Url.Content("~/Content/themes/base/jquery-ui.css")" rel="stylesheet" type="text/css" />}
@section js
{
    <script src="@Url.Content("~/Scripts/jquery-ui-1.9.0.min.js")" type="text/javascript"></script>
}
```

در آخر فایل Index.html به اینصورت شده است:

```
<h2>
    نوع ها
</h2>
<p>
    @Html.ActionLink("ایجاد یک مورد جدید", "Create", null, new { @class = "btn btn-info" })
</p>
<table>
    <thead>
        <tr>
            <th>
                عنوان
            </th>
            <th>
                توضیحات
            </th>
            <th>
                فعال
            </th>
            <th>
            </th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model.OrderBy(m => m.Priority))
        {
            <tr id="@item.Id">
                <td>
                    @Html.DisplayFor(modelItem => item.Title)
                </td>
                <td>
                    @(new HtmlString(item.Description))
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.IsActive)
                </td>
                <td>
                    @Html.ActionLink("ویرایش", "Edit", new { id = item.Id }, new { @class = "btnEdit
label label-warning" })
                    @Html.ActionLink("مشاهده", "Details", new { id = item.Id }, new { @class =
"btnDetails label label-info" })
                    @Html.ActionLink("حذف", "Delete", new { id = item.Id }, new { @class = "btnDelete
label label-important" })
                </td>
            </tr>
        }
    </tbody>
</table>
```

** توجه داشته باشید که من به هر tr یک id اختصاص داده ام که این مقدار id همان مقدار فیلد Id همان رکورد هست ، ما برای مرتب کردن به این Id نیاز داریم (خط 25).

افزودن کدهای کلاینت :

حالا باید کدی بنویسیم که دو کار را برای ما انجام دهد : اول حالت Sort پذیری را به سطرهای Table بدهد و دوم اینکه هنگامی که ترتیب سطرهای تغییر کرد ما را با خبر کند:

```
<script type="text/javascript">
    $(function () {
        $("#table tbody").sortable({
            helper: fixHelper,
            update: function (event, ui) {
                jQuery.ajax('@Url.Action("Sort", "Type", new { area = "Administrator" })', {
                    data: { s: $(this).sortable('toArray').toString() }
                });
            }
        }).disableSelection();
    });
    var fixHelper = function (e, ui) {
        ui.children().each(function () {
            $(this).width($(this).width());
        });
        return ui;
    };
</script>
```

توضیح کد :

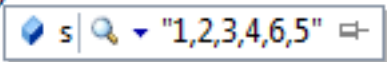
در این کد ما حالت ترتیب پذیری را به Table می دهیم و هنگامی که عمل Update در Table انجام شد تابع مربوطه اجرا می شود. ما در این تابع، ترتیب جدید سطرها را می گیریم (** به کمک مقدار Id که به هر سطر دادیم ، این مقدار Id برابر بود با Id خود رکورد در Database) و به کمک jQuery.ajax به تابع Sort از کنترلر Type در منطقه (Administrator) area ارسال می کنیم و در آنجا ادامه کار را انجام می دهیم.

تابع fixHelper هم به ما کمک می کند که هنگامی که سطرها از جای خود جدا می شوند ، دارای عرض یکسانی باشند و عرض آنها تغییری نکند.

افزودن کد Server :

حالا باید تابع Sort که مقادیر را به آن ارسال کردیم بنویسیم. من این تابع را بر اساس مقداری که از کلاینت ارسال می شود اینگونه طراحی کردم .

```
public EmptyResult Sort(string s)
{
    if (s != null)
    {
```



```
public EmptyResult Sort(string s)
{
    if (s != null)
    {
        var ids = new List<int>();
        foreach (var item in s.Split(','))
```

```

        {
            ids.Add(int.Parse(item));
        }
        int intpriority = 0;
        foreach (var item in ids)
        {
            intpriority++;
            db.Types.Single(m => m.Id == item).Priority = intpriority;
        }
        db.SaveChanges();
    }
    return new EmptyResult();
}
    
```

در ابتدا مقادیر Id که از کلاینت به صورت String ارسال شده است را می‌گیریم و بعد به همان ترتیب ارسال در لیستی از int قرار می‌دهیم ids.

سپس به اضافی هر رکورد Type مقدار اولویت را به فیلدی که برای همین مورد اضافه کردیم Priority اختصاص می‌دهیم. و در آخر هم تغییرات را ذخیره می‌کنیم. (خود کد کاملاً واضح است و نیازی به توضیح بیشتر نیست)

حالا باید هنگامی که لیست Type ها نمایش داده می‌شود به ترتیب (OrderBy) فیلد Priority نمایش داده شود پس تابع Index را اینطور تغییر می‌دهیم.

```

public ActionResult Index()
{
    return View(db.Types.Where(m => m.IsDeleted == false).OrderBy(m => m.Priority));
}
    
```

این هم خروجی کار من:

نوع ها

ایجاد یک مورد جدید

عنوان	توضیحات	فعال	
مدل اهرمی	این مدل از طرح های جدید و نوین بهره می برد.	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
مدل کلاسیک	طرح های قدیمی شیرآلات	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
سینک شاور	توضیحاتی در مورد سینک های شاور	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
دسته ها و بوسته ها	توضیحاتی در مورد دسته ها و بوسته ها	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
علم های دوش	توضیحاتی در مورد علم های دوش	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
سایر محصولات		<input checked="" type="checkbox"/>	ویرایش مشاهده حذف

این عکس مربوط به است به قسمت مدیریت پروژه [شیرآلات مرجان خزر](#) .

نظرات خوانندگان

نویسنده:

سعید

تاریخ:

۲۱:۱۲ ۱۳۹۱/۰۹/۰۸

« این کار احتمالا در ASP.NET WebForm کار سختی نیست »

اتفاقا کار ساده‌ای نیست و همین مراحل باید طی شود. ضمن اینکه کار با jquery ajax در آنجا به این یک دستی نیست. نیاز است در code behind فرم، متد وب سرویس مانندی به صورت استاتیک تعریف شود (که خودش سبب می‌شود تا دسترسی به اعتبار سنجی توکار مبتنی بر فرم‌ها محدود شود) یا اینکه از یک هندلر مجزا بجای یک اکشن متد کمک گرفته شود ... خلاصه خیلی داستان دارد.

نویسنده:

امیرحسین مرجانی

تاریخ:

۲۱:۱۶ ۱۳۹۱/۰۹/۰۸

من خودم این کار رو در webform انجام ندادم و خیلی هم با سختی هاش آشنایی ندارم. ولی فکر کردم شاید با update panel بشه این کار رو راحت انجام داد. خب نظرتون راجب به راه حل من چیه؟

راه حل خوبی ارائه دادم؟ متن قابل فهم بود؟

نویسنده:

سعید

تاریخ:

۲۱:۲۰ ۱۳۹۱/۰۹/۰۸

بسیار عالی.

فقط در مورد update panel ... تنها کاری که انجام می‌دهد کپسوله کردن ارسال مقادیر به سرور است به صورت ajax سازگار با کنترل‌های دارای view state. بیشتر از این کاری انجام نمی‌دهد. مابقی آن اگر یک سری کنترل در toolkit آن موجود باشد برای این کارها یا خیر. این toolkit هم محدود است و آنچنان به روز نمی‌شود.

نویسنده:

امیرحسین مرجانی

تاریخ:

۲۱:۲۳ ۱۳۹۱/۰۹/۰۸

ممنونم از پاسختون

از اینکه سوییچ کردم روی MVC خوشحالم (:

البته یکی از دلایلی هم که این کار رو کردم همین راحتی استفاده Ajax بود.

نویسنده:

امیر آشنا

تاریخ:

۲۲:۴ ۱۳۹۱/۰۹/۰۸

سلام

مطلب مفیدی بود ولی می‌تونستید بیشتر توضیح بدید.

برای ما افراد مبتدی درک بعضی قسمت‌ها کمی مشکل هست.

ولی باز هم ممنونم

نویسنده:

امیرحسین مرجانی

تاریخ: ۲۲:۷ ۱۳۹۱/۰۹/۰۸

سلام

بفرمایید کدوم قسمت‌ها تا بیشتر توضیح بدم.

نویسنده: وحید نصیری
تاریخ: ۲۲:۲۱ ۱۳۹۱/۰۹/۰۸

پیشنیازهای مطلب جاری:

- در مورد RenderSection ([^](#))
- مقدمه‌ای بر jQuery Ajax در MVC ([^](#))
- db.SaveChanges: کل مباحث Entity framework سایت ([^](#))
- EmptyResult و کلا خروجی‌های اکشن متدها: ([^](#))
- علت استفاده از Sort@ Url.Action در حین آدرس دهی: ([^](#))

نویسنده: امیرحسین مرجانی
تاریخ: ۲۲:۲۳ ۱۳۹۱/۰۹/۰۸

واقعا تشکر می‌کنم بابت این لیستی که ارائه کردید.

بحث تکمیل شد

به نظر شما باید بیشتر به جزییات اهمیت داده می‌شد؟

نویسنده: وحید نصیری
تاریخ: ۲۲:۴۱ ۱۳۹۱/۰۹/۰۸

ضرورتی نداره. چون واقعا به اندازه لیستی که عنوان شد نیاز به پیشنیاز درک این مطالب هست و فرصت تکرار آن‌ها نیست. این مطالب جدید، یک سری مطالب تکمیلی هستند نه مطالب پایه و از صفر.

نویسنده: محسن موسوی
تاریخ: ۰:۲۴ ۱۳۹۱/۰۹/۰۹

با استفاده از [Web API Controller](#) تقریبا به همین راحتی امکانپذیره. تغییر زیادی هم نیاز نداره.

نویسنده: امیرحسین مرجانی
تاریخ: ۰:۳۷ ۱۳۹۱/۰۹/۰۹

ممنونم آقای موسوی

قابل توجه آقا سعید (احتمالا این مطلب برای شما مفیده)

نویسنده: سعید
تاریخ: ۰:۵۲ ۱۳۹۱/۰۹/۰۹

- نیاز به vs2012 داره.

- نمیشه یک فرم رو strongly typed تعریف کرد مثل مثال بالا:

```
@model IEnumerable<KhazarCo.Models.Type>
```

نویسنده: محسن موسوی
تاریخ: ۱۰:۲۱ ۱۳۹۱/۰۹/۰۹

بحث در مورد UI نبود. بحث در مورد استفاده کردن بود.
در ضمن در Web Form ها نیز از امکاناتی شبیه به MVC می توان بهره برد. (^)

نویسنده: سعید
تاریخ: ۱۳۹۱/۰۹/۰۹ ۱۰:۴۶

فقط mvc هست که امکان استفاده از چند view engine را با هم دارد.

صفحه ای که لینک دادید مربوط است به [web pages](#) asp.net و نه web forms. این web pages برای کار با webmatrix طراحی شده.

نویسنده: محسن موسوی
تاریخ: ۱۳۹۱/۰۹/۰۹ ۱۱:۴۰

- استفاده از این قابلیت در [Web Page](#) ها توسط VS
 - استفاده از این قابلیت در [Web Form](#) ها توسط VS
- هر دو مورد توسط VS استفاده شده است.

نویسنده: سعید
تاریخ: ۱۳۹۱/۰۹/۰۹ ۱۱:۴۵

```
<iframe src="/twitter" frameborder="0" height="400" width="270" scrolling="no"></iframe>
```

فلسفه و کاربرد web pages متفاوت است. در مورد وب فرم ها هم نمی تونید از razor استفاده کنید چون در یک فایل نمی شود از دو موتور view استفاده کرد. موتور view وب فرم ها، نامش همین view engine web forms است و قابل تعویض هم نیست (برخلاف MVC). در مقاله ای که لینک دادید، داره از یک *iframe* استفاده می کنه برای الحاق razor.

به علاوه در مورد Web API که کلا مطلب جدیدی نیست. نسخه ساده شده WCF است.

نویسنده: محسن موسوی
تاریخ: ۱۳۹۱/۰۹/۰۹ ۱۲:۲۷

دوست عزیز
بحث ما در مورد توانایی انجام موارد فوق الذکر بود. نه اینکه حالا چون ...
به توانایی های MVC شکی نیست. مسئله اصلی اینه که آیا پست جاری را میتوان به راحتی با WebPage و یا WebForm انجام داد؟!
چه از طریق قابلیت های ASP.NET Web Form و ASP.NET Web Page راحتی.
استفاده از Razor چه از طریق JQuery و یا IFrame (نظر قبلی) و یا [به طور مستقل](#) و یا روش های دیگر در ASP.NET Web Form و ASP.NET Web Page
و استفاده معمولی با توانایی های DataBind
یا بطور کامل از [Razor View Engine](#) در Web Page
در نهایت کار سختی نیست.

نویسنده: سعید
تاریخ: ۱۳۹۱/۰۹/۰۹ ۱۲:۴۳

razor بی جهت اینجا طرح شد. من در مورد strongly typed view سطری رو نوشتم. این نوع view ها مستقل از نوع view

engine هستند. در mvc حتی با موتور web forms هم میشه یک چنین view های تحت نظر کامپایلری رو با پشتیبانی کامل از intellisense داشت.

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

تهیه فیلتر سفارشی SiteAuthorize

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

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

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

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

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

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

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

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

        [SiteAuthorize]
        [HttpPost]
        public ActionResult SaveData(string data)
        {
            // ...
        }
    }
}
```

```
{
    if(string.IsNullOrEmpty(data))
        return Content("NOK!");

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

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

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

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

نکته تکمیلی:

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

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

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

نظرات خوانندگان

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

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

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

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

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

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

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

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

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

سلام؛
بسیار مفید و کارآمد بود. بازهم متشکرم

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

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

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

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

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

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

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

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

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

یک نکته‌ی تکمیلی

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

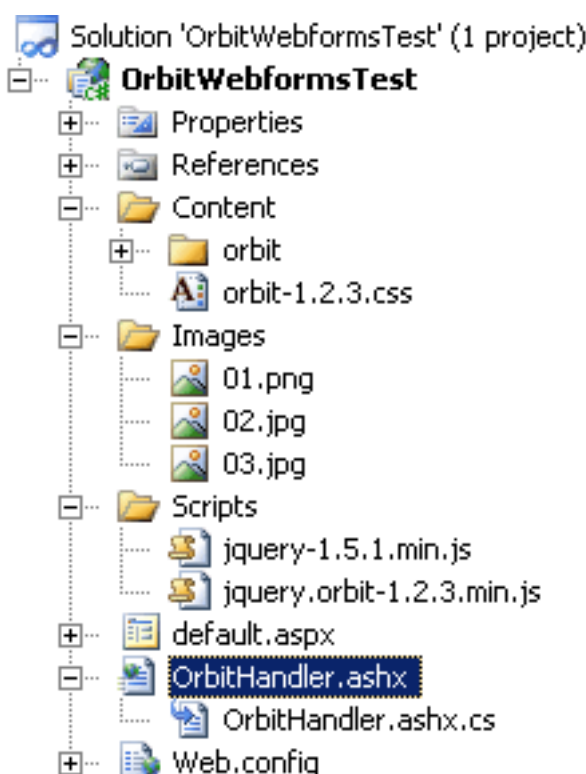
هر از چندی سؤالی «این مثال jQuery رو نمی‌تونم اجرا یا باز سازی کنم» در این سایت یا سایت‌های مشابه تکرار می‌شوند. بنابراین بهتر است نحوه عیب‌یابی برنامه‌های ASP.NET مبتنی بر jQuery را یکبار با هم مرور کنیم. در اینجا، مثال تهیه یک Image Slider را که [پیشتر در سایت مطرح شده](#) است، به نحوی دیگر بررسی خواهیم کرد:

- 1) فراموش می‌کنیم تا اسکریپت اصلی jQuery را به درستی پیوست و مسیردهی کنیم.
- 2) مسیر Generic handler دیگری 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 عمدی 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 نمایید. 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، 5 مورد مشکل ساز یاد شده لحاظ شده‌اند)

محتویات فایل OrbitHandler.ashx.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, "*.jpg"))
            {
                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 پوشه Images واقع در ریشه سایت، قرار دارند. بنابراین توسط متد PicturesDataSource، فایل‌های این پوشه را یافته و مطابق ساختار کلاس Picture بازگشت می‌دهیم. نهایتاً این اطلاعات به ظاهر قرار است با فرمت JSON بازگشت داده شوند تا بتوان نتیجه را توسط افزونه Orbit استفاده کرد.

همچنین کدهای صفحه ASPX ایی 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؟

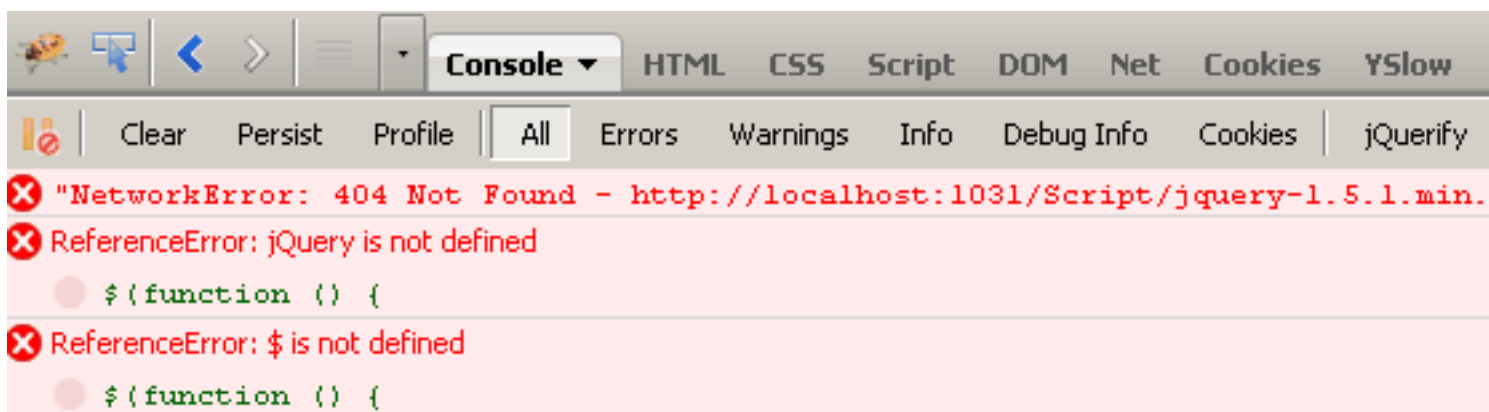
مراحل عیب یابی برنامه‌ای که کار نمی‌کند!

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 خطاها، بر روی برگه Console آن کلیک کنید:

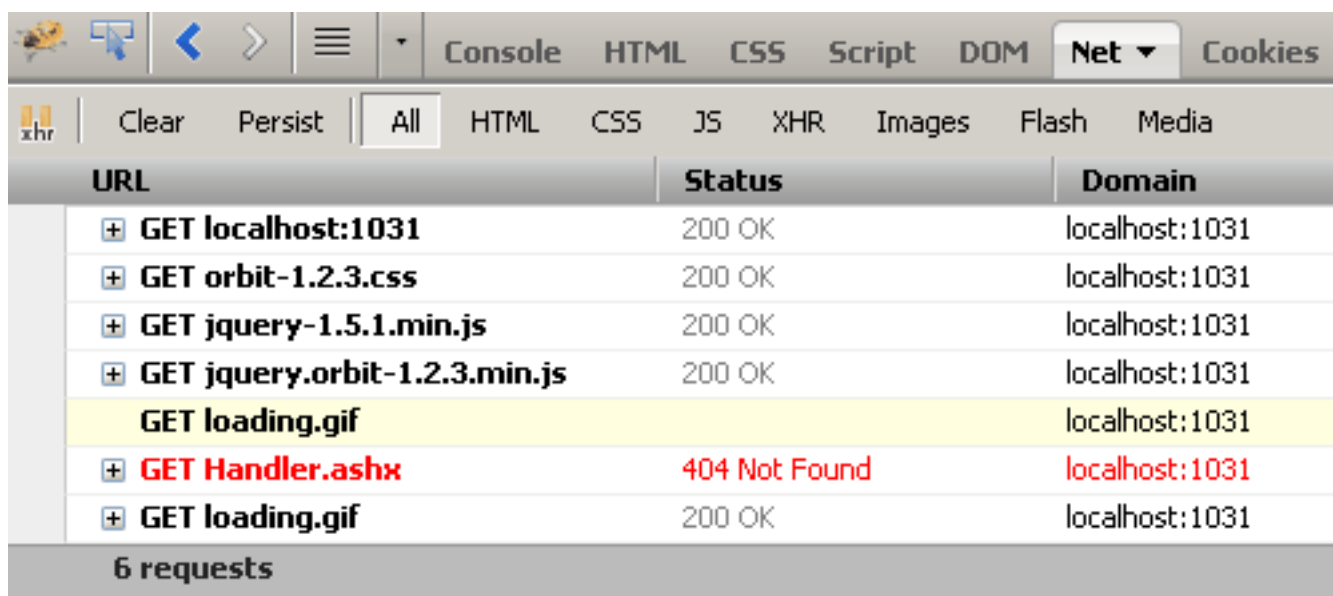


بله. مشخص است که مسیر دهی فایل jquery-1.5.1.min.js صحیح نبوده و همین مساله سبب بروز خطاهای اسکریپتی گردیده است. برای اصلاح آن سطر زیر را در برنامه تغییر دهید:

```
<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.

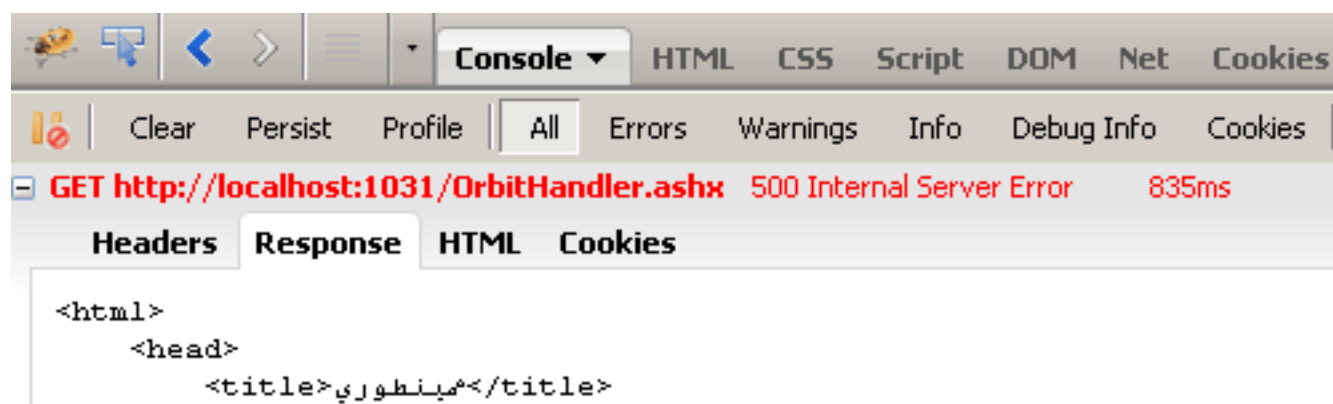
مجددا دکمه F5 را فشرده و سایت را با تنظیمات جدید اجرا کنید. اینبار در برگه Console و یا در برگه شبکه فایرباگ، خطای یافت نشدن Generic handler نمایان می‌شوند:



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.

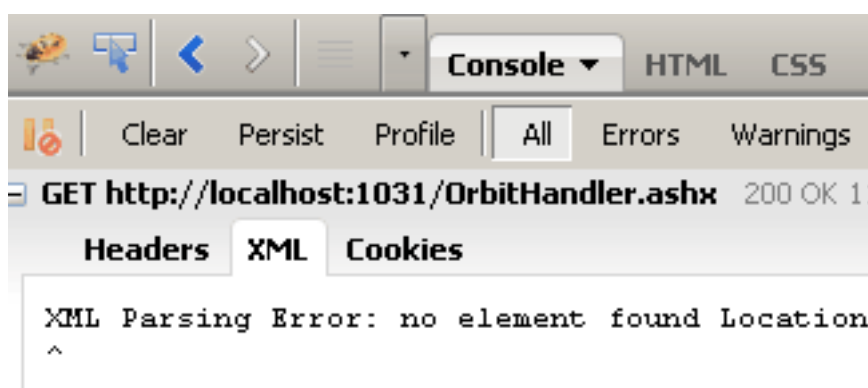


Message	Time
GET http://localhost:1031/OrbitHandler.ashx 500 Internal Server Error	835ms

Headers Response HTML Cookies

```
<html>
  <head>
    <title>مینطوری</title>
```

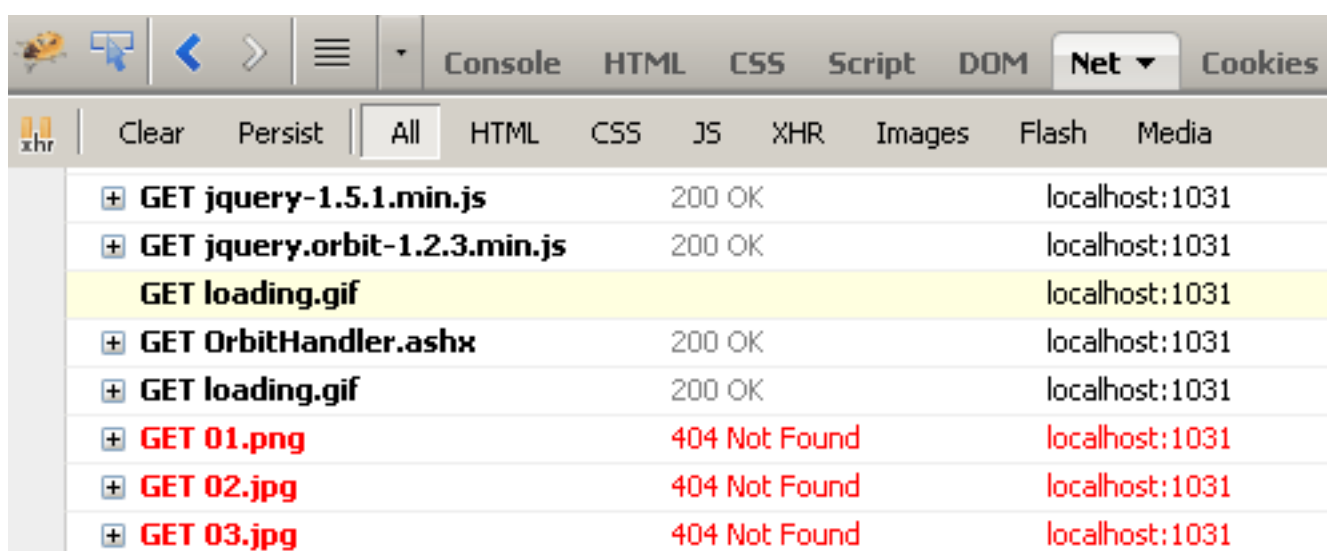
اگر به برگه کنسول دقت کنیم، بروز استثناء در کدها تشخیص داده شده و همچنین در برگه Response پاسخ دریافتی از سرور، جزئیات صفحه خطای بازگشتی از آن نیز قابل بررسی و مشاهده است. اینبار به فایل OrbitHandler.ashx.cs مراجعه کرده و سطر throw new InvalidOperationException را حذف می‌کنیم. در ادامه برنامه را کامپایل و مجددا 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 مطلق 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>();
```

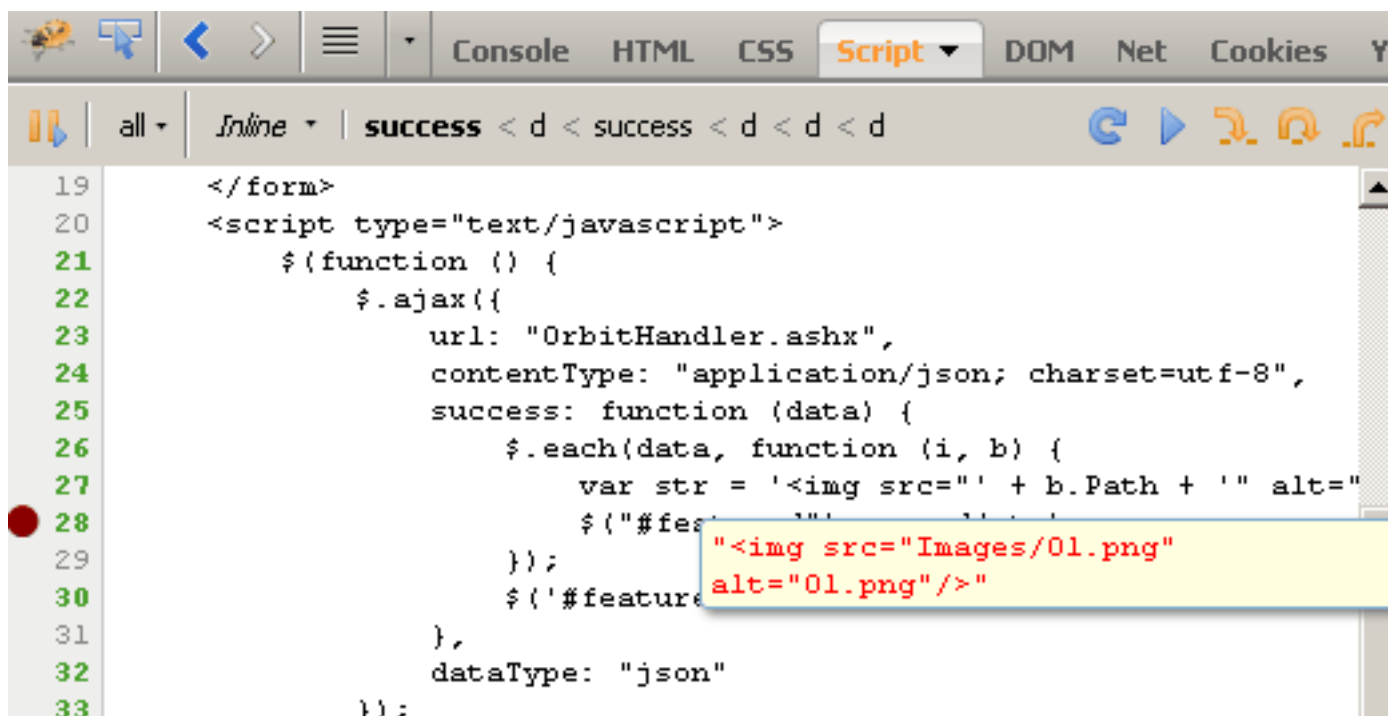
```
var path = HttpContext.Current.Server.MapPath("~/Images");
foreach (var item in Directory.GetFiles(path, "*.png"))
{
    var name = Path.GetFileName(item);
    results.Add(new Picture
    {
        Path = "Images/" + name,
        Title = name
    });
}
return results;
}
```

در این کدها فقط قسمت Path ویرایش شده است تا به مسیر پوشه Images واقع در ریشه سایت اشاره کند. اینبار اگر برنامه را اجرا کنیم، بدون مشکل کار خواهد کرد.

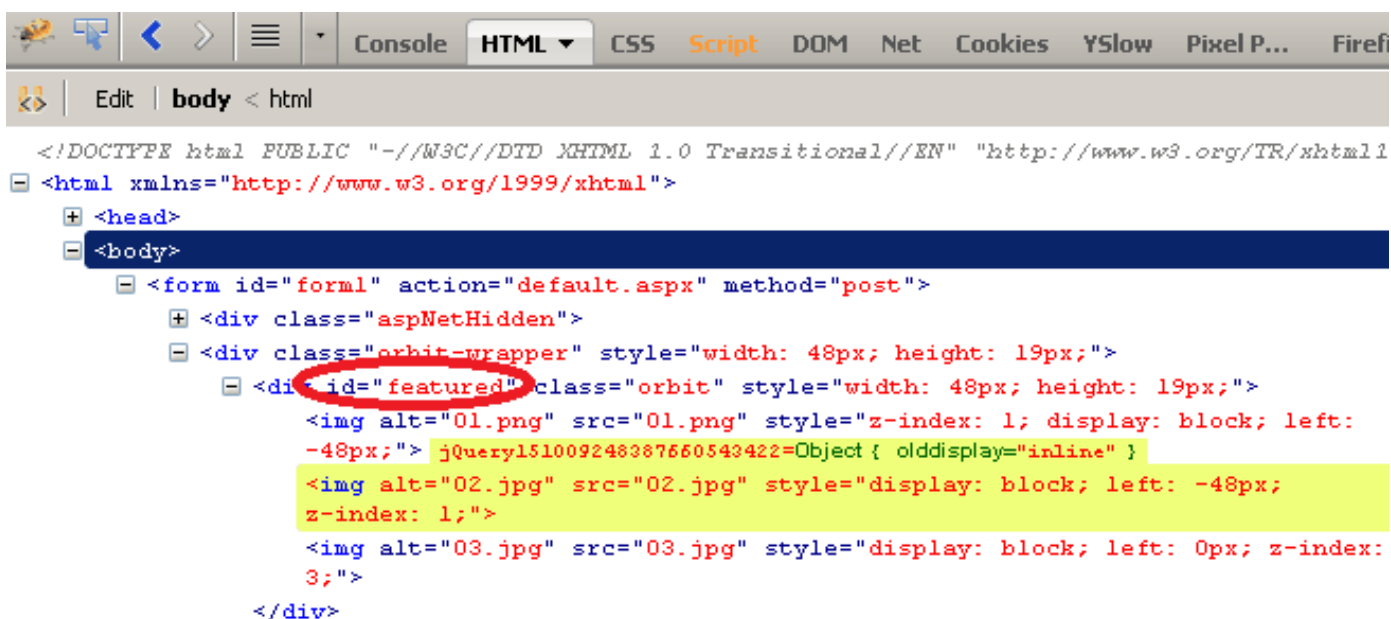
بنابراین در اینجا مشاهده کردیم که اگر «برنامه‌ای مبتنی بر jQuery کار نمی‌کند»، چگونه باید قدم به قدم با استفاده از فایرباگ و امکانات آن، به خطاهایی که گزارش می‌دهد و یا مسیرهایی را که یافت نشد بیان می‌کند، دقت کرد تا بتوان برنامه را عیب‌یابی نمود.

سؤال مهم: اجرای کدهای jQuery Ajax 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 اسکریپتی 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 به مرحله بررسی کرد. به این ترتیب بهتر می‌توان دریافت که آیا src بازگشت داده شده از سرور فرمت 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:



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1
<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;">
           jQuery151009248387660543422=Object { olddisplay="inline" }
          
          
        </div>
      </div>
    </form>
  </body>
</html>
```

نظرات خوانندگان

نویسنده:

صادق

تاریخ:

۱۳۹۲/۰۲/۰۴ ۱۲:۵۱

ممنون آقای نصیری - برای کروم چطور، آیا developer tools کفایت میکند یا ابزار بهتری سراغ دارید؟ البته من گاهی که نیاز به دستکاری ریسپانس و ریکوئست باشه از فیدلر استفاده میکنم

نویسنده:

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.

نویسنده:

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.

نویسنده:

Abolfazl

تاریخ:

۱۳۹۲/۰۲/۰۸ ۲۳:۴۸

سلام آقای نصیری ..ممنونم ..خیلی عالی بود
من قبلاً هم این مشکل را داشتم و نمی‌تونستم برطرفش کنم ..الان درست شد..مرسی

نویسنده:

محسن خان

تاریخ:

۱۳۹۲/۰۲/۰۹ ۱۶:۴۲

```
$(document).ready(function () {
    $(':radio').click(function () {
        debugger; //ویژوال استودیو
    });
});
```

یک نکته جانبی است برای فعال سازی دیباگر خود ویژوال استودیو در حین کار با جی‌کوئری

نویسنده:

ahmad.valipour

تاریخ:

۱۳۹۲/۰۲/۱۴ ۱۲:۱۹

با 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؟

نویسنده:

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 مثلاً در کدهای کار با Entity framework روی سطر break point قرار می‌دید و خروجی

یک کوئری را بررسی می‌کنید، این دیباگر 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: ۱۱:۵ ۱۳۹۲/۰۲/۲۹

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 خطایی وجود ندارد
ممنون

نویسنده: وحید نصیری
تاریخ: ۱۱:۲۹ ۱۳۹۲/۰۲/۲۹

حداقل دو علت می‌تونه داشته باشه:

الف) تصاویر رو نمی‌تونه پیدا کنه، یا صفحه کش شده بیش از حد. قسمت «اجرای کدهای jQuery Ajax 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 ایی 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 متد 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: ۱۰:۵۸ ۱۳۹۲/۰۷/۰۷

با تشکر از مطلب بسیار کاربردیتون ، در هنگام استفاده از URL Routing همانطور که قبلا 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 میتونید 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>
```

URL	Status	Domain	Size	Remote IP	Timeline
GET %D8%B1%D9%86%DA%AF-9	200 OK	localhost:3765	92.3 KB	127.0.0.1:3765	5.5s
GET yellow_mail.png	404 Not Found	localhost:3765	2.7 KB	127.0.0.1:3765	21.02s
GET male_user.png	404 Not Found	localhost:3765	2.7 KB	127.0.0.1:3765	21.03s
GET computer.png	404 Not Found	localhost:3765	2.7 KB	127.0.0.1:3765	21.63s
GET red_heart.png	404 Not Found	localhost:3765	2.7 KB	127.0.0.1:3765	22.17s
GET comments.png	404 Not Found	localhost:3765	2.7 KB	127.0.0.1:3765	22.72s
GET home.png	404 Not Found	localhost:3765	2.7 KB	127.0.0.1:3765	1m 3s
7 requests		108.3 KB		1m 9s (onload: 1m 9s)	

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؛ مثلاً:

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

به این شرط که این اسکریپت داخل صفحه runat=server دار باشد یا داخل head ایی با این مشخصات.



✱ 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 رو 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 می باشد

1-GetDynamicContext

این متد در زمان لود اطلاعات از دیتابیس استفاده می شود (وقتی که postback صورت می گیرد)

```
[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["height"].ToString(),
                    reader["width"].ToString(),
                    reader["text"].ToString(), reader["id"].ToString(),
                    reader["editable"].ToString()));
                }
            }
            conn.Close();
        }
    }

    return CoordsList;
}
```

DeleteCoords و SaveCoords- 2,3

این دو متد هم واسه ذخیره و حذف می باشند 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) {
    ///

```

```
$.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 () {
    ///

```

```
$.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 می‌تونید 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 رو بهتره اعمال کنید تا در دراز مدت و حالت‌های مختلف ورودی به مشکل برنخورید. به صورت خلاصه اطلاعات ارسالی رو جمع‌نزد و تبدیل به رشته نکنید. یک [شیء کامل](#) درست کنید و اجازه بدید JSON.stringify اون رو تبدیل کنه.

نویسنده: یزدان
تاریخ: ۱۳۹۲/۰۳/۱۸ ۱۰:۳۵

اگر متدهای وب سرویس رو درون صفحاتی قرار بدیم که نیاز به لاگین 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 لاگین) 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 اصلا جزئی از مباحث اعتبارسنجی صفحه جاری لحاظ نخواهند 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 handlerهای تولیدی، می‌شود همه نوع اعتبارسنجی رو اعمال کرد اعم از روش Forms Authentication مثلا توسط context.Request.IsAuthenticated یا حتی روش منسوخ شده استفاده از سشنها برای اعتبارسنجی با پیاده سازی IRequiresSessionState.
- در مطلب فوق اصلا از MS Ajax استفاده نشده. اون هم جایگاه خودش رو در کاربردهای خاص خودش دارد.

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 متد را از صفحه Default.aspx صدا بزنم 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 ایی بر روی صفحه ظاهر می‌شد و خطای 401 Unauthorized را نمایش می‌داد. حتی آدرس را به صورت Default/MyMethod/ امتحان کردم باز هم خطا داد و متد را شناسایی نمی‌کرد. به جای این کار کد زیر را در web.config اضافه کردم:

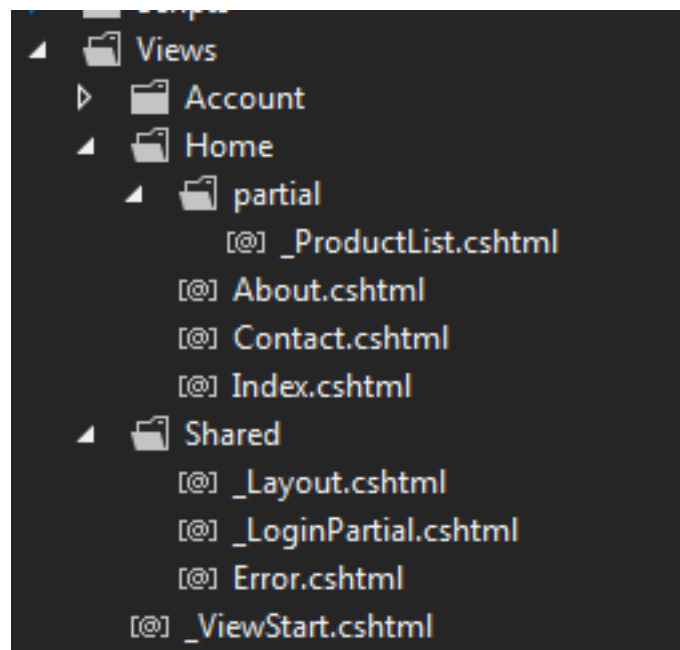
```
<system.web.extensions>
  <scripting>
    <webServices>
      <authenticationService enabled="true" />
    </webServices>
  </scripting>
</system.web.extensions>
```

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.

مزیت استفاده از PartialView ها، ماژولار 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 PartialView پیاده سازی کرد و سپس صفحه اصلی را از کنار هم قرار دادن این PartialView ها تهیه نمود. (منبع).
در این پست قصد دارم به نحوه بارگزاری یک PartialView با استفاده از ASP.NET MVC بپردازم.
ابتدا یک پروژه جدید ایجاد کنید. حال می‌خواهیم زمانیکه صفحه اصلی سایت بارگزاری می‌شود، لیست تمام محصولات را نمایش دهد. برای نمایش محصولات یک PartialView جدید را ایجاد می‌کنیم؛ همانند شکل ذیل:



حال برای ساده کردن مثال، متن ثابتی را درون این PartialView می‌نویسیم:

product List Partial

در ادامه قصد داریم که در Index، این View را بارگزاری کنیم. برای این کار از متد ajax مربوط به کتابخانه jQuery به صورت زیر استفاده می‌کنیم:

```
$(function () {
    $.ajax({
        // مشخص کردن اکشنی که باید فراخوانی شود
        url: '/Home/Details',
        contentType: 'application/html; charset=utf-8',
        type: 'GET',
        // نوع نتیجه بازگشتی
        dataType: 'html'
    })
    .success(function (result) {
        // زمانی که کدهای سمت سرور بدون خطا اجرا شده اند
        // این قسمت فراخوانی می‌شود و نتیجه اکشن درون متغیر
        // result
        // قرار می‌گیرد
        $('#sectionContents').html(result);
    });
});
```

```
})  
.error(function (xhr, status) {  
    alert(xhr.responseText);  
});  
});
```

و پس از آن محلی را که قرار است PartialView در آن بارگزاری شود، ایجاد می‌کنیم:

```
<div id="sectionContents"></div>
```

حال فقط باید اکشن مورد نظر را در HomeController پیاده سازی کنیم:

```
public ActionResult Details()  
{  
    return PartialView("partial/_ProductList");  
}
```

با استفاده از این کد مشخص کردیم که این اکشن، یک PartialView را با نام _ProductList برگشت می‌دهد. البته در یک مثال واقعی باید لیست محصولات را هم به این PartialView پاس دهیم. نتیجه اجرا به صورت زیر است:

your logo here

RegisterLog in

HomeAboutContact

Home Page. Modify this template to jump-start your ASP.NET MVC application.

To learn more about ASP.NET MVC visit <http://asp.net/mvc>. The page features **videos, tutorials, and samples** to help you get the most from ASP.NET MVC. If you have any questions about ASP.NET MVC visit [our forums](#).

We suggest the following:

- Getting Started**
ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and that gives you full control over markup for enjoyable, agile development. ASP.NET MVC includes many features that enable fast, TDD-friendly development for creating sophisticated applications that use the latest web standards. [Learn more...](#)
- Add NuGet packages and jump-start your coding**
NuGet makes it easy to install and update free libraries and tools. [Learn more...](#)
- Find Web Hosting**
You can easily find a web hosting company that offers the right mix of features and price for your applications. [Learn more...](#)

product List Partial

© 2013 - My ASP.NET MVC Application

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۹:۵۹ ۱۳۹۲/۰۵/۲۲

مطالب تکمیلی که از همین روش استفاده می‌کنند:

[به روز رسانی غیرهمزمان قسمتی از صفحه به کمک jQuery در ASP.NET MVC](#)

[پیاده سازی دکمه «بیشتر» یا «اسکرول نامحدود» به کمک jQuery در ASP.NET MVC](#)

[آشنایی با تکنیک‌های Ajax در ASP.NET MVC](#)

نویسنده: محسن خان
تاریخ: ۰:۲۸ ۱۳۹۲/۰۵/۲۳

یک سؤال: «بارگزاری» درست هست یا «[بارگذاری](#)»? البته [مطابق مطلبی که در ویکی‌پدیا نوشته](#) باید از «بارگیری» استفاده کرد!

نویسنده: imo0
تاریخ: ۱۰:۰۶ ۱۳۹۲/۰۶/۰۷

یک سوال. من یک پارشال ویو دارم که خودش نیاز داره به یه سری کدهای جاوا اسکریپت و استایل . من پارشال ویو هارو با اجکس لود میکنم . چطوری اون فایل‌های استایل و غیرشم لود کنم ؟ @RenderSection جواب نمیده. چون اصلا صفحه ریفرش نمیشه.

نویسنده: وحید نصیری
تاریخ: ۱۱:۰۸ ۱۳۹۲/۰۶/۰۷

- راه اول: زمانیکه View اصلی در برگیرنده آن نمایش داده می‌شود، کلیه فایل‌های متناظر را هم الحاق کنید تا به صورت خودکار در جزئی از صفحه آن، که بعدا به روز خواهد شد، نیز اعمال شود.
- راه دوم: اصلا از RenderSection در یک partial view که قرار است Ajax ایی بارگذاری شود، استفاده نکنید. معمولی این‌ها را الصاق یا تعریف کنید. مثل تعاریف یک HTML ساده. [یک نفر هم اینجا](#) براش HtmlHelper نوشته ولی نکته اصلی یکی است؛ الصاق و تعریف معمولی فایل‌های مورد نیاز.
- همچنین خود jQuery امکان بارگذاری اسکریپت‌ها را [به صورت پویا دارد](#) . زمانیکه complete عملیات Ajax ایی رخداد، متد getScript عنوان شده را فراخوانی کنید. برای CSS هم به صورت زیر عمل کنید:

```
$("<style></style>").appendTo("head").html(data);
```

نویسنده: خوشقدم
تاریخ: ۸:۵۴ ۱۳۹۲/۰۶/۱۳

سلام

مشکلی که من دارم این است که در صفحه دوتا PartialView دارم که هر کدام با استفاده از jquery همان طور که شما توضیح داده اید لود می‌شوند ، اما برای رفرش کردن اون‌ها مشکل دارم و نمی‌تونم به هر کدام شماره Page و سایر اطلاعات WebGrid چطوری ارسال کنم؟

ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۶/۱۳ ۹:۰

[نحوه صحیح ارسال پارامترها توسط jQuery Ajax](#)
[ارسال کوئری استرینگ‌ها به همراه عملیات Ajax در jQuery](#)

ضمناً بحث [وب گرید](#) ، خارج از موضوع مطلب جاری است. از پارامتر `ajaxUpdateContainerId` آن افزونه خاص باید برای مدیریت مسایل paging و sorting استفاده کنید.

نویسنده: مجتبی شایان فر
تاریخ: ۱۳۹۲/۰۶/۲۸ ۱۳:۴

باسلام و خسته نباشید.

من مثال فوق را خط به خط اجرا کردم ولی partial view نمایش داده نمیشه. فکر کنم مکان قطعه کد Ajax را اشتباه جایگذاری کردم اگه ممکنه راهنمایی می کنید که قطعه:

```
$( function () {  
$.ajax({  
// مشخص کردن اکشنی که باید فراخوانی شود  
url: '/Home/Details' ,  
contentType: 'application/html; charset=utf-8' ,  
type: 'GET' ,  
// نوع نتیجه بازگشتی  
dataType: 'html'  
})  
.success( function (result) {  
// زمانی که کدهای سمت سرور بدون خطا اجرا شده اند  
// این قسمت فراخوانی می شود و نتیجه اکشن درون متغیر  
// result  
// قرار می گیرد  
$('#sectionContents' ).html(result);  
})  
.error( function (xhr, status) {  
alert(xhr.responseText);  
});  
});
```

دقیقا کجای Index باید قرار داده بشه؟ سورس پروژه را هم ارسال می کردید خیلی خوب می شد .
ممنون.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۶/۲۸ ۲۰:۴

[نحوه استفاده از افزونه Firebug برای دیباگ برنامه های ASP.NET مبتنی بر jQuery](#)

نویسنده: منیژه محمدی
تاریخ: ۱۳۹۲/۰۸/۱۶ ۱۰:۴۴

سلام

من این مثال را قدم به قدم انجام دادم ولی خطای 404 میده و localhost.../home/Details را پیدا نمی کند.
با اینکه من پارشیال ویو را در این تابع تعریف کردم

نویسنده: وحید نصیری

تاریخ: ۱۰:۵۴ ۱۳۹۲/۰۸/۱۶

- برنامه را با فایرباگ [دیباگ کنید](#) تا مسیرهای درخواستی و خطاها را بهتر مشاهده کنید.
- نباید آدرس‌ها را مستقیماً و به صورت رشته‌ای تعریف کنید. [اطلاعات بیشتر](#)
- در مثال عنوان شده «partial/_ProductList» این مورد به معنای وجود یک پوشه partial در اینجا است (تصویر اول). البته غیر الزامی است ولی برای باز تولید این مثال باید به آن دقت داشته باشید.

نویسنده: منیژه محمدی
تاریخ: ۱۱:۳۶ ۱۳۹۲/۰۸/۱۷

با سلام
من همه موارد را خیلی چک کردم از اینکه مسیر یا نام را درست داده باشم
علت این بود که من یک [httpost] بالای ان ActionResult که میگفت پیدا نمیکنم اضافه کرده بودم برای همین ان را پیدا نمی‌کرد
حالا من میخام یک سری اطلاعات به این اکشن پست کنم اینطوری مشکل پیدا میکنم چطور میتونم ان را حل کنم ؟

نویسنده: وحید نصیری
تاریخ: ۱۱:۴۸ ۱۳۹۲/۰۸/۱۷

type در ajax.\$ مثال اصلی بحث جاری، مساوی Get است؛ تبدیلیش کنید به Post. [کامنت اول این مطلب](#) سه مثال مشابه دیگر را عنوان کرده. اکثر این‌ها از حالت Post استفاده کرده‌اند.

نویسنده: آروین
تاریخ: ۱۱:۱۹ ۱۳۹۲/۰۸/۲۱

با عرض سلام و خسته نباشید
این روش فرقی با `jquery.load('url')` چه؟

نویسنده: وحید نصیری
تاریخ: ۱۴:۲۴ ۱۳۹۲/۰۸/۲۱

load در پشت صحنه از همین متد ajax استفاده می‌کند. [تمام متدهای کمکی از این دست](#) فقط محصور کننده تابع ajax اصلی جی کوئری هستند.

نویسنده: رشنو
تاریخ: ۱۰:۳۶ ۱۳۹۲/۱۲/۱۵

با سلام و تشکر
لیست محصولات رو چطور باید پاس داد؟

نویسنده: وحید نصیری
تاریخ: ۱۰:۵۸ ۱۳۹۲/۱۲/۱۵

به روش‌های متداول « [بررسی نحوه انتقال اطلاعات از یک کنترلر به Viewهای مرتبط با آن](#) »

در حین انجام اعمال غیرهمزمان جاوا اسکریپتی مانند فراخوانی‌های jQuery AJAX، برای مدیریت دریافت نتایج، عموماً از یک سری callback استفاده می‌شود. برای مثال:

```
$.get('http://site-url', function(data) {
// این تابع پس از پایان کار عملیات ای‌جکسی در آینده فراخوانی خواهد شد
});
```

تا اینجا مشکلی به نظر نمی‌رسد. اما مورد ذیل چگونه؟

```
$.get('http://site-url/0', function(data0) {
// callback #1
$.get('http://site-url/1', function(data1) {
// callback #2
$.post('http://site-url/2', function(data2) {
// callback #3
});
});
});
```

در اینجا نیاز است پس از پایان کار عملیات Ajax ایی اول، عملیات دوم و پس از آن عملیات سومی انجام شود. همانطور که مشاهده می‌کنید، این نوع کدها به سرعت از کنترل خارج می‌شوند؛ خوانایی پایینی داشته و مدیریت استثناءهای رخ داده در آن‌ها نیز در این بین مشکل است. از این جهت که خطاهای هر کدام به سطحی بالاتر منتقل نمی‌شود و باید همانجا محلی و داخل هر callback مدیریت گردد.

روش‌های زیادی برای حل این مساله ارائه شده‌است و در حال حاضر کار کردن با promiseها متداول‌ترین روش حل مدیریت فراخوانی کدهای همزمان جاوا اسکریپتی است. برای نمونه اگر از AngularJS استفاده کنید، سرویس‌های آن برای دریافت اطلاعات از سرور، از یک چنین مفهومی استفاده می‌کنند.

Promise در جاوا اسکریپت چیست؟

شیء Promise، نمایانگر قراردادی است که در آینده می‌تواند مورد قبول واقع شود، یا رد گردد. بررسی این قرارداد، تنها یکبار می‌تواند رخ دهد (پذیرش یا رد آن). هنگامیکه این بررسی صورت گرفت (رد یا پذیرش آن و نه هردو)، یک callback برای اطلاع رسانی فراخوانی می‌گردد. سپس این callback می‌تواند یک Promise دیگر را سبب شود. به این ترتیب می‌توان Promiseها را زنجیر وار به یکدیگر متصل کرد. برای نمونه jQuery به صورت توکار از promises پشتیبانی می‌کند:

```
// returns a promise
$.get('http://site-url/0')
.then(function(data) {
// callback 1
// returns a promise
return $.get('http://site-url/1');
})
.then(function(data) {
// callback 2
// returns a promise
return $.post('http://site-url/2');
})
.then(function(data) {
// callback 3
});
```

متد get در jQuery یک شیء promise را بازگشت می‌دهد. در ادامه می‌توان این نتیجه را توسط متد then، زنجیروار ادامه داد. متدی که به عنوان پارامتر به then ارسال می‌شود، یک callback بوده و پس از پایان کار promise قبلی رخ می‌دهد. آنگه‌مانی که به

این callback ارسال می‌شود، نتیجه‌ی promise قبلی است. در حین اعمال jQuery Ajax، این callback تنها زمانی فراخوانی می‌شود که عملیات قبلی موفقیت آمیز بوده باشد و data ارائه شده، اطلاعاتی است که توسط response دریافتی از سرور، دریافت گردیده‌است.

در این حالت، هر callback حداقل سه کار را می‌تواند انجام دهد:

(الف) یک promise دیگر را بازگشت دهد. نمونه آن را با `return $.get` در کدهای فوق ملاحظه می‌کنید.

(ب) خاتمه عادی. همینجا کار promise با مقدار بازگشت داده شده، پایان می‌یابد.

(ج) صدور یک استثناء. سبب برگشت خوردن و عدم پذیرش promise می‌شود.

استفاده از Promises در سایر کتابخانه‌ها

jQuery پیاده سازی توکاری از promises دارد؛ اما سایر کتابخانه‌ها، مانند AngularJS ایی که مثال زده شده چطور عمل می‌کنند؟ استاندارد به نام [Promises/A+](#) جهت یک دست سازی پیاده سازی‌های promise در جاوا اسکریپت پیشنهاد شده‌است. jQuery نیمی از آن را پیاده سازی کرده‌است؛ اما کتابخانه‌ی دیگری به نام [Q Library](#)، پیاده سازی نسبتاً مفصل‌تری را از این استاندارد ارائه می‌دهد. فریم ورک AngularJS نیز در پشت صحنه از همین کتابخانه برای پیاده سازی promises استفاده می‌کند.

آشنایی با کتابخانه Q

استفاده مقدماتی از Q همانند مثالی است که از jQuery ملاحظه کردید.

```
Q.fcall(callback1)
  .then(callback2);
```

اشیاء promise بازگشت داده شده توسط jQuery نیز توسط کتابخانه Q مورد پذیرش واقع می‌شوند:

```
Q.fcall(function() {
  return $.get('http://my-url');
})
  .then(callback3);
```

علاوه بر این‌ها مفهومی به نام deferred objects نیز در کتابخانه‌ی Q پیاده سازی شده‌است:

```
function waitForClick() {
  var deferred = Q.defer();

  $('#okButton').click(function() {
    deferred.resolve();
  });

  $('#cancelButton').click(function() {
    deferred.reject();
  });

  return deferred.promise;
}

Q.fcall(waitForClick)
  .then(function() {
    // ok button was clicked
  }, function() {
    // cancel button was clicked
  });
```

توسط deferred objects می‌توان بررسی یک promise را به تاخیر انداخت. در مثال فوق، اولین callback فراخوانی شده به نام `waitForClick`، از اشیاء به تاخیر افتاده استفاده می‌کند. ابتدا توسط فراخوانی متد `Q.defer`، یک deferred object ایجاد می‌شود. در این بین اگر کاربر بر روی دکمه‌ی OK کلیک کرد، با فراخوانی `deferred.resolve`، این promise مورد پذیرش واقع خواهد شد و یا اگر کاربر بر روی دکمه‌ی cancel کلیک کند، با فراخوانی متد `deferred.reject`، این promise رد می‌گردد. نهایتاً شیء promise

توسط deferred.promise بازگشت داده خواهد شد.

در ادامه کار، اینبار متد then، دو callback را قبول می‌کند. Callback اول پس از پذیرش قرار داد و Callback دوم پس از رد قرار داد، فراخوانی خواهد گردید.
در رنجیره تعریف شده، اگر معادلی برای reject در نظر گرفته نشده باشد، مانند مثال ذیل:

```
Q.fcall(myFunction1)
  .then(success1)
  .then(success2, failure1);
```

Q به دنبال نزدیک‌ترین متد callback گزارش خطای کار خواهد گشت. در این حالت متد failure1 در صورت شکست اولین promise فراخوانی خواهد شد.

همچنین اگر نتیجه‌ی success1 با شکست مواجه شود نیز failure1 فراخوانی می‌گردد. اما باید در نظر داشت که شکست success2، توسط failure1 مدیریت نمی‌شود.

AngularJS در Promises

در AngularJS امکانات کتابخانه Q توسط پارامتری به نام \$q در اختیار سرویس‌های برنامه قرار می‌گیرد (تزریق می‌شود):

```
var app = angular.module("myApp", []);
app.factory('dataSvc', function($http, $q){
  var basePath="api/books";
  getAllBooks = function(){
    var deferred = $q.defer();
    $http.get(basePath).success(function(data){
      deferred.resolve(data);
    }).error(function(err){
      deferred.reject("service failed!");
    });
    return deferred.promise;
  };

  return{
    getAllBooks:getAllBooks
  };
});

app.controller('HomeController', function($scope, $window, dataSvc){
  function initialize(){
    dataSvc.getAllBooks().then(function(data){
      $scope.books = data;
    }, function(msg){
      $window.alert(msg);
    });
  }

  initialize();
});
```

در اینجا اگر دقت کنید، مباحث و عملکرد آن دقیقاً مانند قبل است. ابتدا یک deferred object با فراخوانی متد q.defer ایجاد شده است. سپس با استفاده از امکانات توکار http آن (بجای استفاده از jQuery Ajax)، کار فراخوانی یک restful service صورت گرفته است (مثلاً فراخوانی یک ASP.NET Web API). در صورت موفقیت کار، متد deferred.resolve و در صورت عدم موفقیت، متد deferred.reject فراخوانی شده‌است. نهایتاً این سرویس، یک deferred.promise را بازگشت می‌دهد.
اکنون در کنترلی که قرار است از این سرویس استفاده کند، متد then کتابخانه Q را ملاحظه می‌کنید که دو Callback متناظر resolve و reject مدیریت promise بازگشت داده شده را به همراه دارد. اگر عملیات Ajaxی موفقیت آمیز باشد، شیء books را مقدار دهی می‌کند و اگر خیر، پیامی را به کاربر نمایش خواهد داد.

پشتیبانی مرورگرهای جدید از استاندارد Promise

در حال حاضر کروم 32 و نگارش‌های شبانه فایرفاکس، Promise را که جزئی از استاندارد JavaScript شده‌است، به صورت توکار

و بدون نیاز به کتابخانه‌های جانبی، پشتیبانی می‌کنند.

```
if (window.Promise) { // Check if the browser supports Promises
  var promise = new Promise(function(resolve, reject) {
    //asynchronous code goes here
  });
}
```

در اینجا با فراخوانی `window.Promise` مشخص می‌شود که آیا مرورگر جاری از Promises پشتیبانی می‌کند یا خیر. سپس یک شیء `promise` ایجاد شده و این شیء توسط پارامترهای `resolve` و `reject` که هر دو تابع می‌باشند، کار مدیریت کدهای غیرهمزمان را انجام می‌دهد:

```
if (window.Promise) {
  console.log('Promise found');

  var promise = new Promise(function(resolve, reject) {
    // async
    if (result) {
      resolve(data);
    } else {
      reject('error');
    }
  });

  promise.then(function(data) {
    console.log('Promise fulfilled.');
```

```
  }, function(error) {
    console.log('Promise rejected.');
```

```
  });
} else {
  console.log('Promise not available');
```

```
}
```

در مثال فوق ابتدا یک شیء `Promise` ایجاد شده است. این شیء استاندارد بوده و با کروم 32 قابل آزمایش است. سپس در `callback` ابتدایی آن می‌توان یک عملیات `AJAX` ایی را انجام داد. اگر نتیجه‌ی آن موفقیت آمیز بود، تنها کافی است پارامتر اول این `callback` را فراخوانی کنیم و اگر خیر، پارامتر دوم آن را. برای استفاده از این شیء `Promise` ایجاد شده، می‌توان از متد `then` استفاده کرد. این متد نیز در اینجا دو `callback` پذیرش و رد `promise` را می‌تواند دریافت کند. برای زنجیر کردن آن کافی است متد `then`، یک `Promise` دیگر را بازگشت دهد و از نتیجه‌ی آن در `then` بعدی استفاده گردد.

عموماً از ajax برای ارائه سایت‌هایی سریع، با حداقل ریفرش و حداقل مصرف پهنای باند سرور، استفاده می‌شود. اما این روش، مشکلات خاص خود را نیز دارا است. عموماً محتوای پویای بارگذاری شده، سبب تغییر آدرس صفحه‌ی جاری در مرورگر نمی‌شود. برای مثال اگر قرار است چندین برگه در صفحه به صورت ajax ای بارگذاری شوند، تغییر سریع محتوا را مشاهده می‌کنید، اما خبری از تغییر آدرس جاری صفحه در مرورگر نیست. همچنین روش‌های ajax ای عموماً SEO friendly نیستند. زیرا اکثر موتورهای جستجو فاقد پردازشگرهای جاوا اسکریپت می‌باشند و محتوای پویای ajax ای را مشاهده نمی‌کنند. برای آدرس دهی این مشکلات مهم، افزونه‌ای به نام [pjax](#) طراحی شده‌است که کار آن دریافت محتوای HTML ای از سرور و قرار دادن آن در یک جایگاه خاص مانند یک div است. در پشت صحنه‌ی آن از jQuery ajax استفاده شده، به همراه push state

pjax = pushState + AJAX

Push state API همان HTML5 History API است؛ به این معنا که هرچند محتوای صفحه‌ی جاری به صورت پویا بارگذاری می‌شود، اما آدرس مرورگر نیز به صورت خودکار تنظیم خواهد شد؛ به همراه عنوان صفحه. به علاوه تاریخچه‌ی مرور صفحات نیز در مرورگر به روز رسانی شده و امکان حرکت بین صفحات توسط دکمه‌های back و forward همانند قبل وجود خواهد داشت. همچنین اگر مرورگر جاری سایت، امکان استفاده از جاوا اسکریپت را نداشته باشد، به صورت خودکار به حالت بارگذاری کامل صفحه سوئیچ خواهد کرد.

سایت‌های بسیاری خودشان را با این الگو وفق داده‌اند. برای نمونه Twitter و Github از مفهوم pjax استفاده‌ی وسیعی دارند. برای نمونه، layout یا master page یک سایت را در نظر بگیرید. به ازای مرور هر صفحه، یکبار باید تمام قسمت‌های تکراری layout از سرور بارگذاری شوند. توسط pjax به سرور اعلام می‌کنیم، ما تنها نیاز به body صفحات را داریم و نه کل صفحه را. همچنین اگر مرورگر از جاوا اسکریپت استفاده نمی‌کند، لطفاً کل صفحه را همانند گذشته بازگشت بده. به علاوه مسایل سمت کلاینت مانند تغییر آدرس مرورگر و تغییر عنوان صفحه نیز به صورت خودکار مدیریت شوند. این تکنیک را دقیقاً در حین مرور مخزن‌های کد Github می‌توانید مشاهده کنید. فقط قسمتی که لیست فایل‌ها را ارائه می‌دهد، از سرور دریافت می‌گردد و نه کل صفحه.

بکارگیری pjax در ASP.NET MVC

مطابق توضیحاتی که ارائه شد، برای پیاده سازی سازی pjax نیاز به دو فایل layout داریم. یکی برای حالت ajax ای و دیگری برای حالت بارگذاری کامل صفحه. حالت ajax ای آن تنها از رندرکردن body پشتیبانی می‌کند؛ و نه ارائه تمام قسمت‌های صفحه مانند هدر، فوتر، منوها و غیره. بنابراین خواهیم داشت:

الف) تعریف فایل‌های layout سازگار با pjax

ابتدا یک فایل جدید را به نام PjaxLayout.cshtml به پوشه‌ی Shared اضافه کنید؛ با این محتوا:

```
<title>@ViewBag.Title</title>
@RenderBody()
```

سپس layout اصلی سایت را به نحو ذیل تغییر دهید

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width" />
  <title>@ViewBag.Title</title>
  <link href="~/Content/Site.css" rel="stylesheet" />
  <script src="~/Scripts/jquery-1.8.2.min.js"></script>
  <script src="~/Scripts/jquery.pjax.js"></script>
```



```

<script type="text/javascript">
    $(function () {
        $(document).pjax('a[withpjax]', '#pjaxContainer', { timeout: 5000 });
    });
</script>
</head>
<body>
    <div>Main layout ...</div>
    <div id="pjaxContainer">
        @RenderBody()
    </div>
</body>
</html>

```

در فایل PjaxLayout خبری از هدر و فوتر نیست و فقط یک عنوان و نمایش body را به همراه دارد. فایل layout اصلی سایت همانند قبل است. فقط RenderBody آن داخل یک div با id مساوی pjaxContainer قرار گرفته و از آن در فراخوانی افزونه‌ی pjax استفاده شده است. همانطور که ملاحظه می‌کنید، مطابق تنظیمات ابتدای هدر layout، فقط لینک‌هایی که دارای ویژگی withpjax باشند، توسط pjax پردازش خواهند شد.

ب) تغییر فایل ViewStart برنامه

در فایل ViewStart، کار مقدار دهی layout پیش فرض صورت گرفته است. اکنون نیاز است این فایل را جهت معرفی layout دوم تعریف شده مخصوص pjax، اندکی ویرایش کنیم:

```

@{
    if (Request.Headers["X-PJAX"] != null)
    {
        Layout = "~/Views/Shared/_PjaxLayout.cshtml";
    }
    else
    {
        Layout = "~/Views/Shared/_Layout.cshtml";
    }
}

```

افزونه‌ی pjax، هدری را به نام X-PJAX به سرور ارسال می‌کند. بر این اساس می‌توان تصمیم گرفت که آیا از layout اصلی (در صورتیکه مرورگر از جاوا اسکریپت پشتیبانی نمی‌کند و این هدر را ارسال نکرده است) یا از layout سبک‌تر pjax استفاده شود.

ج) آزمایش برنامه

```

using System.Web.Mvc;

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

        public ActionResult About()
        {
            return View();
        }
    }
}

```

یک کنترلر ساده را به نحو فوق با دو اکشن متد و دو View متناظر با آن ایجاد کنید. سپس View متد Index را به نحو ذیل تغییر دهید:

```

@{
    ViewBag.Title = "Index";
}

```

```
<h2>Index</h2>

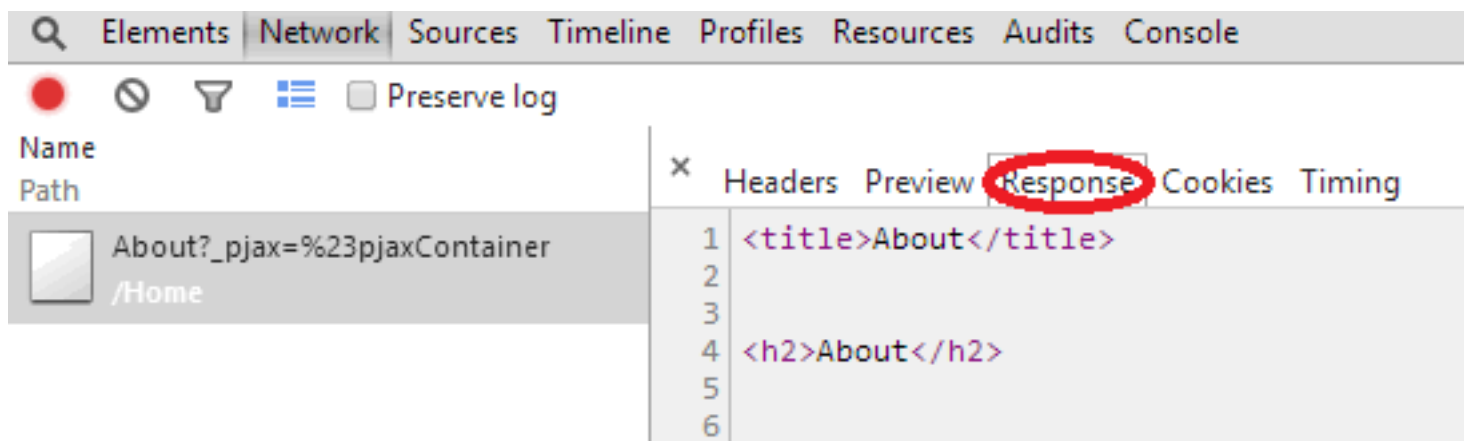
@Html.ActionLink(linkText: "About", actionName:"About", routeValues: null,
                  controllerName:"Home", htmlAttributes: new { withpjax = "with-pjax"})
```

در این View یک لینک معمولی به اکشن متد About اضافه شده است. فقط در ویژگی‌های html آن، یک ویژگی جدید به نام withpjax را نیز اضافه کرده‌ایم تا در صورت امکان و پشتیبانی مرورگر، از pjax استفاده شود. اکنون اگر برنامه را اجرا کنید، چنین خروجی را در برگه‌ی network آن مشاهده خواهید کرد:

The screenshot shows a web browser window with the address bar displaying `localhost:4640/Home/About`. Below the browser, the developer tools are open to the 'Network' tab. A request named `About?_pjax=%23pjaxContainer/Home` is selected. The 'Headers' sub-tab is active, showing the following details:

- Remote Address: 127.0.0.1:5656
- Request URL: `http://localhost:4640/Home/About?_pjax=%23pjaxContainer` (highlighted with a red line)
- Request Method: GET
- Status Code: 200 OK
- Request Headers:
 - Accept: text/html, */*; q=0.01
 - Accept-Encoding: gzip, deflate, sdch
 - Accept-Language: en-US,en;q=0.8
 - Content-Type: application/x-www-form-urlencoded; charset=UTF-8
- Host: localhost:4640
- Proxy-Connection: keep-alive
- Referer: http://localhost:4640/
- User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
- X-PJAX: true (highlighted with a red bar)
- X-PJAX-Container: #pjaxContainer
- X-Requested-With: XMLHttpRequest

همانطور که ملاحظه می‌کنید، با کلیک بر روی لینک About، یک درخواست pjax ای به سرور ارسال شده است؛ به همراه هدرهای ویژه آن. هنوز قسمت‌های اصلی layout سایت مشخص هستند (و مجدداً از سرور درخواست نشده‌اند). آدرس صفحه عوض شده است. به علاوه قسمت body آن تنها تغییر کرده است.



این مثال را از اینجا نیز می‌توانید دریافت کنید

[PajxMvcApp.zip](#)

برای مطالعه بیشتر

[A Faster Web With PJAX](#)

[Favour PJAX over dynamically loaded partial views](#)

[What is PJAX and why](#)

[Pjax.Mvc](#)

[Using pjax with ASP.Net MVC3](#)

[Getting started with PJAX with ASP.NET MVC](#)

[ASP.NET MVC with PAjax or PushState/ReplaceState and Ajax](#)

نظرات خوانندگان

نویسنده: رضایی
تاریخ: ۱۶:۲۳ ۱۳۹۳/۰۳/۰۳

با سلام؛ آیا pjax توسط کلیه مرورگرها پشتیبانی میشه؟ مخصوصا IE ؟

نویسنده: وحید نصیری
تاریخ: ۱۷:۱ ۱۳۹۳/۰۳/۰۳

- فقط در مرورگرهایی پشتیبانی می شود که push state را پیاده سازی کرده باشند: [لیست کامل آن ها](#)
- اگر مرورگری API history.pushState را پشتیبانی نکند، بارگذاری صفحات آن معمولی خواهند بود (شبیه به حالت بارگذاری کامل برای موتورهای جستجو؛ بدون از کار افتادن برنامه).

نویسنده: MD
تاریخ: ۲۳:۵۶ ۱۳۹۳/۰۳/۰۴

با سلام
آیا در Asp Web Form هم قابل استفاده است؟

نویسنده: وحید نصیری
تاریخ: ۰:۲۶ ۱۳۹۳/۰۳/۰۵

در مثالی که توضیح داده شد، محتوای قسمتی از صفحه به صورت پویا با محتوای صفحه ای دیگر جایگزین می شود. صفحه ای اول با master page کامل سایت رندر می شود. با کلیک بر روی لینک مشاهده ای صفحه ای بعدی، فقط محتوای آن صفحه (بدون master page اصلی سایت؛ شکل دوم) بجای div محتوای صفحه ای اول تزریق می شود. اگر صفحه ای دوم به صورت معمولی درخواست شود، با master page کامل سایت رندر خواهد شد. اما ... در وب فرم ها هر چند امکان انتخاب master page به صورت پویا [وجود دارد](#) ، اما به علت اینکه هر صفحه View State خاص خودش را خواهد داشت (بر اساس کنترل هایی که دارد)، تزریق محتوای آن داخل یک صفحه ای دیگر سبب تخریب View state جاری و از پیش موجود می شود. در نتیجه امکان ارسال اطلاعات به سرور را با پیام view state is corrupted از دست خواهید داد.

نویسنده: رضایی
تاریخ: ۱۳:۵ ۱۳۹۳/۰۳/۰۵

با سلام؛ تمام مراحل رو انجام دادم اما جواب نمیده. مشکل از کد زیر که نیست؟

```
@Html.ActionLink(item.Name, item.ActionName, item.ControllerName,
new { Id = item.Id, area = item.AreaName }, new { @class = "drop", withpjax = "with-pjax" })
```

نویسنده: وحید نصیری
تاریخ: ۱۳:۱۳ ۱۳۹۳/۰۳/۰۵

- مثال امتحان شده، در انتهای بحث پیوست شده.
- برای دیباگ و [خطایابی](#) برنامه های جاوااسکریپتی، به این مطلب مراجعه کنید: « [نحوه استفاده از افزونه Firebug برای دیباگ برنامه های ASP.NET مبتنی بر jQuery](#) »

نویسنده: رضایی
تاریخ: ۱۴:۴۳ ۱۳۹۳/۰۳/۰۵

با سلام؛ زمانی که یک ویو رو از طریق pjax نمایش میدم اگه داخل اون ویو یک لینک وجود داشته باشد که از طریق متد post یک اکشن رو فراخوانی می‌کند لینک کار نمی‌کند. مثلاً در پروژه IRIS توی بعضی صفحات که دکمه امتیاز دهی وجود داره اگه این صفحات با pjax لود بشن دیگه دکمه‌ها کار نمی‌کنن. به نظر شما مشکل کجاست؟ آیا اسکریپت‌ها لود نمیشن؟

نویسنده: وحید نصیری

تاریخ: ۱۵:۱ ۱۳۹۳/۰۳/۰۵

- المان‌هایی که به صورت پویا به صفحه اضافه می‌شوند، تحت کنترل مجدد jQuery نخواهند بود، مگر اینکه از متدهای [live](#) (منسوخ شده) و یا [on](#) استفاده شود. مطابق [مستندات](#) این کتابخانه (انتهای صفحه)، متد on به صورت پیش فرض، برای کارکرد مجدد pjax، اعمال می‌شود. بنابراین فقط باید بررسی کنید که آیا نگارش jQuery مورد استفاده، از متد on پشتیبانی می‌کند یا خیر (از آخرین نگارش آن استفاده کنید).

- همچنین تمام کدهای سایر قسمت‌های برنامه هم مانند دکمه امتیازدهی که اشاره شد، باید تغییر کرده و از متد on استفاده کنند. مثلاً اگر click دارند، باید بشوند on click.

```
//The new api
$(document).pjax('a[withpjax]', '#pjax-container')

//Which is roughly the same as
$(document).on('click', 'a[withpjax]', function(event) {
    $.pjax.click(event, '#pjaxContainer')
})
```

نویسنده: مهدی

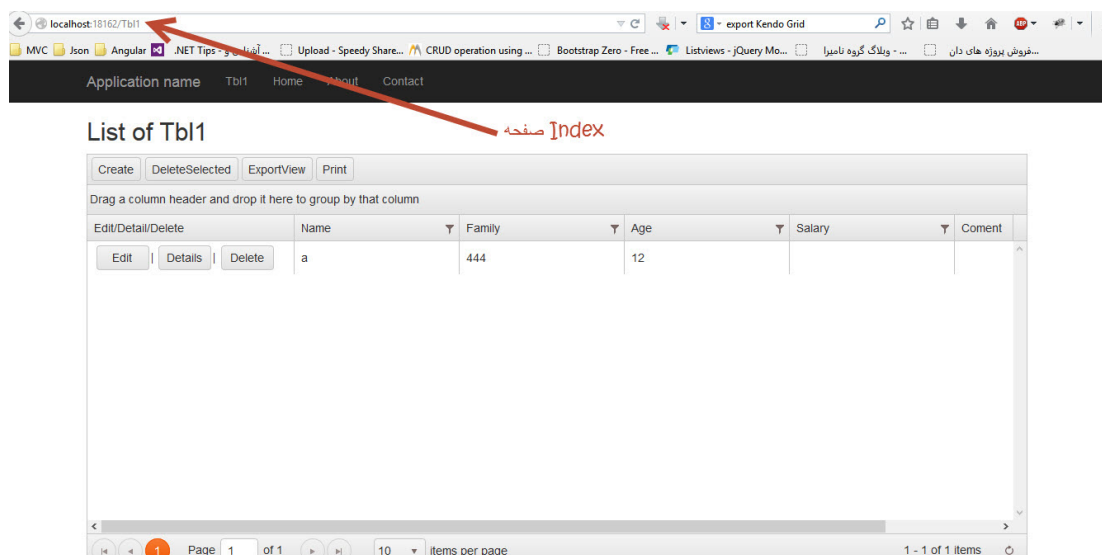
تاریخ: ۲۱:۱۲ ۱۳۹۳/۰۳/۱۶

همه چیز خوبه؛ اما وقتی توی گرید میاییم میزنیم Edit و میریم صفحه Edit و گزینه Save رو میزنیم، برمیگردیم به صفحه Index در حالی که لینک تغییر نکرده و لینک مثلاً اینطوری مونده

www.test.com/edit/1

و وقتی میزنیم گرید به صفحه 2 بره ارور میده که اصلاً همچین صفحه ای وجود نداره

حالا راهی نیست که از Controller که کد زدیم برای Edit و در آخر گفتیم بره به صفحه Index این Pjax و اینا رو آخر لینک ننویسه؟



Application name Tbl1 Home About Contact

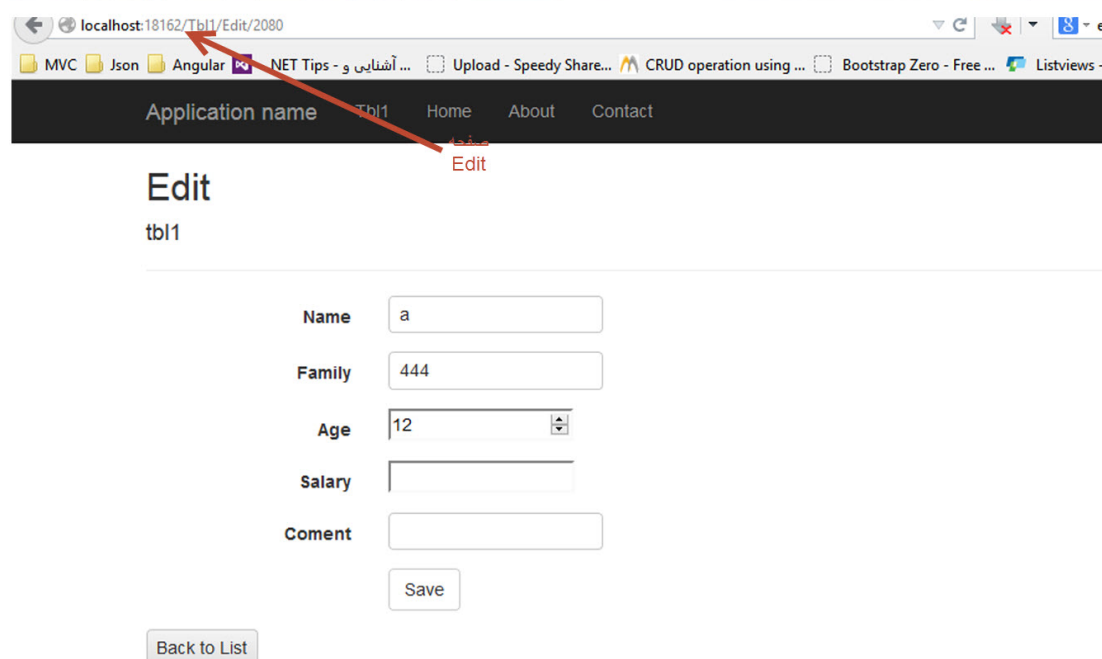
List of Tbl1

Create DeleteSelected ExportView Print

Drag a column header and drop it here to group by that column

Edit/Detail/Delete	Name	Family	Age	Salary	Coment
Edit Details Delete	a	444	12		

Page 1 of 1 10 items per page 1 - 1 of 1 items



Application name Tbl1 Home About Contact

Edit

tbl1

Name

Family

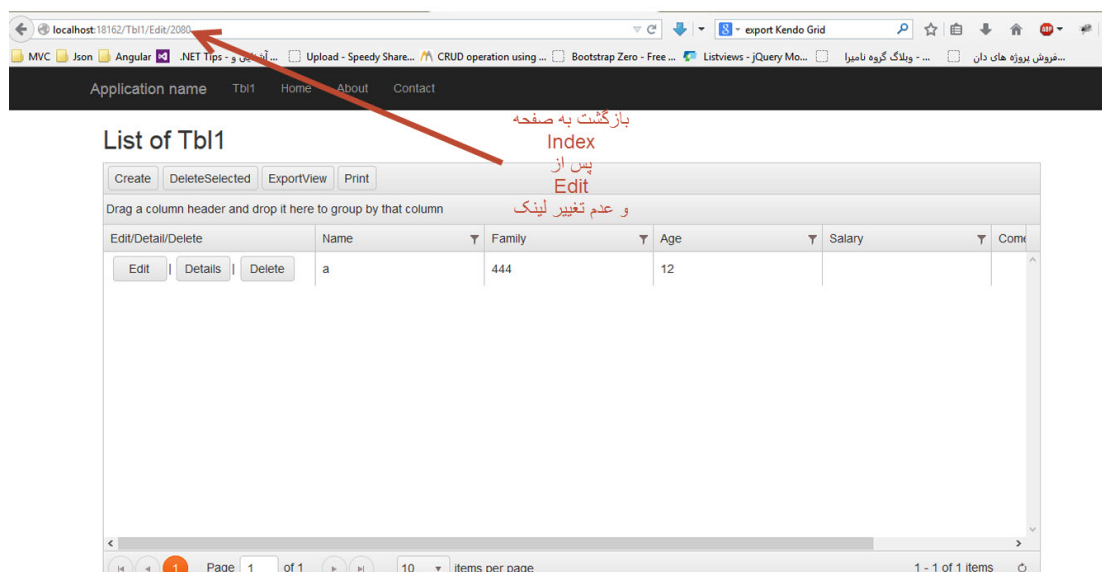
Age

Salary

Coment

Save

Back to List



Application name Tbl1 Home About Contact

List of Tbl1

Create DeleteSelected ExportView Print

Drag a column header and drop it here to group by that column

Edit/Detail/Delete	Name	Family	Age	Salary	Coment
Edit Details Delete	a	444	12		

Page 1 of 1 10 items per page 1 - 1 of 1 items

نویسنده:

وحید نصیری

تاریخ:

۲۱:۲۵ ۱۳۹۳/۰۳/۱۶

در مثالی که زده شد، فقط لینک‌هایی که دارای ویژگی withpjax هستند تحت کنترل این افزونه قرار می‌گیرند و نه هیچ لینک دیگری در برنامه و نه هیچ روش بازگشت دیگری:

```
$(document).pjax('a[withpjax]', '#pjaxContainer', { timeout: 5000 });
```

البته این یک مثال است و اگر مثلاً withpjax آنرا حذف کنید:

```
$(document).pjax('a', '#pjaxContainer', { timeout: 5000 });
```

تمام لینک‌های صفحه تحت کنترل خواهند بود. شبیه به حالتی که عنوان کردید صفحه بندی گرید به هم خورده. برای اینکه این نوع تداخل‌ها رخ ندهند و هر لینکی در صفحه توسط این افزونه پردازش نشود، بهتر است از روش پیشنهادی استفاده کنید.

نویسنده:

رمزینه

تاریخ:

۲۱:۳۸ ۱۳۹۳/۰۳/۱۶

نمی‌دونم این Pjax مشکل را چگونه بوجود میاره، اما از وقتی ارزش استفاده کردم، توی Save کردن در صورتی که از گزینه‌های خود یک Textbox که خود Browser پیشنهاد میداد استفاده بشه، چندتا چندتا ذخیره میشه:

List of Tbl1

CreateDeleteSelectedExportViewPrint

Drag a column header and drop it here to group by that column

Edit/Detail/Delete	Name	Family	Age	Salary	Coment
<div>EditDetailsDelete</div>	a	444	12		

162/Tbl1/Create?_pjax=#pjaxContainer

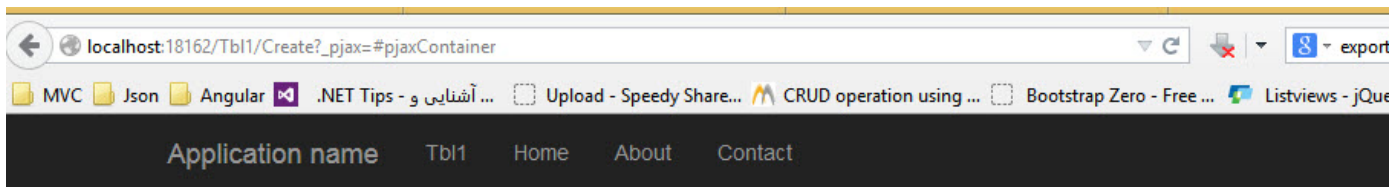
1

of 1

10

items per page

1 - 1 of 1 items



Create

tbl1

رکوردهایی که قبلاً در این فیلد وارد شده است

و حالا مجدداً خود

Browser

پیشنهاد میدهد

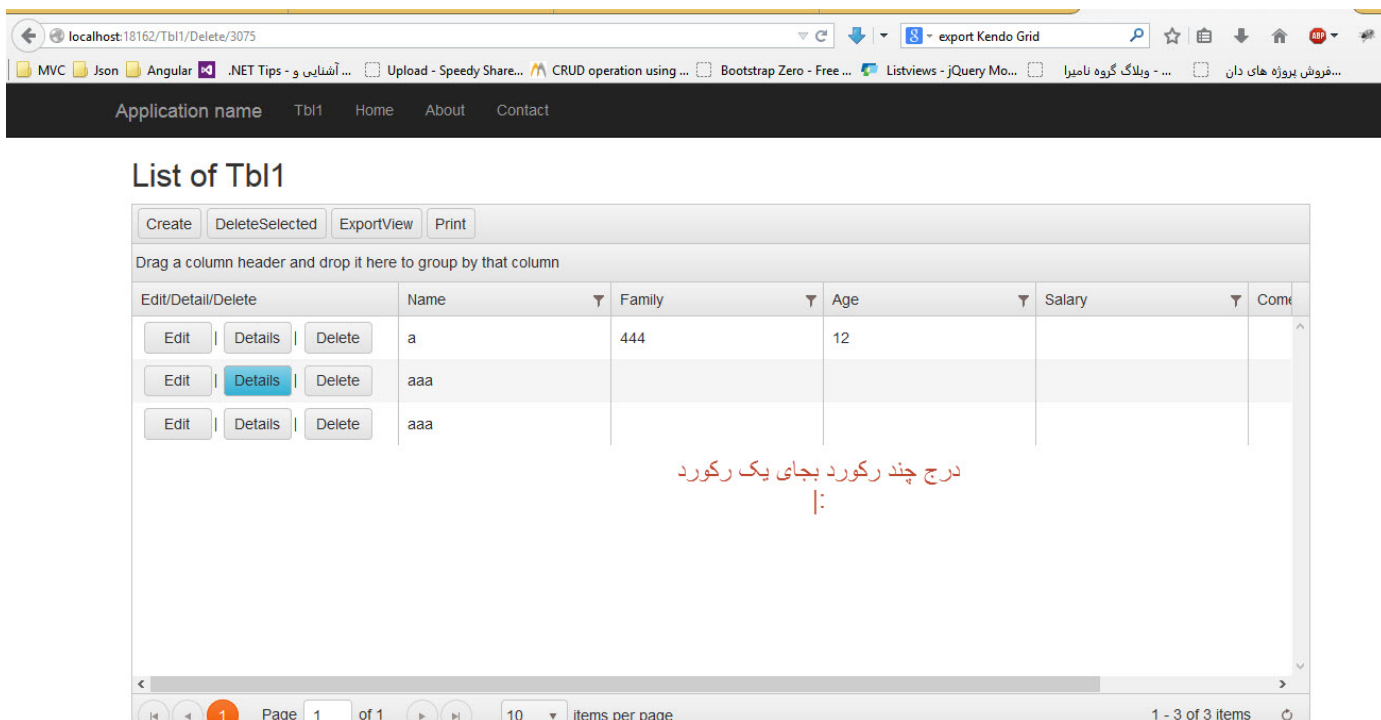
Name	<input type="text" value="a"/>
Family	<div>aaaa as asdv aaa asd a</div>
Age	<input type="text"/>
Salary	<input type="text"/>
Coment	<input type="text"/>

Create

Back to List

© 2014 - My Telerik MVC Application

و اینم خروجی یک Save



نویسنده: رمزینه
تاریخ: ۲۱:۴۳ ۱۳۹۳/۰۳/۱۶

ممنون از جوابتون اما مشکل من همینجاست که اصلاً به هیچ خصوصیتی ندادم

```
<input type="submit" value="Save" />
```

اینو گذاشتم اما چون توی

```
@Ajax.beginform(...)
```

هست و توی

```
<div id="pjaxContainer">
    @RenderBody()
</div>
```

که توی layout هست داره بصورت pjax عمل میکنه.

نویسنده: وحید نصیری
تاریخ: ۲۲:۰۲ ۱۳۹۳/۰۳/۱۶

- لازم هست با نحوه‌ی دیباگ برنامه‌های Ajax و برنامه‌های مبتنی بر jQuery آشنا شوید (جهت بررسی اینکه مشخص شود بدون تنظیمات این افزونه، آیا عملیات انجام شده، Ajax ایی است یا Pjax ایی). [اطلاعات بیشتر](#)

- کار pjax فقط ارائه محتوای صفحات است. اگر فعال هم نباشد، برنامه بدون مشکل کار می‌کند و صفحات آن نمایش داده خواهند شد.

نویسنده: وحید نصیری
تاریخ: ۲۲:۱۳ ۱۳۹۳/۰۳/۱۶

با عکس [نمی‌شود](#) یک برنامه را از راه دور [دیباگ کرد](#) .

نویسنده: احسان شیروان
تاریخ: ۱۱:۳۹ ۱۳۹۳/۰۳/۱۸

تصور بنده بر اینه که شما باید برای داشتن فرم همون Html.BeginForm معمولی رو استفاده کنید و نه Ajax.BeginForm چون به صورت توکار pjax داره از Ajax بهره می‌بره ممکنه یه تداخلی هم این وسط پیش بیاد

نویسنده: مهدی
تاریخ: ۰:۵۳ ۱۳۹۳/۰۳/۱۹

مشکل همینجاست فکر کنم که از beginForm چه عادی چه Ajax ای استفاده میشه در حالت Ajax که مشکل اساسی هست و چندتا چندتا Insert میشه و یه جورایی همینطور تو حلقه میمونه از Html.beginForm استفاده میکنم رفرش میشه از هیچکدوم استفاده نکنم نمیدونم چطوری اطلاعات صفحه رو به یه Action توی کنترلر با دکمه Submit ارسال کنم!

[Ajax.BeginForm](#) در ASP.NET MVC از jQuery Ajax برای ارسال مقادیر فرم، به سرور استفاده می‌کند. در این بین اگر یکی از عناصر فرم، المان ارسال فایل به سرور باشد، مقدار دریافتی در سمت سرور نال خواهد بود. مشکل اینجا است که نمی‌توان به کمک Ajax معمولی (یا به عبارتی XMLHttpRequest) فایلی را به سرور ارسال کرد. یا باید از سیلورلایت یا فلش استفاده نمود و یا از مرورگرهایی که [XMLHttpRequest Level 2](#) را پشتیبانی می‌کنند (از IE 10 به بعد مثلاً) که امکان [Ajax upload](#) توکار به همراه گزارش درصد آپلود را بدون نیاز به فلش یا سیلورلایت، دارند.

در این بین راه حل دیگری نیز وجود دارد که با تمام مرورگرها سازگار است؛ اما تنها گزارش درصد آپلود را توسط آن نخواهیم داشت. در اینجا به صورت پویا یک IFrame مخفی در صفحه تشکیل می‌شود، مقادیر معمولی فرم (تمام المان‌ها، منهای file) به صورت Ajax ایی به سرور ارسال خواهند شد. المان file آن در این IFrame مخفی، به صورت معمولی به سرور Postback می‌شود. البته کاربر در این بین چیزی را مشاهده یا احساس نخواهد کرد و تمام عملیات از دیدگاه او Ajax ایی به نظر می‌رسد. برای انجام اینکار تنها کافی است از افزونه‌ی [AjaxFileUpload](#) استفاده کنیم که در ادامه نحوه‌ی استفاده از آن را بررسی خواهیم کرد.

پیشنیازها

در ادامه فرض بر این است که افزونه‌ی [AjaxFileUpload](#) را دریافت کرده و به فایل Layout برنامه افزوده‌اید:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>@ViewBag.Title - My ASP.NET Application</title>
  <link href="~/Content/Site.css" rel="stylesheet" type="text/css" />
</head>
<body>
  <div>
    @RenderBody()
  </div>

  <script src="~/Scripts/jquery-1.11.1.min.js"></script>
  <script src="~/Scripts/jquery.unobtrusive-ajax.js"></script>
  <script src="~/Scripts/ajaxfileupload.js"></script>
  @RenderSection("Scripts", required: false)
</body>
</html>
```

مدل، کنترلر و View برنامه

مدل برنامه مشخصات یک محصول است:

```
namespace MVCAjaxFormUpload.Models
{
    public class Product
    {
        public int Id { set; get; }
        public string Name { set; get; }
    }
}
```

کنترلر آن از سه متد تشکیل شده‌است:

```
using System.Threading;
using System.Web;
```

```

using System.Web.Mvc;
using MVCAjaxFormUpload.Models;

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

        [HttpPost]
        public ActionResult Index(Product product)
        {
            var isAjax = this.Request.IsAjaxRequest();

            return Json(new { result = "ok" }, JsonRequestBehavior.AllowGet);
        }

        [HttpPost]
        public ActionResult UploadFiles(HttpPostedFileBase image1, int id)
        {
            var isAjax = this.Request.IsAjaxRequest();
            Thread.Sleep(3000); // شبیه سازی عملیات طولانی
            return Json(new { FileName = "/Uploads/filename.ext" }, "text/html",
                JsonRequestBehavior.AllowGet);
        }
    }
}

```

Index اول، کار نمایش صفحه‌ی ارسال اطلاعات را انجام خواهد داد.
 Index دوم کار پردازش Ajax ایی اطلاعات ارسالی به سرور را به عهده دارد. HttpPost آن Ajax ایی است.
 متد UploadFiles، کار پردازش اطلاعات ارسالی از طرف IFrame مخفی را انجام می‌دهد. HttpPost آن معمولی است.

و کدهای View این مثال نیز به شرح زیر است:

```

@model MVCAjaxFormUpload.Models.Product
@{
    ViewBag.Title = "Index";
}

<h2>Ajax Form Upload</h2>

@using (Ajax.BeginForm(actionName: "Index",
    controllerName: "Home",
    ajaxOptions: new AjaxOptions { HttpMethod = "POST" },
    routeValues: null,
    htmlAttributes: new { id = "uploadForm" })))
{
    <label>Name:</label>
    @Html.TextBoxFor(model => model.Name)
    <br />
    <label>Image:</label>
    <br />
    <input type="file" name="Image1" id="Image1" />
    <br />
    <input type="submit" value="Submit" />
    
}

@section Scripts
{
    <script type="text/javascript">
        $(function () {
            $('#uploadForm').submit(function () {
                $('#loading').show();
                $.ajaxFileUpload({
                    url: "@Url.Action("UploadFiles", "Home")", // مسیری که باید فایل به آن ارسال شود
                    secureuri: false,
                    fileElementId: 'Image1', // آی دی المان ورودی فایل
                    dataType: 'json',
                    data: { id: 1, data: 'test' }, // اطلاعات اضافی در صورت نیاز
                    success: function (data, status) {
                        $('#loading').hide();
                    }
                });
            });
        });
    </script>
}

```

```

        if (typeof (data.FileName) != 'undefined') {
            alert(data.FileName);
        }
    },
    error: function (data, status, e) {
        $("#loading").hide();
        alert(e);
    }
});
});
});
</script>
}

```

فرمی که توسط Ajax.BeginForm تشکیل شده است، یک فرم معمولی Ajax ایی است و نکته‌ی جدیدی ندارد. تنها در آن یک المان ارسال فایل قرار گرفته است و همچنین Id آن را نیز جهت استفاده توسط jQuery مشخص کرده ایم.

در ادامه نحوه‌ی فعال سازی ajaxFileUpload را دقیقاً در زمان submit فرم، مشاهده می‌کنید. در اینجا url آن به اکشن متدی که اطلاعات المان file را باید دریافت کند، اشاره می‌کند. fileElementId آن مساوی Id المان فایل فرم Ajax ایی صفحه است. از قسمت data جهت ارسال اطلاعات اضافه‌تری به اکشن متد UploadFiles استفاده می‌شود. سایر قسمت‌های آن نیز مشخص هستند. اگر عملیات موفقیت آمیز بود، success آن و اگر خیر، error آن اجرا می‌شوند.

فقط باید دقت داشت که content type دریافتی توسط آن باید text/html باشد، که این مورد در اکشن متدهای کنترلر مشخص هستند.

به این ترتیب دیگر کاربر نیازی ندارد ابتدا یکبار بر روی دکمه‌ی دومی کلیک کرده و فایل را ارسال کند و سپس بار دیگر بر روی دکمه‌ی submit فرم کلیک نماید. هر دو کار توسط یک دکمه انجام می‌شوند.

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

[MVCAjaxFormUpload.zip](#)

نظرات خوانندگان

نویسنده: M.Shakeri

تاریخ: ۱۰:۱۸ ۱۳۹۳/۰۴/۲۲

ممنون جناب نصیری بابت مطلبی که قرار دادین. من چند ماه پیش از این افزونه استفاده کردم. میخواستم فرمی از اطلاعات رو به همراه فایل مربوطه بصورت ajax ارسال کنم اما جدول آپلود فایل من جدا بود و به ازای هر فرم کلید خارجی اطلاعات مربوطه در جدول آپلود ثبت می‌شد. مسئله اینجا بود که من باید اول فرم رو ثبت می‌کردم و سپس کلید خارجی اون فرم رو در جدول آپلود قرار میدادم اما مشکل من این بود که اول فایل آپلود میشد و بعد فرم ثبت میشد. یعنی اول رویداد submit کلاینت اجرا میشد سپس Ajax.BeginForm. برای این مسئله راهکاری هست؟

نویسنده: وحید نصیری

تاریخ: ۱۲:۴۵ ۱۳۹۳/۰۴/۲۲

در حین تعریف فرم، OnSuccess را به یک متد جاوا اسکریپتی که قرار است پس از اجرای موفقیت آمیز ارسال اطلاعات Ajax ایی به سرور اجرا شود، مقدار دهی کنید:

```
@using (Ajax.BeginForm(actionName: "Index",
    controllerName: "Home",
    ajaxOptions: new AjaxOptions
    {
        HttpMethod = "POST",
        OnSuccess = "doUpload(data, status, xhr)"
    },
    routeValues: null,
    htmlAttributes: new { id = "uploadForm" }))
{
```

این متد یک چنین امضایی را باید داشته باشد:

```
function doUpload(data, status, xhr) {
    alert(data.result);
    // مابقی کدهای آپلود فایل
```

به عبارتی می‌توان Id رکورد insert شده را در اینجا دریافت (توسط data) و سپس به کمک اطلاعات اضافی ارسال به سرور افزونه‌ی ارسال فایل، به اکشن متد ذخیره فایل ارسال کرد.

جهت تکمیل بحث

- OnBegin - xhr
- OnComplete - xhr, status
- OnSuccess - data, status, xhr
- OnFailure - xhr, status, error

پارامترهای AjaxOptions یک چنین اطلاعاتی را از سرور دریافت می‌کنند که نمونه‌ای از آن در متد doUpload فوق استفاده شد.