

یک صفحه شلوغ و سنگین را در نظر بگیرید. برای مثال قرار است ابتدا مطلب خاصی در سایت نمایش یابد و سپس ادامه صفحه که شامل انبوهی از لیست نظرات مرتبط با آن مطلب است به صورت غیرهمزمان و Ajax ایی بدون توقف پردازش صفحه، در فرصتی مناسب از سرور دریافت و به صفحه اضافه گردد (به روز رسانی قسمتی از صفحه در فرصت مناسب). در این حالت چون نمایش اولیه صفحه سریع صورت می‌گیرد، کاربر نهایی آنچنان احساس کند بودن بازکردن صفحات سایت را نخواهد داشت. در ادامه نحوه پیاده سازی این روش را به کمک jQuery Ajax بررسی خواهیم کرد.

## مدل و کنترلر برنامه

```
namespace jQueryMvcSample07.Models
{
    public class BlogPost
    {
        public int Id { set; get; }
        public string Title { set; get; }
        public string Body { set; get; }
    }
}
```

```
using System.Web.Mvc;
using System.Web.UI;
using jQueryMvcSample07.Models;
using jQueryMvcSample07.Security;

namespace jQueryMvcSample07.Controllers
{
    public class HomeController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            return View(); // نمایش یک منوی ساده در ابتدای کار
        }

        [HttpGet]
        public ActionResult ShowSynchronous()
        {
            var model = getModel();
            return View(model); // نمایش همزمان
        }

        private static BlogPost getModel()
        {
            // شبیه سازی یک عملیات طولانی
            System.Threading.Thread.Sleep(3000);
            var model = new BlogPost
            {
                Title = "... عنوان ...",
                Body = "... مطلب..."
            };
            return model;
        }

        [HttpGet]
        public ActionResult ShowAsynchronous()
        {
            return View(); // نمایش ابتدایی صفحه
        }

        [HttpPost]
        [AjaxOnly]
        [OutputCache(Location = OutputCacheLocation.None, NoStore = true)]
        public ActionResult RenderAsynchronous()
        {
        }
    }
}
```

```
//دریافت اطلاعات به صورت غیرهمزمان
var model = getModel();
return PartialView(viewName: "_Post", model: model);
}
}
```

مدل برنامه، بیانگر ساختار اطلاعات مطلبی است که قرار است نمایش یابد.  
در کنترلر Home، ابتدا اکشن متد Index آن فراخوانی شده و در این حالت دو لینک زیر نمایش داده می‌شوند:

```
@{
    ViewBag.Title = "Index";
}
<h2>
    نمایش اطلاعات به صورت همزمان و غیرهمزمان
</h2>
<ul>
    <li>
        @Html.ActionLink(linkText: "نمایش همزمان", actionName: "ShowSynchronous", controllerName:
"Home")
    </li>
    <li>
        @Html.ActionLink(linkText: "نمایش غیر همزمان", actionName: "ShowAsynchronous", controllerName:
"Home")
    </li>
</ul>
```

لینک اول، به اکشن متد ShowSynchronous اشاره می‌کند و لینک دوم به اکشن متد ShowAsynchronous.  
در هر دو صفحه نهایتاً از یک Partial View به نام Post.cshtml استفاده خواهد شد:

```
@model jQueryMvcSample07.Models.BlogPost
<fieldset>
    <legend>@Model.Title</legend>
    @Model.Body
</fieldset>
```

زمانیکه کاربر بر روی لینک نمایش همزمان کلیک می‌کند، به صفحه زیر هدایت می‌شود:

```
@model jQueryMvcSample07.Models.BlogPost
@{
    ViewBag.Title = "ShowSynchronous";
}
<h2>نمایش همزمان</h2>
@{ Html.RenderPartial("_Post", Model); }
```

این صفحه، یک صفحه متداول است و اطلاعات آن دقیقاً در زمان نمایش صفحه اخذ شده و چون در اینجا از یک Sleep عمدی برای تولید اطلاعات استفاده گردیده است، نمایش آن حداقل سه ثانیه طول خواهد کشید.  
در حالیکه کاربر بر روی لینک نمایش غیرهمزمان کلیک می‌کند، صفحه زیر را مشاهده خواهد کرد:

```
@{
    ViewBag.Title = "ShowAsynchronous";
    var loadInfoUrl = Url.Action(actionName: "RenderAsynchronous", controllerName: "Home");
}
<h2>
    نمایش غیر همزمان
</h2>
<div id="info" align="center">
</div>
<div id="progress" align="center" style="display: none">
    <br />
    
</div>
@section JavaScript
{
    <script type="text/javascript">
        $(function () {
            $("#progress").css("display", "block");
            $.ajax({
```

```
type: "POST",
url: '@loadInfoUrl',
complete: function (xhr, status) {
    var data = xhr.responseText;
    if (xhr.status == 403) {
        window.location = "/login";
    }
    else if (status === 'error' || !data || data == "nok") {
        alert('خطایی رخ داده است');
    }
    else {
        $("#progress").css("display", "none");
        $("#info").html(data);
    }
}
});
});
</script>
}
```

نمایش ابتدایی این صفحه بسیار سریع است. در ابتدای کار progress bar ایی فعال شده و سپس از طریق jQuery Ajax درخواست دریافت اطلاعات رندر شده اکشن متدی به نام RenderAsynchronous به سرور ارسال می‌شود. چون عملیات Ajax غیرهمزمان است، کاربر نیازی نیست تا رندر شدن کامل صفحه ابتدا صبر کند و سپس کل صفحه به او نمایش داده شود. در اینجا ابتدا صفحه به صورت کامل نمایان شده و سپس درخواستی Ajax ایی به سرور ارسال می‌گردد. در پایان عملیات، Partial View یاد شده رندر گردیده و در div ایی با id مساوی info نمایش داده می‌شود. به این ترتیب می‌توان حس سریع بودن سایت را زمانیکه قسمتی از صفحه نیاز به زمان بیشتری برای نمایش اطلاعات دارد، به کاربر منتقل کرد.

دریافت پروژه کامل این قسمت

[jQueryMvcSample07.zip](#)

## نظرات خوانندگان

نویسنده: رشوند  
تاریخ: ۱۸:۲۲ ۱۳۹۳/۰۶/۰۶

با سلام  
در قسمتی از پروژه کاملاً همانند شما عمل کردم و حتی فایل پروژه را دانلود و نگاه انداختم کاملاً شبیه هم بود، ولی متأسفانه در هنگام برگرداندن اطلاعات اکشن RenderAsynchronous، برنامه به layout\_ رفته، موارد آنجا را هم (از جمله لینک ها، توضیحات و ...) در خروجی (div info) اضافه می‌کند... اگر این اشکال متداول است لطفا راهنمایی بفرمایید که چه نکته ای را رعایت نکرده ام.

نویسنده: وحید نصیری  
تاریخ: ۱۹:۰۶ ۱۳۹۳/۰۶/۰۶

- بررسی کنید آیا return PartialView دارید یا return View ؟ حالت return View فایل layout را هم لحاظ می‌کند.
- همچنین امکان نال تعریف کردن layout به صورت صریح هم وجود دارد:

```
@{  
Layout = null;  
}
```