

## Ajax : تحولی بزرگ در عرصه وب ( بخش هفتم )

آنچه تاکنون گفته شده است :

<a href="#">بخش اول</a>	تاثیر متقابل وب و نرم افزار بر یکدیگر
<a href="#">بخش دوم</a>	Ajax و فناوری های مرتبط با آن
<a href="#">بخش سوم</a>	بررسی نمونه برنامه های مبتنی بر Ajax
<a href="#">بخش چهارم</a>	برنامه نویسی غیرهمزمان در برنامه های وب
<a href="#">بخش پنجم</a>	بررسی معماری ASP. NET Ajax
<a href="#">بخش ششم</a>	بررسی مدل پیاده سازی با محوریت سرویس گیرنده و مدل پیاده سازی با محوریت سرویس دهنده .

تاکنون و در طی شش مقاله با مواردی نظیر شی XMLHttpRequest ، برخی الگوهای Ajax استفاده شده در زمان پیاده سازی برنامه های وب قدرتمند با توانایی ایجاد تعامل بیشتر ، معماری ASP.NET AJAX و سناریوهای مختلف پیاده سازی آشنا شدیم .

در این بخش با استناد به موارد فوق با نحوه استفاده از این فریمورک آشنا خواهیم شد و صفحات وب مبتنی بر Ajax را با تبعیت از مدل پیاده سازی با محوریت سرویس دهنده ایجاد خواهیم کرد .

### مدل پیاده سازی با محوریت سرویس دهنده

همانگونه که در [بخش ششم](#) اشاره گردید ، در این مدل ، منطق برنامه و اکثر عناصر بخش رابط کاربر بر روی سرویس دهنده باقی می ماند و در ادامه ، تغییرات مورد نیاز رابط کاربر برای برنامه مرورگر ارسال می گردد ( در مقابل این که تغییرات از طریق اجرای اسکریپت های سمت سرویس گیرنده ایجاد شوند) . برای آشنائی عملی با مدل فوق ، در ادامه یک نمونه مثال ساده و در عین حال کاربردی را با یکدیگر دنبال می نمائیم .

### مثال : ایجاد یک صفحه مبتنی بر Ajax با تمرکز بر روی پتانسیل های سمت سرویس دهنده فریمورک

#### Ajax ASP. NET

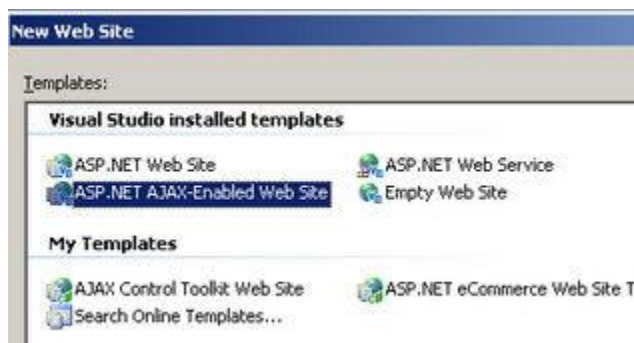
در این مثال می خواهیم تعداد مقالاتی را که بر روی سایت سخا روش و در هر یک از گروه های مختلف منتشر شده است ، به اطلاع ملاقات کنندگان سایت برسانیم . برای سادگی کار ، فرض می شود که تعداد مقالات منتشر شده بر روی ساعت از طریق بانک اطلاعاتی بازبایی نخواهد شد و در مقابل از یک کلاس با نام Maghalat برای بازبایی تعداد مقالات منتشر شده در هر گروه استفاده خواهیم کرد . کلاس فوق دارای صرفاً یک متد با GetNumberOfMaghalat است که وظیفه آن برگرداندن تعداد مقالات منتشر شده در هر گروه است . پس از آشنائی اولیه با صورت مسئله ، مراحل زیر را برای پیاده سازی یک صفحه وب مبتنی بر Ajax با محوریت سرویس دهنده ، دنبال می نمائیم .

### مرحله اول : ایجاد سایت

اولین مرحله در پیاده سازی هر نوع برنامه وب مبتنی بر ASP. NET ، ایجاد وب سایت اولیه است . بدین منظور از

برنامه ویژوال استودیو نسخه های 2005 و یا 2008 استفاده کرده ( و یا نسخه Visual Web Developer ) و برای وب سایت خود تمپلیت ASP.NET AJAX-Enabled Web Site را انتخاب می کنیم . بدین ترتیب ، یک وب سایت با قابلیت استفاده از اسمبلی ASP.NET AJAX ( با نام System.Web.Extensions.dll ) از طریق GAC ( برگرفته شده از Global Assembly Cache ) ایجاد می گردد . این کار همچنین باعث ایجاد یک فایل web.config پیچیده می گردد که شامل تنظیمات اضافه دیگر برای ارتباط با فریمورک ASP.NET AJAX است .

شکل 1 ، نحوه انتخاب تمپلیت ASP.NET AJAX-Enabled Web Site را در زمان ایجاد یک وب سایت نشان می دهد .



شکل 1 : انتخاب تمپلیت ASP.NET AJAX-Enabled Web Site در زمان ایجاد یک وب سایت

### مرحله دوم : طراحی و پیاده سازی کلاس Maghalat

Maghalat یک کلاس ساده است که دارای صرفاً یک متد با نام GetNumberOfMaghalat است که گروه مقاله را به عنوان پارامتر ورودی گرفته و تعداد مقالات منتشر شده در آن گروه را برمی گرداند . بدین منظور از یک ساختار Select Case استفاده شده است تا بر اساس مقدار پارامتر ورودی ، یک عدد را بطور ایستا برگرداند ( برای سادگی کار نام گروه مقاله و تعداد آن مستقیماً در کد درج شده اند ) .

شکل 2 ، ساختار کلاس Maghalat را نشان می دهد .



شکل 2 : ساختار کلاس Maghalat

کد زیر ، کلاس فوق به همراه متد مربوطه را نشان می دهد .

```

کلاس Maghalat.VB
Public Class Maghalat

```

```

Public Shared Function GetNumberOfMaghalat(ByVal
ArticleGroup As String) As Integer
Dim Count As Integer = 0
Select Case ArticleGroup
    Case "Software"
        Count = 11
    Case "Hardware"
        Count = 12
    Case "Security"
        Count = 13
    Case "Network"
        Count = 14
    Case "other"
        Count = 15
End Select
GetNumberOfMaghalat = Count
End Function
End Class

```

### مرحله سوم : ایجاد یک صفحه وب ASP.NET

در زمان ایجاد وب سایت ، بطور اتوماتیک یک صفحه Default.aspx نیز ایجاد می گردد . کد اولیه این صفحه به صورت زیر است .

```

<%@ Page Language="VB" AutoEventWireup="true" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" dir="rtl">
<head runat="server">
    <title>شماره یک تست</title>
</head>
<body>
    <form id="form1" runat="server">
        <asp:ScriptManager ID="ScriptManager1" runat="server" />
        <div>
        </div>
    </form>

```

```
</body>
</html>
```

تنها تفاوت صفحه فوق با سایر صفحات Default.aspx ایجاد شده توسط ویژوال استودیو ، اضافه شدن کنترل Ajax ScriptManager است . همانگونه که قبلا" اشاره گردید ، کنترل فوق به عنوان مغز متفکر یک صفحه مبتنی بر Ajax ایفای وظیفه می نماید . کنترل فوق ، مسئولیت توزیع کدهای سمت سرویس گیرنده برای مرورگر و مدیریت بهنگام سازی جزئی ( نه تمامی صفحه ) صفحه را برعهده دارد .

پس از ایجاد اولیه صفحه وب ، از کنترل های دیگری برای طراحی بخش رابط کاربر متناسب با خواسته های موجود استفاده می کنیم . در این مثال خاص از یک کنترل ListBox جهت نمایش گروه مقاله با قابلیت AutoPostBack و یک کنترل Label به منظور نمایش نتایج استفاده شده است .

```
<%@ Page Language="VB" AutoEventWireup="true" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" dir="rtl">
<head runat="server">
  <title> شماره یک تست </title>
</head>
<body>
  <form id="form1" runat="server">
    <asp:ScriptManager ID="ScriptManager1" runat="server" />
    <div style="text-align: center">
      <table>
        <tr>
          <td colspan="3">
            <strong>&nbsp; Ajax یک صفحه وب مبتنی بر ایجاد<br />
            (( ASP.NET پیاده سازی با محوریت پتانسیل های سمت سرویس دهنده فریمورک ))
            Ajax )) </strong>
          <br />
        </td>
      </tr>
      <tr>
        <td style="width: 150px">
          لطفا" یک گروه مقاله را انتخاب نمائید
        </td>
        <td align="right" colspan="2" dir="rtl">
```

```

<asp:ListBox ID="ArticleGroup" runat="server"
AutoPostBack="True" Font-Names="Tahoma"
Font-Size="Small" ForeColor="#C04000">
    <asp:ListItem> نرم افزار </asp:ListItem>
    <asp:ListItem> سخت افزار </asp:ListItem>
    <asp:ListItem> امنیت اطلاعات </asp:ListItem>
    <asp:ListItem> شبکه </asp:ListItem>
    <asp:ListItem> سایر </asp:ListItem>
</asp:ListBox></td>
</tr>
<tr>
    <td style="width: 150px"> </td>
    <td style="width: 100px"> </td>
    <td style="width: 131px"> </td>
</tr>
<tr>
    <td colspan="3" style="height: 18px">
        <asp:Label ID="LblFinalResult" runat="server" Font-
Names="Tahoma" Font-Size="Small"
        ForeColor="Black" Text="Label"></asp:Label></td>
    </tr>
</table>
</span>
</div>
</form>
</body>
</html>

```

از `ListBox` برای نمایش لیست گروه مقالات استفاده شده است . مقدار خصلت `AutoPostBack` کنترل فوق ، `True` در نظر گرفته شده است تا بلافاصله پس از انتخاب یکی از آیتم های موجود در لیست ، یک `PostBack` به سمت سرور ارسال دهنده تحقق یابد . این کار از طریق رویداد `SelectedIndexChanged` محقق خواهد شد که متعاقب آن رویتین `ArticleGroup_SelectedIndexChanged` فراخوانده می شود . در انتهای صفحه از یک کنترل `Label` استفاده شده است تا به کمک آن بتوان نتایج را در خروجی نمایش داد .

از طریق کد نوشته شده در رویتین `ArticleGroup_SelectedIndexChanged` ، انتخاب کاربر تشخیص و متد مربوط به کلاس `maghalat` صدا زده می شود تا پس از برگرداندن نتایج ، حاصل کار از طریق کنترل `Label` نمایش داده شود . کد رویتین `ArticleGroup_SelectedIndexChanged` در جدول زیر نشان داده شده است .



```

<ContentTemplate >
    <asp:ListBox ID="ArticleGroup" runat="server" AutoPostBack="True"
Font-Names="Tahoma"
        Font-Size="Small" ForeColor="#C04000">
        <asp:ListItem> نرم افزار </asp:ListItem>
        <asp:ListItem> سخت افزار </asp:ListItem>
        <asp:ListItem> امنیت اطلاعات </asp:ListItem>
        <asp:ListItem> شبکه </asp:ListItem>
        <asp:ListItem> سایر </asp:ListItem>
    </asp:ListBox></td>
</tr>
<tr>
    <td style="width: 150px"> </td>
    <td style="width: 100px"> </td>
    <td style="width: 131px"> </td>
</tr>
<tr>
    <td colspan="3" style="height: 18px">
        <asp:Label ID="LbfFinalResult" runat="server" Font-
Names="Tahoma" Font-Size="Small"
            ForeColor="Black" Text="Label"></asp:Label></td>
</tr>
</ContentTemplate>
</asp:UpdatePanel>
...

```

به صورت پیش فرض ، محتویات موجود بین تگ ContentTemplate مربوط به کنترل UpdatePanel در زمان بروز یک postback غیرهمزمان بطور اتوماتیک بهنگام خواهند شد. Postback فوق بطور غیرهمزمان اتفاق می افتد و رفتار آن با postback عادی متفاوت است . ( در postback عادی ، یک درخواست برای سرویس دهنده ارسال می گردد تا پس از انجام پردازش های ضروری در سمت سرویس دهنده ، رابط کاربر جدید برای مرورگر ارسال گردد ) . و اما یک سوال مهم که ممکن است در ذهن شما مطرح شده باشد . یک postback غیرهمزمان چیست ؟ اکثر پیاده کنندگان برنامه های وب به کمک فناوری ASP.NET صرفاً با یک نوع postback آشنا می باشند . با بکارگیری UpdatePanel ، صفحه روال طبیعی حیات خود را طی می نماید ، ولی PostBack به عنوان یک postBack غیرهمزمان شناخته می شود . که در آن از روش های هوشمندانه تری استفاده می گردد و صفحه با مدلی متفاوت تر در حین چرخه حیات خود پردازش می گردد .

در ادامه ، همزمان با اجرای صفحه و انتخاب یک گروه مقاله ، بخش رابط کاربر بطور اتوماتیک و بدون نیاز به بازخوانی

کامل صفحه بهنگام خواهد شد . بطور خلاصه ، با اضافه کردن یک مجموعه اندک از کنترل های سرویس دهنده بر روی صفحه ، از بازخوانی مجدد تمامی صفحه ممانعت بعمل آمده و همچنین در تعامل کاربو با برنامه وقفه ای ایجاد نخواهد شد .

### مرحله پنجم : بهینه سازی خروجی و ضرورت استفاده از کنترل UpdateProgress

در صورتی که با یک خط سرعت پائین برنامه فوق اجراء گردد ، مشاهده خواهیم کرد که زمان نسبتاً زیادی بطول خواهد انجامید تا نتایج نمایش داده شود . این موضوع ( تاخیر محسوس در مشاهده خروجی کار ) باعث می شود کاربری که برنامه را اجراء کرده است ، نگران عملکرد صحیح برنامه باشد و اینگونه برداشت کند که برنامه دچار اختلال شده است .

قبل از معرفی Ajax ، برای صفحاتی که بازخوانی مجدد می گردیدند از علامتی استفاده می شد که کاربران متوجه این موضوع می شدند که فرآیندی در حال انجام است و یا درخواست آنان مورد پذیرش قرار گرفته شده است . هم اینک با توقف postback طبیعی ، کاربران متوجه این موضوع نمی شوند که عملیاتی در شرف انجام است تا این که فرآیند مورد نظر به اتمام برسد ( از علامتی برای نشان دادن این که فرآیندی در حال انجام است ، استفاده نمی شود ) . در چنین مواردی ، کاربران نیازمند یک فیدبک دیداری ( ویژوال ) هستند تا نسبت به انجام یک فرآیند در پس زمینه آگاه گردند .

کنترل UpdateProgress یک راه حل جهت مشکل فوق را ارائه می نماید . هدف کنترل فوق ، نشان دادن یک نشانه دیداری به کاربران در زمان بروز یک postback غیرهمزمان است . بدین منظور می توان کد زیر را به انتهای صفحه اضافه نمود .

```
<asp:UpdateProgress ID="UpdateProgress1" runat="server">
  <ProgressTemplate >
     ... بارگذاری در حال
  </ProgressTemplate>
</asp:UpdateProgress>
```

پس از اجراء برنامه ( با لحاظ کردن کنترل UpdateProgress ) ، پس از انتخاب یک گروه مقاله توسط کاربر یک نشانه دیداری نمایش داده خواهد شد . شکل 4 ، خروجی برنامه را به همراه نشانه دیداری نشان می دهد .





فریمورک ASP. NET Ajax آشنا شدیم . بدین منظور یک نمونه مثال ساده را بررسی کردیم که در آن از کنترل های UpdatePanel و UpdateProgress به منظور نشان دادن قابلیت Ajax در صفحات ASP.NET استفاده شده بود . در بخش بعد با نحوه ایجاد یک صفحه مبتنی بر Ajax با تمرکز بر روی پتانسیل های سمت سرویس گیرنده فریمورک ASP. NET Ajax ، آشنا خواهیم شد .