

بسم الله الرحمن الرحيم

## آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

### ASP.NET در AJAX آموزش کاربرد کنترل

مدرس : مهندس افشین رفوآ

#### آموزش کاربرد کنترل AJAX در ASP.NET

**AJAX** سرواژه ی **Asynchronous JavaScript and XML** می باشد. ایجکس در واقع یک فناوری یا تکنولوژی **-cors**

**platform** (به آن دسته از نرم افزارها گفته می شود که در چندین بستر اجرای نرم افزار رایانه ای یا همان **platform**

قابل اجرا هستند) که زمان پاسخ دهی (**response time**) را به گونه ی قابل توجهی کاهش می دهد. به عبارت دیگر ایجکس مجموعه ای از استانداردها و فناوری های ویژه ی وب است که با بهره گیری از آنها می توان برنامه هایی مبتنی بر وب نوشت که به آسانی با کاربران تعامل (**interaction**) داشته باشند. با استفاده از این فناوریها و با کمک انتقال تکه های کوچک داده و اطلاعات از **Server**، صفحات وب از حالت منفعل خارج می شوند و واکنشهایی مناسب با رویدادها انجام می دهند. با استفاده از این معماری، صفحات وب تعامل بسیار خوبی با کاربران خواهند داشت. علاوه بر آن، مهم ترین مزیت این معماری این است که دیگر برای انجام هر کاری، احتیاج نیست صفحه وب دوباره بارگذاری شود.

کنترل های سمت سرور ایجکس (**AJAX server controls**) به صفحه ی وب اسکریپت اضافه می شوند که توسط

مروگر پردازش شده، سپس اجرا می گردد.

با این حال درست مانند دیگر کنترل های سمت سرور **ASP.NET**، کنترل های سمت سرور ایجکس (**AJAX server controls**) نیز می توانند دارای متدها و **event handler** های (مدیریت کننده ی رخداد) مرتبط با آن متدها باشند که در سمت سرور (**server side**) پردازش می شوند.

**control toolbox** در محیط برنامه نویسی یکپارچه ی **visual studio** شامل مجموعه ای از کنترل ها می باشد که **'AJAX**

**Extensions'** یا به عبارتی دیگر افزونه های ایجکس خوانده می شوند.



## کنترل ScriptManager

کنترل **ScriptManager** مهم ترین control تلقی می گردد، به گونه ای که کنترل های دیگر موجود در صفحه برای عملکرد صحیح به وجود **ScriptManager** احتیاج دارند.

کنترل **ScriptManager** ، وظیفه مدیریت پردازش ها (**script**) بر روی صفحات **ASP.Net** ای که امکان و قابلیت **Ajax** آنها فعال سازی شده است را بر عهده دارد . به صورت پیش فرض، کنترل **ScriptManager** اسکریپت های به وقوع پیوسته متعلق به عملیات ایچکس در صفحه را ، با مجموعه سایر پردازش های (**script**) صفحه مرتبط می کند . این کار امکان استفاده از قابلیت های مرورگر در سمت سرور (گیرنده (**client-side**)) و بروز رسانی بخش های مختلف صفحه بدون بارگذاری مجدد یا تجدید کامل آن را می دهد.

نحوه ی کلی نگارش (**syntax**) کنترل مذکور به ترتیب زیر می باشد:

```
<asp:ScriptManager ID="ScriptManager1" runat="server">  
</asp:ScriptManager>
```

چنانچه وب سایتی ایجاد کنید که تکنولوژی ایچکس در آن بکار گرفته شده (**Ajax Enabled site**) یا یک فرم وب ایچکس (**AJAX Web Form**) از طریق پنجره ی معاوره (**dialog box**) به آن اضافه گردیده، در آن صورت مشاهده خواهید که فرم وب شما به صورت خودکار کنترل **script manager** را در خود خواهد داشت. کنترل **ScriptManager** ترتیب کلیه ی اسکریپت های سمت سرور (گیرنده (**client-side script**)) را به ازای تمامی کنترل های سمت سرور (دهنده (**server side control**)) می دهد.

## کنترل UpdatePanel

**UpdatePanel control** در حقیقت یک کنترل دربرگیرنده (**container control**) است که از کلاس **Control** مشتق شده است. کنترل ذکر شده به مثابه ی یک ظرف (نگهدارنده) برای کنترل های فرزند (**child control**) عمل می کند. این کنترل همچنین دارای رابط (**interface**) مختص خود نیست. هنگامی که یکی از کنترل های موجود در آن، در صدد

انجام و راه اندازی **postback** برمی آید، **UpdatePanel** مداخله کرده و باعث می شود ارسال (**post**) به صورت ناهمزمان<sup>1</sup> صورت گرفته (آغاز شده) و تنها قسمت مورد نظر صفحه بروز رسانی شود.

کنترل **UpdatePanel** این امکان را به برنامه نویس می دهد، تا برنامه های تحت وب کارآمد و با رابط کاربری همانند برنامه های رایانه های رومیزی بسازید . به وسیله این کنترل می توانید قسمت یا بخش هایی از صفحه را که می خواهید اطلاعات آن تغییر کرده را بروز رسانی کرده و بدین وسیله از تجدید یا بارگذاری دوباره و **Postback** کل صفحه به **server** جلوگیری کنید.

**UpdatePanel control**، با جدا سازی بخش مورد نظر از صفحه، حین به روز رسانی خود ، فقط آن قسمت های مرتبط را آپدیت کرده و مانع از بارگذاری مجدد یا بازسازی و **Postback** کل صفحه ی مورد نظر می شود.

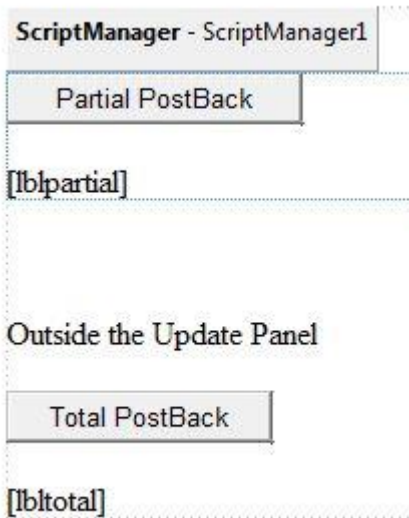
برای مثال اگر یک دکمه ی موجود در کنترل نام برده کلیک شود، تنها کنترل های داخل **update panel** تحت تاثیر قرار گرفته و کنترل های موجود در بخش های دیگر صفحه هیچ تغییری نمی کنند. به این فرایند **postback** جزئی / بخشی یا ناهمگام (**asynchronous / partial postback**) گفته می شود.

مثال:

ابتدا یک فرم وب ایجکس (**AJAX web form**) به **application** (برنامه ی کاربردی) خود اضافه کنید. این فرم به صورت پیش فرض کنترل **Script Manger** را در خود دارد. سپس یک **update panel** داخل آن قرار دهید (درج کنید). حال یک کنترل **button** به همراه یک کنترل **label** داخل **update panel control** جای گذاری کنید. بار دیگر یک کنترل **button** به همراه کنترل **lable** اضافه می کنیم اما این بار خارج از **panel** مربوطه.

---

<sup>1</sup> asynchronous



:Source file

```
<form id="form2" runat="server">
  <div>
    <asp:ScriptManager ID="ScriptManager2" runat="server" />
  </div>
  <asp:UpdatePanel ID="UpdatePanel2" runat="server">
    <ContentTemplate>
      <asp:Button ID="Button1" runat="server" onclick="btnpartial_Click" Text="Partial PostBack"/>
      <br />
      <br />
      <asp:Label ID="Label1" runat="server"></asp:Label>
    </ContentTemplate>
  </asp:UpdatePanel>
  <p> </p>
  <p>Outside the Update Panel</p>
  <p>
    <asp:Button ID="Button2" runat="server" onclick="btntotal_Click" Text="Total PostBack" />
  </p>
  <asp:Label ID="Label2" runat="server"></asp:Label>
</form>
```

هر دو کنترل های button دارای کد یکسان برای event handler می باشند:

```
string time = DateTime.Now.ToLongTimeString();
lblpartial.Text = "Showing time from panel" + time;
lbltotal.Text = "Showing time from outside" + time;
```

همان طور که مشاهده می کنید زمانی که صفحه اجرا می شود، اگر روی دکمه ی **total post back** کلیک شود، زمان بر

روی هر دو **label** بروز رسانی می شود. اما اگر دکمه ی **partial post back** از **panel** موردنظر کلیک شود، فقط **label**

موجود در کنترل **update panel** بروز رسانی می شود.

PartialPostBack

Showing time from panel11:51:31

Outside the Update Panel

TotalPostBack

Showing time from outside11:18:10

یک صفحه می تواند دربردارنده ی چندین **panel** باشد که هر یک خود شامل کنترل های دیگر مانند **grid** بوده و بخش های مختلف داده را نمایش دهد.

هنگامی که یک **total post back** رخ می دهد، محتوا **update panel (content)** به صورت پیش فرض بروز رسانی می شود. این حالت **(mode)** پیش فرض را می توان با اصلاح خاصیت **UpdateMode (property)** تغییر داد.

### خاصیت های کنترل **UpdatePanel**

جدول زیر برخی از خاصیت های کنترل مذکور را برای شما لیست کرده است:

خاصیت	شرح
ChildrenAsTriggers	مقداری بازیابی کرده یا تنظیم می کند که نشانگر این است که آیا postback های آمده از کنترل های فرزند (child control) محتوای panel را بروز رسانی می کنند یا خیر. چنانچه مایلید postback های آمده از child control ها باعث بروز رسانی محتویات panel شوند، این خاصیت را روی مقدار true تنظیم کنید.
ContentTemplate	این خاصیت قالبی (template) که محتویات کنترل UpdatePanel را تعریف می کند، بازیابی یا مقداردهی می کند.
ContentTemplateContainer	این property یک شیء control بازیابی می کند که می توان به آن به صورت برنامه نویسی / با برنامه نویسی (programmatically) کنترل فرزند اضافه کرد.

<p><b>IsInPartialRendering</b></p>	<p>مقداری را برمی گرداند که تعیین می کند آیا کنترل UpdatePanel در نتیجه ی روی دادن postback ناهمگام (asynchronous postback) بروز رسانی می شود یا خیر.</p> <p>مشخص می کند آیا panel در نتیجه (به عنوان بخشی از) postback جزئی (partial postback) آپدیت می شود یا خیر.</p>
<p><b>RenderMode</b></p>	<p>مقداری بازیابی یا تنظیم می کند که نشان می دهد محتوای کنترل UpdatePanel در یک تگ &lt;div&gt; یا &lt;span&gt; HTML محصور شده است یا خیر.</p> <p>چنانچه به خاصیت RenderMode مقدار Inline تخصیص داده شود، محتوای کنترل UpdatePanel داخل تگ &lt;span&gt; render می شود. اگر خاصیت RenderMode روی مقدار Block تنظیم شده باشد، داخل المان &lt;div&gt; render می شود.</p>
<p><b>UpdateMode</b></p>	<p>مقداری بازیابی یا تنظیم می کند که نشان می دهد چه زمانی محتوای کنترل UpdatePanel بروز رسانی می شود.</p> <p>این خاصیت rendering mode (حالت ارائه ی ماشینی تصویر) را با مشخص کردن برخی شرایط بازیابی و مقداردهی می کند.</p>
<p><b>Triggers</b></p>	<p>آرایه ای از اشیاء trigger تعریف کرده که هر یک مربوطه به یک رخداد است. این رخدادهای باعث می شوند panel به صورت خودکار بروز رسانی شود.</p> <p>مقداری برمی گرداند که نشانگر آرایه یا مجموعه ای از اشیاء از نوع AsyncPostBackTrigger و PostBackTrigger می باشد.</p> <p>خاصیت Triggers آرایه ای از کل trigger ها که برای کنترل UpdatePanel تعریف شده است برمی گرداند.</p>

## توابع کنترل **UpdatePanel**

متدهای کنترل نام برده به شرح زیر می باشند:

متد	شرح
<b>CreateContentTemplateContainer</b>	این تابع یک شیء Control ایجاد می کند که به عنوان یک طرف برای کنترل های فرزند عمل می کند. کنترل های فرزند محتوایات کنترل UpdatePanel را تعریف می کنند.
<b>CreateControlCollection</b>	آرایه ای از کلیه ی کنترل های موجود در UpdatePanel control را بازیابی می کند.
<b>Initialize</b>	در صورتی که partial-page rendering (ارائه ی ماشینی تصویر به صورت جزئی) فعال شده باشد، آرایه ای از trigger های کنترل UpdatePanel را مقداردهی اولیه می کند.
<b>Update</b>	این متد باعث می شود محتوای کنترل UpdatePanel بروز رسانی شود.

عملکرد کنترل **UpdatePanel** به مقادیر خاصیت های **UpdateMode** و **ChildrenAsTriggers** بستگی دارد.

UpdateMode	ChildrenAsTriggers	نتیجه
Always	False	پارامترهای غیر مجاز
Always	True	چنانچه کل صفحه بازسازی شود یا یک کنترل فرزند به هنگام postback باعث بازسازی صفحه شود، UpdatePanel نیز refresh می شود.
Conditional	False	چنانچه کل صفحه بازسازی / بروز رسانی (refresh) شود یا یک triggering control (کنترلی که به عنوان

		trigger تعریف شده باشد) بیرون از آن باعث بازسازی صفحه شود، کنترل UpdatePanel نیز refresh می شود.
Conditional	True	اگر کل صفحه بروز رسانی شود یا کنترل فرزند باعث شود یک postback (بازگردانی اطلاعات) رخ دهد و یا triggering control (کنترلی که به عنوان trigger تعریف شده باشد) بیرون از صفحه منجر به بازسازی صفحه شود، UpdatePanel نیز بروز رسانی می شود.

## کنترل UpdateProgress

به منظور نمایش یک نمایه تصویری یا یک پیام، که میزان پیشرفت فرایند عملیات بارگذاری و به روز رسانی صفحه را نشان می دهد ، مورد استفاده قرار می گیرد . برای مثال می توان به زمانی اشاره کرد که کاربر در حال ثبت ورود (log in) می باشد یا زمانی که سرور در حال اجرای برخی عملیات مربوط به پایگاه داده بود و کاربر منتظر دریافت پاسخ از سرور می دهنده (server) می باشد، اشاره کرد.

نحوه ی نگارش کنترل UpdateProgress به صورت زیر می باشد:

```
<asp:UpdateProgress ID="UpdateProgress2" runat="server" DynamicLayout="true"
AssociatedUpdatePanelID="UpdatePanel1" >
  <ProgressTemplate>
    Loading...
  </ProgressTemplate>
</asp:UpdateProgress>
```

تکه کد فوق یک پیغام ساده را داخل تگ ProgressTemplate نمایش می دهد. البته بجای پیغام می توان یک تصویر یا هر کنترل مربوطه ی دیگری قرار داد. کنترل UpdateProgress به ازای هر postback ناهمگام (asynchronous) که رخ می دهد یک بار نمایش داده می شود، مگر اینکه با استفاده از خاصیت AssociatedUpdatePanelID به یک updatepanel متصل شده باشد (شما می توانید به کمک property یا خاصیت AssociatedUpdatePanelID کنترل UpdateProgress را به یک کنترل UpdatePanel متصل کنید . هنگامی که عمل Postback یا بازسازی (refresh) توسط کنترل UpdatePanel انجام می شود ، کنترل UpdateProgress مربوط به آن نیز نمایش داده می شود .



اگر کنترل **UpdateProgress** را به هیچ کنترل **UpdatePanel** خاصی متصل نکنید ، با هر بار رخداد **PostBack** یا به روز رسانی ، آن کنترل در صفحه نمایش داده خواهد شد .

### خاصیت های کنترل **UpdateProgress**

خاصیت	شرح
<b>AssociatedUpdatePanelID</b>	شناسه (ID) مرتبط با کنترل update panel را بازگردانی کرده یا مقداردهی می کند.
<b>Attributes</b>	خصیصه های (attribute) CSS کنترل UpdateProgress را بازیابی کرده یا مقداردهی می کند.
<b>DisplayAfter</b>	مدت زمانی که باید سپری شود تا progress template ( قالب پیشروی پروسه ) نمایش داده شود را بر حسب میلی ثانیه خوانده و مقداردهی می کند. پیش فرض 500 می باشد.
<b>DynamicLayout</b>	مقداری خوانده و تنظیم می کند که نشان می دهد آیا progress template به صورت پویا (dynamic) render می شود یا خیر. این خاصیت تعیین می کند آیا فضای لازم برای محتوای progress template تنها زمانی به صفحه تخصیص داده شود که لازم است یا خیر.
<b>ProgressTemplate</b>	قالبی (template) که محتوای کنترل UpdateProgress را تعریف می کند، مقداردهی کرده یا برمی گرداند. نشانگر قالبی است که حین عمل postback ناهمگام نمایش داده می شود. این فرایند زمان بیشتری نسبت به DisplayAfter طول می کشد.

توابع کنترل **UpdateProgress**:

شرح	تابع
<p>فهرستی از مولفه ها (component)، عملکردها (behavior) و کنترل های سمت سرویس گیرنده (client control) بازیابی می کند که برای راه اندازی قابلیت های کنترل UpdateProgress سمت سرویس گیرنده لازم و ضروری می باشد.</p> <p>این متد را برای تعریف کامپوننت ها، عملکردها یا کنترل های client از کتابخانه ی ایجکس مایکروسافت (Microsoft AJAX Library) که server control ها به آن نیاز دارند، پیاده سازی کنید.</p>	GetScriptDescriptors
<p>آرایه ای از اشیاء ScriptReference باز می گرداند که تعریف کننده ی منابع اسکریپت (script resources) بوده و کنترل مورد نظر به آن نیاز دارد.</p>	GetScriptReferences

## کنترل Timer

از این کنترل به منظور راه اندازی عمل **postback** به صورت خودکار استفاده می شود. کنترل **Timer** به صورت متناوب، در فواصل زمانی تعیین شده، صفحه را **Postback** می کند. همچنین اگر این کنترل را به همراه یک کنترل **UpdatePannel** به کار ببرید، می توانید تنها آن بخشی از صفحه که کنترل **UpdatePannel** شامل آن می شود را به صورت متناوب، بروز رسانی کنید. این کار از دو طریق زیر امکان پذیر می باشد:

### 1. متصل کردن و مقداردهی خاصیت **Triggers** کنترل **UpdatePanel**:

```
<Triggers>
  <asp:AsyncPostBackTrigger ControlID="btnpanel12" EventName="Click" />
</Triggers>
```

2. با قرار دادن یک کنترل **timer** به صورت مستقیم داخل **UpdatePanel** که به عنوان **trigger** کنترل فرزند عمل می کند. توجه داشته باشید که تنها یک کنترل **timer** می تواند به عنوان **trigger** برای چندین **UpdatePanel** عمل کند.

```
<asp:UpdatePanel ID="UpdatePanel12" runat="server" UpdateMode="Always">
```

```
<ContentTemplate>
  <asp:Timer ID="Timer2" runat="server" Interval="1000">
    </asp:Timer>
  <asp:Label ID="Label1" runat="server" Height="101px" style="width:304px" >
</asp:Label>
</ContentTemplate>
</asp:UpdatePanel>
```