

مقدمه

در روش سنتی زمانی که کلیک می کردیم صفحه PostBack می شد و User را به صفحه دیگری برده و اعمال لازم را انجام میداد. علت این کار این بود که پیاده سازی صفحه به صورتی که اشیا مقدار خود را حفظ کنند مشکل بود.

مایکروسافت این مشکل را با استفاده از تکنولوژی ViewState برطرف کرد. در این تکنولوژی اشیا پس از PostBack شدن صفحه مقدار خود را حفظ می کنند. پس از آن Validation ها را به صفحات وب اضافه کرد. در Validation ها در روش سنتی همیشه به صورت ServerSide عمل می شد. مایکروسافت انواع Validation ها را در اختیار برنامه نویسان قرار داد که بصورت ClientSide پیغام می داد و جلوی ادامه کار را می گرفت ولی ServerSide بودن آن باعث می شد که صفحه لااقل یک بار PostBack شود (برای جلوگیری از حمله های از نوع BackDoor).

در ابتدا این ایده شکل گرفت که یک سری از صفحات را مجهز به قابلیت XML و SOAP کنند. در این معماری پس از تغییر قسمت مورد نظر یک تابع JavaScript اجرا می شد و درخواست خود را به صفحه ASP یا PHP می فرستاد و این صفحه یک صفحه جدید مبتنی بر XML می ساخت و شما با استفاده از پروتکل SOAP مقادیر مورد نظر خود را درخواست می کردید و سپس با استفاده از JavaScript مقادیر مورد نظر در صفحه خود را Update می کردید.

یک برنامه نویس آلمانی با استفاده از این ایده یک Library نوشت که این کار را در Server بصورت Runtime انجام می داد ولی کماکان JavaScript های سمت Client را بایستی خود برنامه نویس می نوشت. این Library به Ajax مشهور شده است. مایکروسافت با استفاده از این ایده یک Library ساخت به نام Atlas که هم اسکریپت های سمت Client را بصورت خودکار می ساخت و هم اسکریپت های سمت Server را.

شروع با Atlas

پس از نصب Atlas (آن را باید از سایت مایکروسافت دریافت کنید) یک Template جدید به بخش Template های موجود در پنجره New Web Site اضافه می شود با عنوان: Atlas Web Site

پس از انتخاب New Web Site نوع پروژه را از نوع Atlas Web Site انتخاب می کنیم و کلید OK را انتخاب می نماییم. صفحه پیش فرض را به Learn1 تغییر نام می دهیم. یک TextBox روی صفحه قرار می دهیم. نام آن را txtData نامگذاری می کنیم. یک Button روی صفحه قرار داده و نام آن را btnSubmit می گذاریم. در رویداد Click آن کد زیر را قرار می دهیم:

```
txtData.Text="Hellow, World!";
```

تا اینجا به روش سنتی عمل کرده ایم یعنی پس از کلیک روی Submit صفحه Postback می شود.

حالا یک صفحه جدید با همان مشخصات صفحه قبلی به پروژه اضافه نموده و نام آن را به Learn2 تغییر می دهیم. در Learn2 می خواهیم از Atlas استفاده کنیم. ابتدا باید یک شی از نوع ScriptManager روی صفحه قرار دهیم. کار این شی ساختن Script های لازم در لایه Client می باشد. این کار را بصورت زیر انجام می دهیم:

```
<atlas:ScriptManager ID="ScriptManager1" runat="server"
EnablePartialRendering="true" />
```

توجه داشته باشید که همیشه ScriptManager باید بعد از تگ Form قرار گیرد. معنی PartialRendering این است که وقتی در یک ناحیه تغییری صورت گرفت فقط همان ناحیه تغییر کند.

چیزهایی که به هم مرتبط هستند را باید در یک ناحیه قرار داد. در مثال ما TextBox و Button به هم وابسته می باشند بنابراین آنها را در یک ناحیه قرار می دهیم. این کار را با استفاده از کد زیر انجام می دهیم:

```
<atlas:UpdatePanel ID="up1" runat="server">
  <ContentTemplate>
    <asp:TextBox ID="txtData" runat="server"
      Width="215px"></asp:TextBox>&nbsp;
    <asp:Button ID="btnSubmit" runat="server"
      OnClick="btnSubmit_Click" Text="Submit" />
  </ContentTemplate>
</atlas:UpdatePanel>
```

در این حالت با انتخاب کلید Submit دیگر Postback اتفاق نمی افتد.

حالا یک کپی از صفحه Learn2 را به پروژه اضافه نموده و نام آن را به Learn3 تغییر می دهیم. برای اینکه به User اعلام کنیم که صفحه در حال پردازش است (تا از کلیک مجدد کاربر جلوگیری کنیم) باید کارهای زیر را انجام دهیم:

برای نمایش Progress به کاربر باید از UpdateProgress استفاده کنیم . به مثال زیر توجه نمایید:

```
<atlas:UpdateProgress ID="UpdateProgress" runat="server">
  <ProgressTemplate>
    <div class="Progress">
      Please Wait...
    </div>
  </ProgressTemplate>
</atlas:UpdateProgress>
```

برای جلوگیری از تغییر مکان Layout در هنگام نمایش پیغام، تغییرات زیر را در آن انجام می دهیم:

```
<div class="Header">
  <atlas:UpdateProgress ID="UpdateProgress" runat="server">
    <ProgressTemplate>
      <div class="Progress">
        Please Wait...
      </div>
    </ProgressTemplate>
  </atlas:UpdateProgress>
</div>
```

یک فولدر به نام CSS ساخته و یک StyleSheet به آن اضافه می نماییم. کد زیر را درون آن قرار می دهیم:

```
body
{
  font-family: Verdana;
  font-size: 8pt;
}

.Header
{
  height: 80px;
}

.Progress
{
  color: Red;
}
```

برای Trace کردن رفتارهای Atlas باید در اجرای آن در حالت Local وقفه ایجاد نماییم تا رفتار واقعی آن را مشاهده کنیم. برای این کار در رویداد Click کلید Submit کد زیر را قرار می دهیم:

```
System.Threading.Thread.Sleep(2000);
```

این کار باعث یک وقفه 2 ثانیه ای می شود. توجه داشته باشید که این کد را در حالت واقعی وقتی که صفحات را روی وب قرار می دهیم باید حذف کنیم و از آن فقط برای تست استفاده نماییم.

معمولا برای حالت Progress از تصاویر خاصی استفاده می کنند که فرمت آنها Gif Animation می باشد.

✽ با تشکر از جناب مهندس تصدیقی