ASP.NET AJAX

مقدمة

في الدرس السابق قد تم شرح طريقة أستخدام أداة UpdatePanel و تكلمنا عن ماهي وضيفتها و ما أهم الخصائص المتعلقة بها ، وكانت وظيفة هذه الأداة هي تجنب التحديث الكامل للصفحة.

وقبل أستخدام هذه الأداة وعند عمل أي تحديث للصفحة تستطيع ملاحظة وجود حركة بسيطة يقدمها المتصفح وتختلف هذه الحركة بأختلاف المتصفح الذي تسخدمه، ولكن بعد أستخدام أداة UpdatePanel أصبحت العمليات تنفذ بدون ظهور هذه الحركة من المتصفح، عندها سوف يصبح المستخدم لا يعلم هل أنه تم أرسال طلبة لمعالجته أو أن هناك خطئ في هذا الطلب .

أذا لابد من أظهار على سبيل المثال رسالة تظهر للمستخدم أنه يجب عليه الانتظار وذلك ليتم تنفيذ طلبة، فهنا يأتي دور الأداة المساعدة لأداة UpdatePanel وهي UpdateProgress. أذا لنتعرف على هذه الأداة وماهي أهم الخصائص وكيفية ربطها مع أداة UpdatePanel لأظهار رسالة تنبه المستخدم تفيد بإن طلبة يتم معالجتة أو نستطيع تسميتها User Feedback.

UpdateProgress

هي أداة نربطها مع أداة الـ UpdatePanel للمساعدة على أظهار تنبيه للمستخدم بأن الطلب الذي أرسلة للخادم يتم معالجتة، ولنفرض أنة يوجد لا هناك أي تنبيه ففي بعض الحالات يقوم المستخدمون بالضغط على كل الأزرار الموجودة ولنفترض وجود بعض الأزرار تحتاج إلى وقت لإرجاع النتائج وبعضها يكون ذو وقت أقل، فإن المستخدم سوف يواصل الضغط وصولا إلى هذا الزر الذي سوف يرد نتيجتة في وقت أقل.

وكما ذكرت بإن هذه الأداة لها خصائص لأ بد من تطبيقها ليتم أستخدامها، أذا لنتعرف على هذه الخصائص:

AssociatedUpdatePanelID

تعتبر أهم خاصية لأستخدام أداة UpdateProgress، وهذه الخاصية تستخدم لتحديد أسم الـ UpdatePanel التي سوف يظهر التنبيه عندما يتم تحديث هذه الـ UpdatePanel.

ProgressTemplate

هذه الخاصية التي تحتوي على رسالة التنبيه الذي نريده أن يظهر للمستخدم وعلى سبيل المثال أبسط العبارات ...Please Wait. ويكون تعريفها بهذا الشكل:

```
<asp:UpdateProgress ID="UpdateProgress1" AssociatedUpdatePanelID="" runat="server">
     <ProgressTemplate>
         <%--The Content of UpdateProgress--%>
     </ProgressTemplate>
</asp:UpdateProgress>
```

DisplayAfter

هذه الخاصية تُستخدم لتحديد المدة الزمنية التي تظهر فيها أداة الـ UpdateProgress المحتوى الموجود في ProgressTemplate وكما ذكرت في الأعلى على سبيل المثال ....Please Wait. الوقت الموجود عند استخدام الأداة هو 500 milliseconds أي ما يعادل نصف الثانية ويفضل أستخدام هذا الوقت لأن هناك بعض الحالات يكون فيها زمن الأرسال والمعالجة والأرجاع من الخادم تكون بسيطة ولنقل على سبيل المثال ثانيه واحدة فعندها نستخدم نحن زمن نصف ثانية لتظهر لنا كلمة ...Please wait لأعلام المستخدم بأن هناك معاجلة لطلبك كما ذكرت سابقا.

DynamicLayout

تُحدد هذه الخاصية الوضع في أظهار المحتوى الموجود في الـ ProgressTemplate وهي تحتوي على خيارين وهما true أو false ، وتكون طريقة هذه الوضعية بأستخدام Display الموجودة كما في الـ CSS و Visibility.

أذا لنتقدم ونعمل مثال لتطبيق طريقة عمل هذه الأداة كالتالي:

كالعادة أفتح برنامج Visual Studio ومن قائمة File أختر New web Site وبعدها نختار ASP.NET Empty Web Site ، وقم بتستمية بما تريد.

بعدها لنقم بوضع أداة UpdatePanel و نضع بداخلها كل من Label و Button ليكون الشكل النهائي كالتالي:

```
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
     <ContentTemplate>
         <asp:Label ID="lbl1" runat="server"></asp:Label><br />
             <asp:Button ID="Button1" runat="server" Text="Refresh the label"
                   onclick="Button1_Click" />
     </ContentTemplate>
</asp:UpdatePanel>
```

ونحتاج الى تفعيل حدث Click للزر الموجود بداخل الـ UpdatePanel. و تحت أداة الـ UpdatePanel مباشرة نضع أداة الـ UpdateProgress أي تحت </UpdatePanel> وأضافة ProgressTemplate، ليكون الشكل النهائي كالتالي:

```
<asp:UpdateProgress ID="UpdateProgress1" runat="server">
    <ProgressTemplate>

    </ProgressTemplate>
</asp:UpdateProgress>
```

كما ذكرنا سابقا بأننا نحتاج إلى تحديد أسم الـ UpdatePanel وذلك بكتابة أسمها في خاصية AssociatedUpdatePanelID وفي مثالنا سوف يكون أسم الـ UpdatePanel هو UpdatePanel1، وأيضا سوف نكتب داخل الـ ProgressTemplate جملة Please Wait… ليكون الشكل النهائي كالتالي:

```
<asp:UpdateProgress ID="UpdateProgress1" AssociatedUpdatePanelID="UpdatePanel1"
runat="server">
   <ProgressTemplate>
        Please Wait...
   </ProgressTemplate>
</asp:UpdateProgress>
```

وفي خطوة أخيرة يتم أضافة الكود لملف الـ Code Behind كالتالي:

```
protected void Page_Load(object sender, EventArgs e)
{
    lbl1.Text = DateTime.Now.ToLongTimeString();
}


protected void Button1_Click(object sender, EventArgs e)
{
    System.Threading.Thread.Sleep(2000);
    lbl1.Text = DateTime.Now.ToLongTimeString();
}
```

ملاحظة: تم أستخدام السطر الأول من الكود الموجود في حدث الـ Click للرز فقط من أجل أضافة تأخير وذالك لرؤية النتيجة التي نريد أن نوصل لها.

أفتح الصفحة وشاهد العملية عندما يتم الضغط على الزر لتحديث الوقت.

أيضاً لنأخذ جانب بسيط لشرح خاصية الـ DynamicLayout وفهم الفرق بين أختيار True وهي الحالة الأساسية وبين تغير هذه القيمة إلى False، طبعاً نحتاج لتعديل بسيط على مثالنا وهو وضع أداة الـ UpdateProgress مباشر فوق أداة الـ UpdatePanel.

وقبل أن نغير القيمة من True إلى False، أفتح الصفحة ولأحظ النتيجة، حيث سوف نلاحظ بأنه لأ يوجد أثر لأداة الـ UpdateProgress الأ بعد الضغط على الرز. لأن في هذه الحالة سوف يستخدم خاصية Display وجعل القيمة none لأخفاء محتوى الـ UpdateProgress.

في الخطوة التالية قم بتغير قيمة الـ DynamicLayout من True الى False وافتح الصفحة، سوف تلاحظ وجود مسافة قبل أداة الـ UpdatePanel وذلك لأنه سوف يتم أعتبار ما يوجد في UpdateProgress على أنة ولكن تم أخفاء المحتوى الموجود بداخلة بأستخدام خاصية Visibility وجعل القيمة hidden وجعل خاصية Display ذات قيمة block.

وقبل الختام، نستطيع وضع صور ذات امتداد gif والتي تدل على فترة تحميل داخل الـ ProgressTemplate وذلك بجلب الصورة الى محتويات الموقع ومن ثم وضعها من ضمن المحتوى. والمثال التالي هو لتعديل بسيط على مثالنا ووضع صورة تدل على فترة التحميل:

```
<asp:UpdateProgress ID="UpdateProgress1" AssociatedUpdatePanelID="UpdatePanel1"
                    runat="server" DynamicLayout="true" DisplayAfter="500">
     <ProgressTemplate>
           Please Wait...<br />
           <img src="ajaxloader.gif" alt="" />
     </ProgressTemplate>
</asp:UpdateProgress>
```

أفتح الصفحة ولاحظ الفرق.

وفي الختام، أضافت أداة الـ UpdateProgress فاعيلة أضافية على أداة الـ UpdatePanel و ذلك بتفاعلها مع المستخدم وأعلامة بما يحصل وقت معالجة الطلب الذي هو قدمة للخادم.