

مقدمه

از آنجا که در این مقاله از تکنولوژی Atlas استفاده شده است در صورتی که با آن آشنا نیستید ابتدا مقاله "آشنایی با Atlas" را مطالعه نمایید. این مقاله را می توانید از وبلاگ CSharpBlog دانلود نمایید.

ابتدا در VS 2005 گزینه New Web Site را انتخاب نموده و از پنجره نمایش داده شده Template با نام Atlas Web Site را انتخاب نمایید.

برای ایجاد کردن Database از منوی کلیک راست گزینه "Add New Item" را انتخاب نمایید. از گزینه های موجود در پنجره مورد نظر، SQL Database را انتخاب نمایید. پیغامی مبنی بر قرار دادن آن در فولدر App_data نمایش داده می شود، آن را تایید کنید. یک فایل mdf در فولدر App_data ساخته می شود که شما می توانید نام آن را تغییر دهید. در این مثال نام آن را به TaskManager.mdf تغییر دهید. یک جدول به نام Task بسازید و در آن سه فیلد به نامهای TaskID، Name، Completed تعریف کنید.

یک Typed Dataset به پروژه اضافه نمایید. نحوه انجام این کار بدین صورت است که : از پنجره "Add New Item" گزینه Dataset را انتخاب نموده و نام آن را به MyDataset نامگذاری می کنیم. پیغامی مبنی بر اضافه شدن Dataset به فولدر App_Code نمایش داده می شود، آن را تایید کنید. پس از آن یک TableAdaptor روی صفحه قرار داده و پنجره TableAdaptor Configuration Wizard نمایش داده می شود. در مرحله اول بانک اطلاعاتی که می خواهیم به آن متصل شویم را باید معرفی کنید. در این مثال گزینه TaskManager.mdf را از لیست انتخاب می کنیم و کلید Next را کلیک می کنیم. در مرحله بعد از شما در مورد ذخیره ConnectionString در فایل تنظیمات برنامه (WebConfig) پرسیده می شود. آن را تایید کنید. در مرحله بعد نوع دسترسی TableAdapter به بانک اطلاعاتی را مشخص می کنیم. در این مثال گزینه Use SQL Statement را انتخاب کنید. در مرحله بعد باید دستور SQL مورد نظرتان را بنویسید. با استفاده از Query Builder جدول و یا جدولهای مورد نیاز را به Query Builder اضافه و فیلدهای مورد نظر را مشخص کنید (در این مثال همه فیلدها را انتخاب نمایید).

برای اینکه در این مثال بتوانید بصورت پارامتریک عمل کنید باید عبارت SQL را تکمیل نمایید (بخش WHERE را به آن اضافه نمایید):

```
SELECT TaskID, Description, Completed  
FROM Tasks  
WHERE Completed=@IsComplete
```

در مرحله بعد گزینه Fill a DataTable را از حالت انتخاب خارج نمایید و نام متد قسمت Return a DataTable را به GetDataByStatus تغییر دهید. کلید Finish را کلیک کنید. یک TableAdapter ساخته می شود که حاوی فیلدهای جدول مورد نظر و همچنین متدی به نام GetDataByStatus می باشد.

یک صفحه به نام TaskManager.aspx به پروژه اضافه کنید و روی آن یک DropDownList و یک Label قرار دهید. ID کنترل DropDownList را به CboStatus تغییر دهید. در حالت Design روی SmartTag آن کلیک کرده و گزینه Enable Auto Postback آن را انتخاب نمایید. علت این کار این است که می خواهیم به محض عوض کردن مقدار آن، در Grid فهرست متناسب با آن انتخاب شود. گزینه Edit Items را نیز کلیک کنید و دو آیتم با مشخصات ذکر شده در جدول زیر به آن اضافه نمایید:

Selected	Value	Text
True	False	Not Complete
False	True	Complete

یک شی از نوع ObjectDataSource روی صفحه قرار دهید و نام آن را به TaskDataSource تغییر دهید. با استفاده از این شی می توانیم به جدول خاصی در یک Dataset متصل شویم. در صورتی که بخواهید به صورت مستقیم به Database متصل شوید می توانید از SqlDataSource استفاده نمایید. در پنجره Configure Data Source نام TableAdaptor مربوطه را انتخاب نمایید. در مرحله بعد کلید Next را انتخاب نمایید تا همان مقادیر پیش فرض برای Select, Update, Delete, Insert نظر گرفته شود. از آنجایی که دستور SQL تعیین شده بصورت پارامتریک می باشد پنجره ای ظاهر می شود که در آن پارامتر موجود در دستور Select را با مقدار مناسب ارتباط می دهیم. از آنجایی که در این مثال می خواهیم اطلاعات نمایش داده شده با توجه به مقدار موجود در DropDownList تعیین شود بنابراین از لیست Parameter Source گزینه Control و از لیست ControlID گزینه cboStatus را انتخاب و کلید Finish را کلیک نمایید.

مرحله بعد اضافه کردن یک کنترل GridView به صفحه می باشد. پس از اضافه کردن آن از SmartTag نمایش داده شده Auto Format را انتخاب و از لیست Scheme ها گزینه Slate را انتخاب نمایید. در قسمت Choose DataSource گزینه TaskDataSource را انتخاب نمایید. روی ستون TaskID کلیک کنید تا به حالت انتخاب درآید سپس در قسمت SmartTag گزینه Remove Column را انتخاب کنید. گزینه های Enable Paging و Enable Sorting و Enable Editing را فعال نمایید.

یک کنترل DetailsView روی صفحه قرار دهید. از منوی SmartTag گزینه های Enable Inserting و Enable Editing را انتخاب کنید. در قسمت Choose DataSource گزینه TaskDataSource را انتخاب کنید. در قسمت Auto Format گزینه Slate را کلیک کنید. حالا آن را اجرا کنید. خواهید دید که با نوشتن یک Task جدید در DetailView و انتخاب Insert اطلاعات موجود در GridView به روز می شود. اکنون وقت آن است تا این صفحه را مجهز به تکنولوژی Atlas نمایید!

اضافه کردن Atlas

برای اضافه نمودن Atlas به صفحه ابتدا یک ScriptManager روی صفحه قرار داده و ID آن را به ScripManager تغییر دهید. ScripManager باید در ابتدای Form قرار گرفته باشد.

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

پس از آن یک UpdateProgress به آن اضافه کنید:

```
<div class="Header" style="height: 17px">
  <atlas:UpdateProgress ID="UpdateProgress" runat="server">
    <ProgressTemplate>
      <div class="Progress">
        Please Wait...
      </div>
    </ProgressTemplate>
  </atlas:UpdateProgress>
</div>
```

یک UpdatePanel اضافه کرده و GridView را در آن قرار دهید:

```
<atlas:UpdatePanel ID="upGridView" runat="server">
  <ContentTemplate>
    <asp:GridView ...>
      .
      .
      .
    </asp:GridView>
  </ContentTemplate>
</atlas:UpdatePanel>
```

تا اینجا اگر پروژه را اجرا نمایید خواهید دید که تغییراتی مانند Edit و Sort در GridView باعث PostBack شدن صفحه نخواهد شد. در این حالت اگر مقدار DropDownList تغییر کند برای تغییرات متناسب در GridView، صفحه PostBack می شود. برای جلوگیری از این کار از آنجا که GridView و DropDownList در یک UpdatePanel نیستند باید از Trigger استفاده کنید!

با استفاده از Trigger ها می توانید محتویات UpdatePanel را نسبت به بخش خاصی از صفحه حساس کنید بدین نحو که به محض تغییر در آن بخش خاص، محتویات UpdatePanel بدون PostBack تغییر کند.

دو نوع Trigger داریم به نامهای:

ControlEventTrigger: در صورت رخ دادن Event خاص

ControlValueTrigger: در صورت تغییر در مقدار Property خاص در یک کنترل

تغییرات زیر را در کد انجام دهید:

```
<atlas:UpdatePanel ID="upGridView" runat="server">
    <ContentTemplate>
        <asp:GridView ...>
            .
            .
            .
        </asp:GridView>
    </ContentTemplate>
    <Triggers>
        <atlas:ControlValueTrigger ControlID="cboStatus"
            PropertyName="SelectedValue" />
    </Triggers>
</atlas:UpdatePanel>
```

اکنون آن را اجرا نمایید. خواهید دید که در صورت تغییر مقدار موجود در DropDownList صفحه PostBack نخواهد شد.

برای اینکه عمل Insert نیز با استفاده از Atlas انجام شود اعمال زیر را انجام دهید.

یک UpdatePanel اضافه می کنیم. Mode آن را به Conditional تغییر می دهیم و DetailsView را در آن قرار می دهیم:

```
<atlas:UpdatePanel ID="upDetailsView" runat="server"
Mode="Conditional">
    <ContentTemplate>
        <asp:DetailsView ...>
            .
            .
            .
        </asp:DetailsView>
    </ContentTemplate>
</atlas:UpdatePanel>
```

در صورتی که Mode آن Always باشد یعنی اینکه هر تغییری که در صفحه انجام شد UpdatePanel را به روز کند. در صورتی که Mode آن Conditional باشد هر گاه خود UpdatePanel تغییر داده شد دیگران تغییر داده شوند.

* با تشکر از جناب مهندس تصدیقی