

1. 概述

- 使用 `DOM` 进行动态显示及交互
- 使用 `XML` 及 `XSLT` 进行数据交换及操作
- 使用 `XMLHttpRequest` 进行异步数据检索
- 使用 `JavaScript` 将所有的东西绑定在一起
- 最大的优点
 - 能在不更新整个页面的下与服务器交换数据并更新部分网页内容
 - 避免了在网络上发送没有改变过的数据
- 异步刷新
 - 整个页面不刷新，实现局部网页刷新
 - 实现不刷新页面完成前后台数据更新

2. ScriptManager 控件

2.1. ScriptManager

- `AsyncPostBackTimeout`：异步回传时超时限制，默认90s
- 实现选择不同项，局部刷新文本

```
<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
<asp:DropDownList ID="DropDownList1" runat="server"
OnSelectedIndexChanged="DropDownList1_SelectedIndexChanged">
    <asp:ListItem>Hello World</asp:ListItem>
    <asp:ListItem>Hello Ajax</asp:ListItem>
</asp:DropDownList>
```

```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e){
    Label1.Text = DropDownList1.SelectedValue;
}
```

2. UpdatePanel

- 定义一个局部实现此局部内部的更新

2.1. 常用属性

2.1.1. Triggers 属性实现不同的刷新方式

- 通过 `Triggers` 指定刷新方式

```
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
    <ContentTemplate>

        <asp:Label ID="Label1" runat="server" Text="当前时间: "></asp:Label>
        <asp:Label ID="Label2" runat="server" Text="Label"></asp:Label>
        <br /> <br />
        <asp:Button ID="Button1" runat="server" Text="Button1" onclick="Button1_Click"
    />

        <br />

    </ContentTemplate>

    <Triggers>
<!--同步--> <asp:PostBackTrigger ControlID="Button1" />
<!--异步--> <asp:AsyncPostBackTrigger ControlID="Button2" />
    </Triggers>

</asp:UpdatePanel>
<br />
<asp:Button ID="Button2" runat="server" Text="Button2" onclick="Button2_Click" />
```

```
protected void Button1_Click(object sender, EventArgs e){
    this.Label2.Text = DateTime.Now.ToString();
}

protected void Button2_Click(object sender, EventArgs e){
    this.Label2.Text = DateTime.Now.ToString();
}
```

2.1.2. UpdateMode 属性

- 当页面上有多个 `UpdatePanel` 时, `UpdateMode` 默认是 `Always`, 其中一个触发局部更新, 那么所有都会更新
- `Always`: 无论是否定义 `Triggers` 都更新
- `Conditional`: 只有定义了 `Triggers` 时才更新

```
<!-- 设置当前的updatepanel的更新方式 -->
<asp:UpdatePanel ID="UpdatePanel1" runat="server" UpdateMode="Conditional">
    <ContentTemplate>
        <fieldset style="width:180px;">
            <legend >UpdatePanel1</legend>
            <asp:ListBox ID="ListBox1" runat="server" Width="170px"></asp:ListBox>
```

[illegible]

```
protected void InsertButton_Click(object sender, EventArgs e){
    ListBox1.Items.Add(TextBox1.Text);
    TextBox1.Text = String.Empty;
}

protected void Cancelbutton_Click(object sender, EventArgs e){
    TextBox1.Text = String.Empty;
}
```

3. Timer 控件

- 通过 `Timer` 控件可以实现局部页面的定时更新
- `Interval` 属性：指定更新的间隔时间
 - 单位是 `ms`
- `Tick` 属性：指定间隔到期后触发，即重新计时
- `Enabled` 属性：表示是否允许 `Tick`，`False` 表示停止计时

- `Timer` 定义在 `UpdatePanel` 内部时, 计时器只会有一次回传完成后才开始从0计时
- 定义在外部时, 当回传被引发后立即开始计时

- 实现使用 `Timer` 定时更新完成图片轮播

```
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
<ContentTemplate>
<asp:Timer ID="Timer1" runat="server" Interval="1000" ontick="Timer1_Tick">
</asp:Timer>
<asp:Image ID="Image1" runat="server" ImageUrl="TimerExample1.gif"/>
</ContentTemplate>
</asp:UpdatePanel>
```

```
protected void Page_Load(object sender, EventArgs e){
    if (Page.IsPostBack == false){
        //设置网页上的变量
        ViewState["number"] = 1;
    }
}
protected void Timer1_Tick(object sender, EventArgs e){
    ViewState["number"] = (int)ViewState["number"] % 4 + 1;
    Image1.ImageUrl = string.Format("~/images/{0}.png", ViewState["number"]); // {0} 类
    似%d
}
```

4. 习题

- AJAX 技术原理

当用户与浏览器中的页面进行交互时, 会触发页面元素对象的响应事件, 客户端捕获相应的事件后, 如果需要提交给服务器进行处理, 客户端就将要处理的数据(包括 状态描述)转换为 XML 或 JSON 格式的字符串, 再利用异步传送方式将这些数据提交给 服务器;服务器处理后, 同样利用 XML 或 JSON 格式和异步传送方式将处理结果返回给客户端;客户端再从返回结果中提取需要的部分, 并将提取的部分利用 JavaScript 对网页进行“悄悄地”局部更新, 而不是刷新整个页面。

- AJAX 的优势(作用)

(1) 页面局部刷新。在读取数据的过程中, 用户所面对的不是白屏, 而是原来的页面状态, 或者正在更新的信息提示状态, 只有当接收到全部数据后才更新相应部分 的内容, 而这种更新也是瞬间的, 用户几乎感觉不到。

(2) 把一些由服务器承担的工作转移到客户端处理, 这样可以充分利用客户端闲置 的处理能力, 减轻服务器和网络传输的负担。

(3) 使用基于标准化的并被广泛支持的技术, 不需要安装插件即可运行。

(4) 使 Web 中的界面与应用分离, 也可以说是数据与呈现分离。

