ASP .NET

ASP.NET

- ASP.NET works on top of the HTTP protocol
- Takes advantage of HTTP commands and policies to set up two-way, browser-to-server communication and cooperation
- Web Forms model: the event-driven model of interaction finally comes to the Web

Event-Driven Programming over HTTP

- Implementing an event model over the Web requires any data related to the client-side user's activity to be forwarded to the server for corresponding and *stateful* processing
- The server needs to process the output of client actions and trigger reactions while being aware of the overall state of the application

States

- The state of the application contains two types of information: the state of the client and the state of the session
- The state of the client is easily accessible through the syntax and the implementation of the *<form>* HTML element. But what about the overall state of the session?
- A reentrant form is an HTML *<form>* element that posts to the same page that contains it

Web Forms

- A WebForm consists of two components: the *user interface* (*UI*) and the *programming (application) logic*
- The user interface is the visual component of a WebForm; it consists of HTML and controls specific to the Web application
- The programming logic of a Web application in ASP.NET is contained in a separate file that contains the code to handle the user's interaction with the form

HTTP

- Defines how web browsers and web servers communicate with each other
- Typically transmitted over TCP
- HTML

```
<html>
 <body>
  Hello, world
 </body>
</html>
```

HTTP Request

```
GET /simple.html HTTP/1.1
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
If-Modified-Since: Wed, 24 Oct 2001 14:12:36 GMT
If-None-Match: "50b0d3ee955cc11:a78"
User-Agent: Mozilla/4.0.(compatible; MSIE.6.0; Windows
  NT 5.1)
Host: www.wintellect.com
Connection: Keep-Alive
[blank line]
```


HTTP Response

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Wed, 24 Oct 2001 14:12:37 GMT
Content-Type: text/html
Accept-Ranges: bytes
Last-Modified: Wed, 24 Oct 2001 14:00:53 GMT
ETag: "d02acf81975cc11:a78"
Content-Length: 46
[blank line]
<html>
<body>
Hello, world
</body>
</html>
```


HTML Forms

- Forms add user input capabilities to static HTML files
- Appears between <form> and </form> tags

```
<html>
  <body>
    <form>
      <input type="text" name="op1" />
      +
      <input type="text" name="op2" />
      <input type="submit" value="  =  " />
    </form>
  </body>
</html>
```


Submit Buttons

- <input type="submit">
- When clicked, it submits the form to a Web server
- The browser submits the form along with any input in the form's controls
  - GET
  - POST

GET

```
GET /calc.html?op1=2&op2=2 HTTP/1.1
   .
   .
   .
Connection: Keep-Alive
[blank line]
```


POST

- Rather than transmit user input in the URL, with a POST command the browser passes it in the body of the HTTP request

```
POST /calc.html HTTP/1.1
   .
   .
   .
Content-Type: application/x-www-form-
  urlencoded
Content-Length: 11
[blank line]
op1=2&op2=2
```

Postback

- Regardless of whether a GET or a POST command is used, when input from an HTML form is submitted back to the server, we say that a “postback” has occurred

Server-Side Processing

- How do we go from the previous HTML page to this one

```
<html>
  <body>
    <form>
      <input type="text" name="op1" value="2" />
      +
      <input type="text" name="op2" value="2" />
      <input type="submit" value="  =  " />
      4
    </form>
  </body>
</html>
```


Solution 1

- Use Common Gateway Interface (CGI)
- Applications that use it are typically written in Perl
- CGI applications read the input accompanying postbacks through server environment variables and standard input (stdin), and they write HTTP responses to standard output (stdout)

Solution 2

- ISAPI extension DLL
- Internet Server Application Programming Interface
- ISAPI extensions are Windows DLLs that are hosted by Internet Information Services
- IIS forwards HTTP requests to an ISAPI DLL by calling a special function exported from the DLL
- The DLL generates HTTP responses

Solution 3

- The Active Server Pages Solution
- Active Server Pages lower the barrier to entry for Web developers by allowing HTML and server-side script to be freely mixed in ASP files
- Scripts are written in JScript or VBScript

ASP

- Active Server Pages lower the barrier to entry for Web developers by allowing HTML and server-side script to be freely mixed in ASP files

ASP

```
<%@ Language="VBScript" %>
<html>
  <body>
    <form>
      <input type="text" name="op1" value="<%= Request ("op1") %>"/>
      +
      <input type="text" name="op2" value="<%= Request ("op2") %>" />
      <input type="submit" value="  =  " />
      <%
        If Request ("op1") <> "" And Request ("op2") <> "" Then
            a = CInt (Request ("op1"))
            b = CInt (Request ("op2"))
            Response.Write (CStr (a + b))
        End If
      %>
    </form>
  </body>
</html>
```


ASP

- What's wrong with ASP?
- It's slow
  - Why?
  - Interpreted rather than compiled
- Lacks encapsulation

ASP.NET Web Forms

- Web forms bring object-oriented programming to the Web
- Combine ASP's ease of use with the speed of compiled code

ASP.NET Web Forms

```
<html>
  <body>
    <form runat="server">
      <asp:TextBox ID="op1" RunAt="server" />
      +
      <asp:TextBox ID="op2" RunAt="server" />
      <asp:Button Text="  =  " OnClick="OnAdd" RunAt="server" />
      <asp:Label ID="Sum" RunAt="server" />
    </form>
  </body>
</html>
<script language="C#" runat="server">
  void OnAdd (Object sender, EventArgs e)
  {
      int a = Convert.ToInt32 (op1.Text);
      int b = Convert.ToInt32 (op2.Text);
      Sum.Text = (a + b).ToString ();
  }
</script>
```


Generated HTML

```
<html>
  <body>
    <form name="_ctl0" method="post" action="calc.aspx"
  id="_ctl0">
      <input type="hidden" name="__VIEWSTATE"
  value="dDwxOTE0NDY4ODE2Ozs+" />

      <input name="op1" type="text" id="op1" />
      +
      <input name="op2" type="text" id="op2" />
      <input type="submit" name="_ctl1" value="  =  " />
      <span id="Sum"></span>
    </form>
  </body>
</html>
```


Summary

- Completed pages 177-193