

# 单元5-

## ASP.NET状态管理与常用对象

1. ASP.NET的状态管理
2. ASP.NET常见系统对象I  
实作1-在线测验功能I
3. ASP.NET常见系统对象II  
实作2-在线测验功能II



# 1. ASP.NET的状态管理



# ASP.NET的状态管理

- ASP.NET 技术基于**HTTP协议**，是一种**“无状态的”**的网页连接机制。服务器处理客户端请求的网页后，与该客户端的连接就中断了，也就是服务器不会保存客户端数据。
- 在ASP.NET环境中**需要使用与状态管理相关的对象**来保存用户数据。使再次请求页面和本次请求之间的关系和相关数据得以持续。
- 具体而言，“状态管理”是指**使用ViewState、Cookie、Session和Application等对象实现页面数据缓存和传递**的技术。
- ASP.NET 页面中包含一系列的对象，在页面中可以直接使用，可以将它们称为**系统对象或内置对象**。



# ASP.NET的状态管理

- **客户端状态管理**：将状态数据保存在客户端，当客户端向服务器发送请求时，状态数据会随之发送到服务器端。有ViewState、ControlState、HiddenField、Cookie和查询字符串等可选。
- **服务器状态管理**：将状态数据保存在服务器上。有Session、Application或数据库形式等方式可选。
- 客户端状态不消耗服务器内存资源，但容易泄露数据信息。而服务器端状态将消耗服务器端内存资源，但具有较高的安全性。



## 2. ASP.NET常见系统对象I



# ASP.NET 的系统对象介绍

对象名称	说 明
<b>Page</b>	指向页面自身的方式
<b>Response</b>	封装了Web服务器向浏览器发送的HTTP响应信息。
<b>Request</b>	封装了浏览器向Web服务器发送的HTTP请求信息。
<b>Cookie</b>	用来保存特定用户相关的信息，数据保存在客户端。
<b>Server</b>	用于获取服务器的相关信息的对象。
<b>Session</b>	用来保存特定用户相关的信息，数据保存在服务器端。
<b>Application</b>	代表运行在Web服务器上的ASP.NET应用程序。



# 系统对象被运用的例子

在一个购物网中需要多个页面合作完成商品交易：登录页、商品浏览页、详细商品页、购物车页、结帐页等。（至少用了5个系统对象）

- 在应用程序启动时：透过Global.aspx将商品目录放入Application变项，供所有用户共享。
- 在登录页：透过Cookie找到上回登录信息，直接进入系统。并将用户信息放入Session变项中。
- 在商品浏览页：在Application变项中取出商品信息并呈现，当点击某项商品时透过Response.Redirect() 将商品ID传入详细商品页
- 在详细商品页：以Request.QueryString() 获取商品ID，以调出商品信息。客户满意时将它加入购物车，加入时需用户ID，在Session变项中取得。
- 在购物车页：在Session中取出用户ID，呈现商品，在用户确认后进入结帐。
- 在结帐页：在Session中取出用户ID，填写地址，在用户确认后进入付款页。



# Page对象

- ASP.NET 页面都继承自 `System.Web.UI.Page` 类，所以每个 ASP.NET 页面在运行时就是一个 Page 对象。
- Page 对象提供对当前页面对象的访问，也包括大部份的系统对象。
- @Page指令（位于.aspx）
  - Language:** 页面中代码使用的语言，通常是C#语言。
  - AutoEventWireup:** 页面与事件程序的关联自动设置，默认为 `true`。
  - CodeBehind:** 后置代码的文件名，后置代码与页面合成一完整的类。
  - Inherits:** 用于指定后置代码中页面类的名称。

```
<%@ Page Title="About" Language="C#" MasterPageFile="~/Site.Master"  
AutoEventWireup="true" CodeBehind="About.aspx.cs" Inherits="uBooksOnline.About" %>
```





# Page对象

## Page对象常用成员

成员名称	说 明
<b>IsPostBack</b> 属性	bool值，指示该页面是否为响应客户端回发而加载，或是首次加载
<b>IsValid</b> 属性	bool值，指示该页面验证是否成功
<b>Load</b> 事件	Page_Load 事件发生于页面被载入时，常用于初值设置
<b>MasterPageFile</b> 属性	用于获取或设置母版页的文件名

```
//下面语句置入Page_Load事件中
if (!IsPostBack) {
    //此区块只有第1次加载会被运行,适合作初值设置
}
```



# Response对象

- Response对象用于将数据从服务器发送回浏览器，并提供有关响应的信息。常页面复位向、保存Cookie等。
- Page类的Response属性是HttpResponse类的一个实例对象。

成员名称	说 明
<b>Write()</b> 方法	将指定的内容输出至页面（浏览器）
<b>Redirect()</b> 方法	将页面复位向到另一页面
<b>Cookies</b> 属性	添加或修改客户端的Cookie
<b>WriteFile()</b> 方法	将指定的文件写入输出流
<b>End()</b> 方法	终止页面的执行
<b>ContentType</b> 属性	输出流的MIME类型，默认是“text/html”



# Response对象

范例:



```
//用write方法输出当前时间
Response.Write("当前时间是:" + DateTime.Now.ToString());

for (int i = 0; i < 20; i++) {
    Response.Write(i.ToString() + " ");
    //到10结束输出,亦造成原本的Button等也未能出现.
    if (i == 10) Response.End();
}
```

```
//跳转到Request.aspx页并传递参数n
Response.Redirect("Request.aspx?n=12");
```

Request.aspx页面

当前时间是: 2022/2/4 14:19:47  
0 1 2 3 4 5 6 7 8 9 10

Source: Examples/Unit05/Response.aspx



# Request对象

- Request封装了浏览器向Web服务器发送的HTTP请求信息。Request属性能控制请求数据，如浏览器信息、查询字符串、Cookie等信息。
- Page类的Request属性值是HttpRequest类的一个实例对象，它封装了HTTP请求信息。

成员名称	说 明
<b>QueryString</b> 属性	从查询字符串中读取用户提交的数据
<b>Cookies</b> 属性	获取客户端的Cookie数据
<b>ServerVariables</b> 属性	获得服务器端或客户端的环境变量信息
<b>Browser</b> 属性	获得客户端浏览器信息
<b>RequestType</b> 属性	请求是Get还是Post请求
<b>IsSecureConnection</b> 属性	HTTP连接是否使用加密



# Request对象

## ● 范例:

### Request.aspx

#### Request对象使用范例:

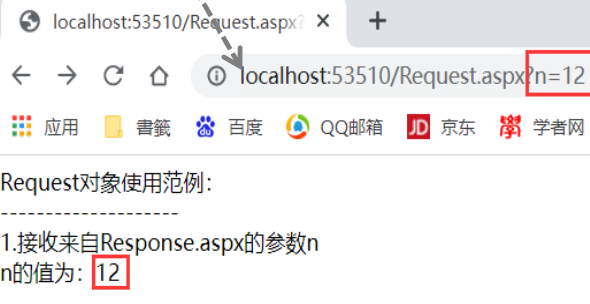
1. 接收来自Response.aspx的参数n  
n的值为: 显示n的值

2. Request对象属性测试  
Request属性值

```
Page_Load() {  
    lblShow.Text = Request.QueryString["n"];  
}
```

#### 2. Request对象属性测试

服务器名称: localhost  
服务器端口: 53510  
浏览器类型: Chrome  
浏览器版本: 97.0  
是否支持Cookies: true  
浏览器所在的平台WinNT  
请求型态: POST  
是否使用加密: False



"服务器名称: " + Request.ServerVariables["SERVER\_NAME"] ;  
"服务器端口: " + Request.ServerVariables["SERVER\_PORT"] ;  
"浏览器类型: " + Request.Browser["Browser"] ;  
"浏览器版本: " + Request.Browser["Version"] ;  
"是否支持Cookies: " + Request.Browser["Cookies"] ;  
"浏览器所在的平台: " + Request.Browser.Platform.ToString();  
"请求型态: " + Request.RequestType ;  
"是否使用加密: " + Request.IsSecureConnection;

Source: Examples/Unit05/Request.aspx



# Cookie对象

- Cookie就是Web服务器保存在用户硬盘上的一段文本，且它允许服务器随后再取回它。信息的片断以‘键/值’对的形式存储。
- Response对象支持一个名为Cookies的集合，可以将Cookie对象添加到该集合中，从而向客户端输出Cookie。
- 通过Request对象的Cookies集合来访问Cookie。
- Cookie对象为System.Web.HttpCookie类的实例。
- Cookie应用的几条限制：
  - 大多数浏览器支持最多可达4096字节的Cookie。
  - 浏览器限制站点可在用户端保存的Cookie数，如每个站保存20个Cookie。
  - 用户可以设置自己的浏览器，拒绝接受Cookie。



# Cookie对象

成员名称	说 明
<b>Expires</b> 属性	获取或设置Cookie的过期日期和时间
<b>Name</b> 属性	获取或设置Cookie的名称
<b>Value</b> 属性	获取或设置一个Cookie值
<b>Add()</b> 方法	增加一个Cookie变量
<b>Clear()</b> 方法	删除Cookie集合内的变量
<b>Get()</b> 方法	通过变量名取得Cookie值
<b>Remove()</b> 方法	通过Cookie变量名来删Cookie变量



# Cookie对象

范例：（先CookieLogin.aspx写入Cookie）

CookieLogin.aspx

localhost:53510/CookieLogin.a x +

localhost:53510/CookieLogin.aspx

应用 书签 百度 QQ邮箱 JD 京东 学者网

Cookie对象使用范例:

帐号: 张三

密码: ....

登录

```
protected void btnLogin_Click() {  
    if(txtName.Text != "" && txtPassword.Text != "") {  
        //创建一个名为name的cookie  
        HttpCookie cookie = new HttpCookie("name");  
        //使用Server.UrlEncode可处理中文  
        cookie.Value = Server.UrlEncode(txtName.Text.Trim());  
        //设定一天过期  
        cookie.Expires = DateTime.Now.AddDays(1);  
        //写入客户端  
        Response.Cookies.Add(cookie);  
        Response.Redirect("CookieHome.aspx");  
    }  
}
```

CookieHome.aspx

Source: Examples/Unit05/CookieLogin.aspx





# Cookie对象

- 范例：（若曾写入Cookie可直接访问CookieHome.aspx）

## CookieHome.aspx

```
protected void Page_Load() {  
    if (Request.Cookies["name"] != null) {  
        string name = Server.UrlDecode(Request.Cookies["name"].Value);  
        lblMessage.Text = name + ", 欢迎您的访问!";  
    } else {  
        Response.Redirect("CookieLogin.aspx");  
    }  
}
```

Source: Examples/Unit05/CookieHome.aspx





# 实作1-在线测验功能I

Source:

Labs/Unit05/Question-1.aspx

Labs/Unit05/Question-2.aspx



# 实作1-在线测验功能I

- 实作说明：本实作以查询字符串/Request/Response及Cookie实作在线测验功能
- 实作步骤：
  1. 方案 / 项目：LabASPNET，在Labs/Unit05中添加「Web窗体」->命名为**Question-1.aspx及Question-2.aspx**，页面设计分别如次页。（Unit05请添加在Labs下）
  2. 在Question-1.aspx显示试题，并将第1题用户答案以查询字符串传到下一题。Question-2.aspx显示第2题，并用Request接收第1题答案后，以Cookie方式将第1,2题答案放入Cookie，以备下个实作取出。



# 实作1-在线测验功能I

Question-1.aspx

实作-在线测验 (查询字符串/Response)

Q1: 请问下列那一个不是Microsoft的技术?

- ☐ ASP
- ☐ ASP.net
- ☐ PHP
- ☐ ASP.net MVC

下一题 请选择答案

RadioButtonList

RequiredFieldValidator

Source: Labs/Unit05/Question-1.aspx

Question-2.aspx

实作-在线测验 (Request/Response/Cookie)

Q2: 请问下列那一项不是服务器端网页技术?

- ☐ JavaScript
- ☐ ASP.net
- ☐ Servlet
- ☐ ASP.net MVC

下一题 请选择答案

Source: Labs/Unit05/Question-2.aspx



# 实作1-在线测验功能I

localhost:59797/Question-1.aspx

← → ↻ 🏠 localhost:59797/Question-1.aspx

应用 書籤 百度 QQ邮箱 JD 京东 学者网

**实作-在线测验(查询字符串/Response)**

Q1: 请问下列那一个不是Microsoft的技术?

- ☐ ASP
- ☐ ASP.net
- ☐ PHP
- ☐ ASP.net MVC

未选答案出现提示

下一题 请选择答案

localhost:59797/Question-1.aspx

← → ↻ 🏠 localhost:59797/Question-1.aspx

应用 書籤 百度 QQ邮箱 JD 京东 学者网

**实作-在线测验(查询字符串/Response)**

Q1: 请问下列那一个不是Microsoft的技术?

- ☐ ASP
- ☐ ASP.net
- ☒ PHP
- ☐ ASP.net MVC

下一题



localhost:59797/Question-2.aspx?Q1=PHP

← → ↻ 🏠 localhost:59797/Question-2.aspx?Q1=PHP

应用 書籤 百度 QQ邮箱 JD 京东 学者网 Coc

**实作-在线测验(Request/Response/Cookie)**

Q2: 请问下列那一项不是服务器端网页技术?

- ☒ JavaScript
- ☐ ASP.net
- ☐ Servlet
- ☐ ASP.net MVC

网址出现查询字符串

下一题



### 3. ASP.NET常见系统对象II



# Server对象

- Page类的Server属性（HttpServerUtility类）提供了对服务器上的方法和属性的访问。

成员名称	说 明
<b>MachineName</b> 属性	获取服务器的名称
<b>ScriptTimeout</b> 属性	设置页面执行的最长时间，默认90秒
<b>Execute()</b> 方法	停止执行当前页面，转到并且执行新页面，执行完毕后返回原页面，继续执行后续语句
<b>HtmlEncode()</b> <b>/ HtmlDecode()</b> 方法	HtmlEncode() 将字符串中的HTML标记进行编码，如将“<”转换为“&lt”;HtmlDecode() 对HTML编码字符串进行解码
<b>MapPath()</b> 方法	获取页面的物理路径



# Server对象

成员名称	说 明
<b>Transfer()</b> 方法	停止执行当前页面，转到并且执行新页面，执行完毕后不再返回原页面
<b>UrlEncode()</b> <b>/ UriDecode()</b> 方法	UrlEncode () 将字符串中某些特殊字符转换为URL编码，如将 “/” 转换为 “%2f”，空格转换为 “+” ;UriDecode () 为解码
<b>UriPathEncode()</b> 方法	对URL字符串的路径部份进行URL编码, 并返回编码字符串





# Server对象

- Response.Redirect()、Server.Execute()和Server.Transfer()的区别
  - **地址变化：** Redirect()方法复位向发生在客户端，可从浏览器地址栏中看到地址变化；而Execute()和Transfer()方法的复位向实际发生在服务器端，在浏览器的地址栏中看不到地址变化。
  - **页面变化：** Redirect()和Transfer()方法执行完新页面后，并不返回原页面；而Execute()方法执行完新页面后会返回原页面继续执行。
  - **网站变化：** Redirect()可复位向到同一网站的不同页面，也可到其它网站的页面；而Execute()和Transfer()只能复位向到同一网站的不同页面。

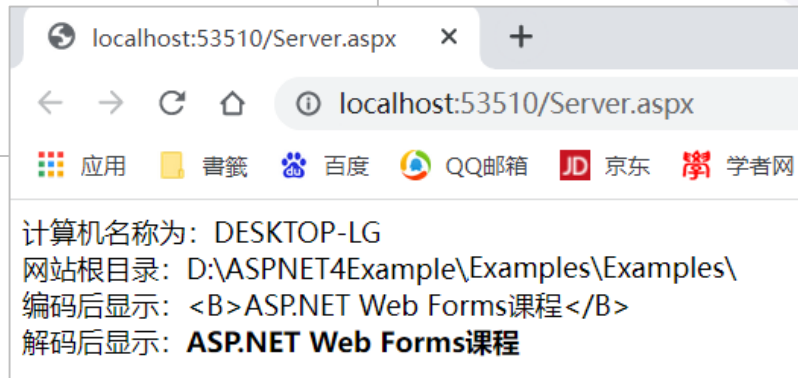


# Server对象

范例:

```
protected void Page_Load(object sender, EventArgs e) {  
    //获取计算机名  
    string serverName = Server.MachineName.ToString();  
    Response.Write("计算机名称为: " + serverName);  
    //页面的物理路径-网站根目录  
    Response.Write("网站根目录: " + Server.MapPath("/"));  
    //HTML编码和解码  
    string str = Server.HtmlEncode("<B>ASP.NET Web Forms课程</B>");  
    Response.Write("编码后显示: " + str);  
    str = Server.HtmlDecode(str);  
    Response.Write("解码后显示: " + str);  
}
```

Source: Examples/Unit05/Server.aspx





# Session对象

- 用来保存特定用户相关的信息，其数据保存在服务器端。
- 通过SessionID来区别不同用户，并将一个有限时间窗口且来自同一浏览器的请求标示为一会话。
- 若客户端支持Cookie，则将Session ID保存到相应的Cookie中；若不支持，就将Session ID添加到URL中。
- 常用于存储用户信息、多页面间的信息传递、购物车等功能的实现。
- Page的Session属性（对象）实际上为HttpSessionState类的实例。



# Session对象

成员名称	说 明
<b>SessionID</b> 属性	用于标识会话的唯一编号
<b>TimeOut</b> 属性	会话状态持续时间，单位为分钟，默认为20分钟
<b>Count</b> 属性	获取会话状态下Session对象的个数
<b>Abandon()</b> 方法	取消当前会话
<b>Remove()</b> 方法	删除会话状态集合中的项
<b>Add()</b> 方法	向当前会话状态集合中添加一个新项
<b>Clear()</b> 方法	清空当前会话状态集合中所有键和值



# Session对象

范例：（先SessionLogin.aspx，若先SessionHome.aspx则会重返SessionLogin.aspx）

localhost:53510/SessionLogin.aspx

应用 书籤 百度 QQ邮箱 JD 京东 学者网

Session对象使用范例:

帐号:

密码:

```
protected void btnLogin_Click() {  
    if(txtName.Text != "" && txtPassword.Text != "") {  
        Session["name"] = txtName.Text;  
        Session["password"] = txtPassword.Text;  
        Response.Redirect("SessionHome.aspx");  
    }  
}
```

Examples/Unit05/SessionLogin.aspx

```
protected void Page_Load() {  
    if (Session["name"] != null) {  
        lblMessage.Text = Session["name"] + " 欢迎您! ";  
    } else {  
        Response.Redirect("SessionLogin.aspx");  
    }  
}
```

Examples/Unit05/SessionHome.aspx

localhost:53510/SessionHome.aspx

应用 书籤 百度 QQ邮箱 JD 京东 学者网

会员首页

李四 欢迎您!



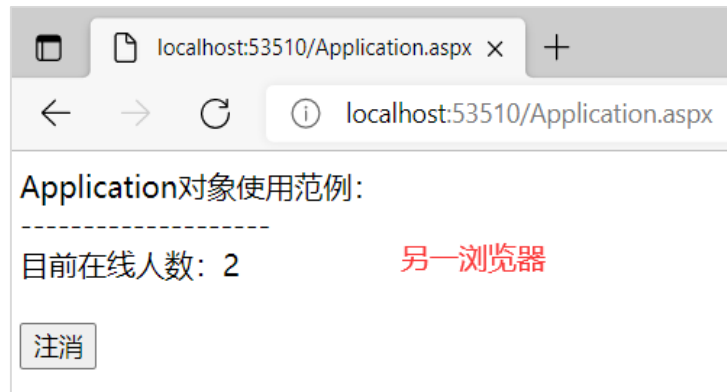
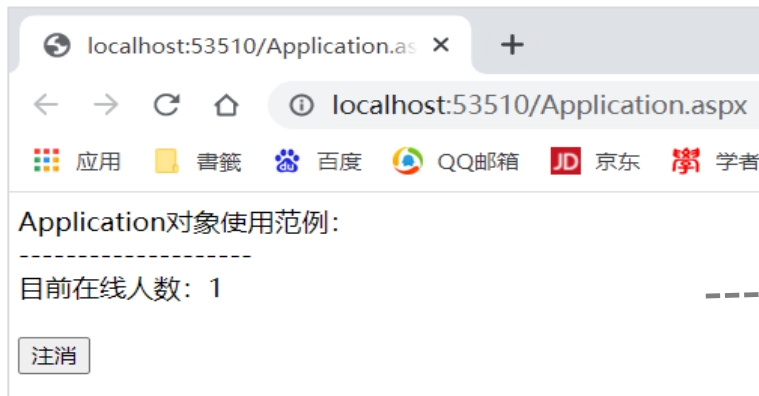
# Application对象

- Application为Page属性，是HttpApplicationState类的实现。
- 定义ASP.NET应用程序中的所有应用程序对象通用的方法、属性和事件。
- 主要的方法：Application\_Start()、Application\_End()方法包含于Global.asax。
- 要修改Application状态值时，首先要调用Application.Lock()方法锁定Application状态，值修改后再调用Application.Unlock()方法解除锁定。



# Application对象

范例:



Application.aspx.cs

```
protected void Page_Load() {  
    lblShow.Text = Application["Visitors"].ToString();  
}
```

Source: Examples/Unit05/Application.aspx



# Global.asax文件

## Global.asax

```
public class Global : System.Web.HttpApplication {  
    void Application_Start(object sender, EventArgs e) {  
        //初始化访客计数器, 应用程序启动时为0  
        Application["Visitors"] = 0;  
    }  
    void Application_End(object sender, EventArgs e) {  
        //应用程序关闭时要执行的代码放此  
    }  
    void Session_Start(object sender, EventArgs e) {  
        //访客上网,人数+1  
        if (Application["Visitors"] != null) {  
            Application.Lock();  
            Application["Visitors"] = (int)Application["Visitors"] + 1;  
            Application.Unlock();  
        }  
    }  
}
```

```
void Session_End(object sender,  
EventArgs e) {  
    //访客离线,人数-1  
    //要引发Session_End事  
件,sessionstate模式需采用默认的  
InProc模式  
    if (Application["Visitors"] != null) {  
        Application.Lock();  
        Application["Visitors"] =  
(int)Application["Visitors"] - 1;  
        Application.Unlock();  
    }  
}
```

Source: LearnASPNET/Global.asax





## 实作2-在线测验功能II

Source:

Labs/Unit05/Question-3.aspx

Labs/Unit05/Result.aspx



## 实作2-在线测验功能II

- ◆ 实作说明：本实作以Cookie及Session实作在线测验功能
- ◆ 实作步骤：
  1. 方案 / 项目：LabASPNET，在Labs/Unit05中添加「Web窗体」->命名为**Question-3.aspx及Result.aspx**，页面设计分别如次页。。
  2. 在Question-3.aspx中将传过来的第1,2题答案及本题选答，均存入Session变项，并传到Result.aspx。Result.aspx从Session变项中取出答案计分并呈现分数及正确答案。



## 实作2-在线测验功能II

Question-3.aspx

实作-在线测验 (Session)

Q3: 请问下列那一个不是客户端的网页技术?

☐ JavaScript

☐ Ajax

☐ jQuery

☐ JSP

下一题 请选择答案

RadioButtonList

RequiredFieldValidator

Source: Labs/Unit05/Question-3.aspx

Result.aspx

实作-在线测验 (结果)

Label

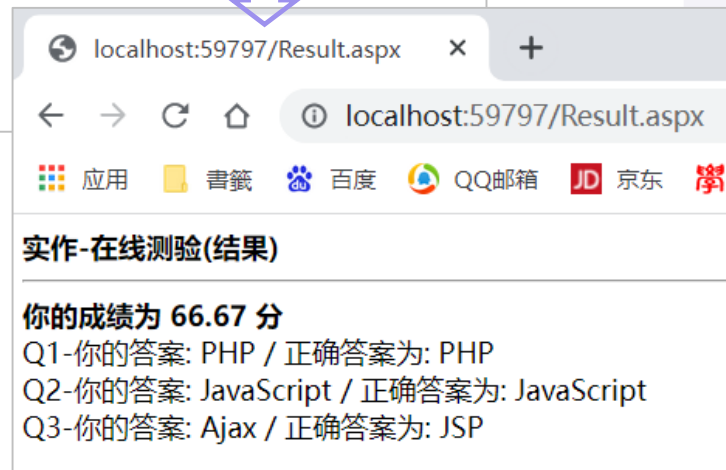
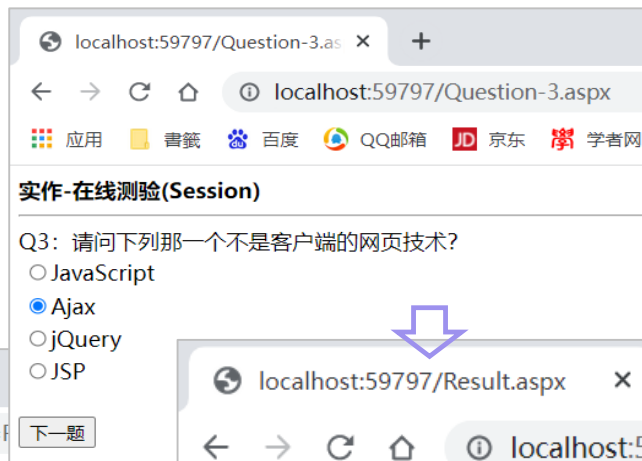
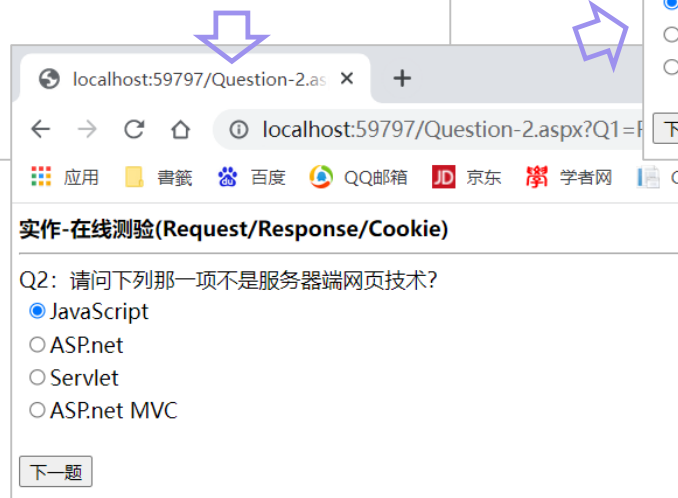
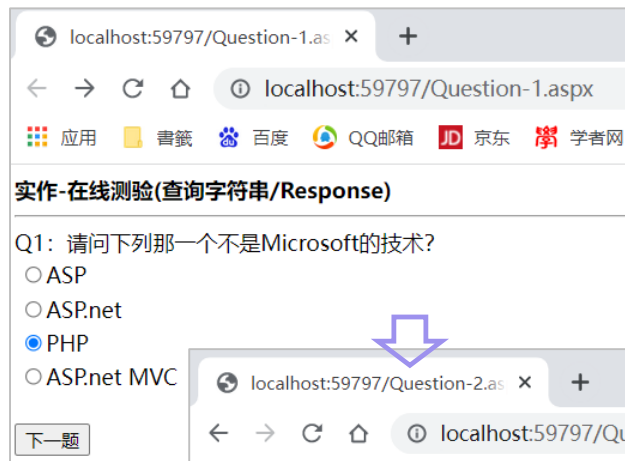
Label

两个Label由后置代码控制  
并呈现分数及正确答案信息

Source: Labs/Unit05/Result.aspx



## 实作2-在线测验功能II





## 单元小结

- 本单元介绍了ASP.NET的状态管理及常用的系统对象，这些对象包含：Page、Response、Request、Server、Session、Cookie及Application 等。
- 大部份的系统对象都在处理一个Web程序的状态管理，互相合作以完成各项Web功能需求。
- 系统对象在状态管理时大致分为两类：客户端状态管理及服务器状态管理。前者如常用的Cookie和查询字符串技术可选，后者有Session、Application或数据库形式等方式可选。在运用时应考虑应用程序需求、客户端及服务器的情况后再行决定。