

بسم الله الرحمن الرحيم

## آموزشگاه تحلیل داده

ایران در دیتابیس و نویسی برنامه مرکز ترین تخصصی

## آموزش کنترل های Server Side

مدرس : مهندس افشین رفوآ

### آموزش کنترل های Server Side

درمباحث پیشین با **Page lifecycle** (چرخه ی حیات صفحه) و اینکه چگونه یک صفحه **control** های مختلف را دربر می گیرد آشنا شدیم. **page** خود به صورت یک شیء **control (control object)** نمونه سازی می شود (**= instantiate**) ایجاد یک نمونه از کلاس). کلیه **web form** ها در اصل نمونه هایی از کلاس **ASP.NET Page** می باشند. کلاس **Page** خاصیت های (**property**) بسیار سودمند و کارامدی دارد که در زیر برای شما فهرست شده اند :

Session

Application

Cache

Request

Response

Server

User

Trace

در این مبحث تنها به تشریح اشیاء **Server**، **Request** و **Response** می پردازیم.

### ❖ Server Object (شیء server)

شیء **Server** (در **ASP.NET**) در حقیقت نمونه ی (instance) از کلاس **System.Web.HttpServerUtility** می باشد. کلاس **HttpServerUtility** خاصیت ها (property) و توابع (method) فراوانی ارائه می دهد که هر یک برنامه ها و کارهای (job) متفاوتی را انجام می دهند (منظور از job حجم معینی از پردازش هاست که به صورت واحد توسط یک کامپیوتر اجرا می شوند).

### خاصیت ها و توابع شیء Server

**ASP.NET** متدها و خاصیت های کلاس **HttpServerUtility** را داخل شیء درونی **Server (intrinsic object)** کپسوله سازی می کند.

جدول زیر فهرستی از این خاصیت ها (property) را ارائه می دهد :

خاصیت	شرح
MachineName	اسم کامپیوتر server / اسم رایانه ی سرویس دهنده را برمی گرداند.
ScriptTimeout	مدت زمان time-out درخواست (request) را بر حسب ثانیه گرفته و تنظیم (get) و (set) می کند.

جدول زیر فهرستی از توابع (method) را به نمایش می گذارد :

متد	شرح
CreateObject(String)	یک نمونه از شیء COM ایجاد کرده که توسط ProgID شناسایی می شود.
CreateObject(Type)	نمونه ای از شیء COM را ایجاد کرده که توسط نوعش شناسایی می شود.

Equals(Object)	تعیین می کند آیا شیء مشخص شده با شیء کنونی یکسان است یا خیر.
Execute(String)	. این متد در حقیقت درخواست هایی که برای یک آدرس مجازی ارسال می شود را اجرا می کند. (یک فایل asp را از داخل یک فایل asp دیگر اجرا می کند).
Execute(String, Boolean)	درخواست هایی که برای آدرس مجازی ارسال شده اند را اجرا کرده و تعیین می کند آیا collection های QueryString و form پاک شوند یا خیر.
GetLastError	آخرین استثنا (exception) را برمی گرداند.
GetType	نوع نمونه ی جاری را برمی گرداند.
HtmlEncode	یک رشته ی معمولی را به رشته ای با کاراکترهای استاندارد HTML تغییر می دهد (تبدیل می کند).
HtmlDecode	یک رشته ی html را به یک رشته ی معمولی تبدیل می کند.
ToString	شیء جاری را به صورت یک رشته برمی گرداند.
Transfer(String)	متد transfer اجرای صفحه ی فعلی را خاتمه داده و تمامی اطلاعات مربوط به وضعیت (state) جاری را به صفحه دیگری که در URL مشخص شده ارسال می کند. برخلاف متد execute، اجرا (execution) در صفحه ای که متد transfer را فرامی خواند از سر گرفته نمی شود.
UrlDecode	یک رشته ی URL را به یک رشته ی معمولی تبدیل می کند.
UrlEncodeToken	درست مشابه UrlEncode عمل می کند اما روی یک آرایه ی باینری که در بردارنده داده هایی است که با base64 کدگذاری شده اند

UrlDecodeToken	درست مشابه متد UriDecode کار می کند اما روی یک آرایه ی بایتی ( byte array ) دربردارنده ی داده هایی که با base64 کدگذاری شده اند (base64-encoded data).
MapPath	مسیر فیزیکی را برمی گرداند که با مسیر مجازی فایل روی رایانه ی سرویس دهنده (sever) همخوانی دارد. این تابع یک مسیر مشخص شده را به مسیر فیزیکی نگاشت (map) می کند.
Transfer	اجرا (execution) را در برنامه ی جاری به یک صفحه ی وب دیگر انتقال می دهد.

## شیء Request

شیء Request در واقع نمونه ای از کلاس **System.Web.HttpRequest** است. شیء request درخواست هایی را که مرورگر سرویس گیرنده (client) طی یک **HTTP request** به سرور ارسال شده را بازیابی (retrieve) می کند. این شیء به منظور دریافت اطلاعات از کاربر بکار گرفته می شود. با بهره گیری از این شیء شما می توانید یک صفحه ی وب به صورت پویا ایجاد کرده و عملیات متفاوت سمت سرور را بر پایه ی ورودی (input) دریافتی از کاربر انجام دهید.

## متدها و خاصیت های شیء Request

جدول زیر برخی از این خواص و متدها را فهرست می کند :

Property(خاصیت)	شرح
AcceptTypes	آرایه ی رشته ای از MIME accept type ها را که توسط سرویس گیرنده (client) پشتیبانی می شود را برمی گرداند.
ApplicationPath	مسیر مجازی برنامه را از root آن محاسبه می کند.
Browser	اطلاعاتی درباره ی قابلیت های مرورگر کلاینت درخواست دهنده را گرفته (get) یا تنظیم (set) می کند.

ContentEncoding	این خاصیت مجموعه کاراکترهای بدنه ی موجودیت (entity body) را خوانده (get) و قرار می دهد (set) .
ContentLength	طول content ارسال شده از سرویس گیرنده را برحسب بایت مشخص می کند.
ContentType	خاصیت MIME content type که از درخواست آمده را تنظیم می کند.
Cookies	Collection ای از cookie ها که توسط سرویس گیرنده ارسال شده را برمی گرداند.
FilePath	مسیر مجازی درخواست (request) جاری را برمی گرداند.
Files	مجموعه ای (collection) از فایل های آپلود شده توسط سرویس گیرنده را با فرمت چند بخشه ی MIME برمی گرداند .
Form	مجموعه ای از متغیرهای form را برمی گرداند.
Headers	Collection ای از سراینده های (HEADER) HTTP را برمی گرداند.
HttpMethod	روش ارسال داده های http که توسط کلاینت مورد استفاده قرار می گیرد را (اعم از GET ,POST و یا HEAD) برمی گرداند.
InputStream	محتوایی (content) را که از طریق http entity body آمده را برمی گرداند.
IsSecureConnection	یک مقدار را برمی گرداند که نشان می دهد آیا اتصال http از socket های امن استفاده می کند یا خیر.
QueryString	آرایه ای (collection) از متغیرهای HTTP query string برمی گرداند.
RawUrl	url خالص درخواست جاری را برمی گرداند.
RequestType	نشانگر روش ارسال اطلاعات از کلاینت به سرور (post و get) می باشد.
ServerVariables	آرایه ای از متغیرهای سرور برمی گرداند.

TotalBytes	تعداد بایت های مربوط به stream جاری را بازمی گرداند.
Url	اطلاعاتی را درمورد url درخواست (request) جاری برمی گرداند.
UrlReferrer	اطلاعاتی درباره ی url درخواست (request) قبلی کلاینت که به url جاری متصل می باشد را برمی گرداند.
UserAgent	رشته ای شامل agent مرورگر کاربر را بر می گرداند.
UserHostAddress	IP سرویس گیرنده را بر می رگداند.
UserHostName	نام dns کاربر remote را برمی گرداند.
UserLanguages	آرایه ی رشته ای مرتب شده از زبان تنظیم شده ی سرویس گیرنده برمی گرداند.

جدول زیر برخی از متدهای مهم را ارائه می دهد :

تابع	شرح
BinaryRead	این متد به منظور بازیابی داده ارسال شده به سرور از کلاینت (به عنوان بخشی از post request) بکار گرفته می شود. متد مزبور داده ها را در یک SafeArray ذخیره می کند. Safearray آرایه ای است که اطلاعاتی درباره ی تعداد بعدها (dimension) و حدود (bound) آرایه ذخیره می کند.
Equals(Object)	بررسی می کند آیا شی مشخص شده با شی کنونی یکسان است یا خیر.
GetType	نوع نمونه ی جاری را برمی گرداند.
MapImageCoordinates	پارامتر image-field form را به مقادیر مختصات X و y نگاشت می کند.
MapPath(String)	مسیر مجازی مشخص شده را به مسیر فیزیکی نگاشت می کند.
SaveAs	HTTP request را روی دیسک ذخیره می کند.

Tostring	شیء جاری را به صورت رشته نمایش می دهد (برمی گرداند).
ValidateInput	collection هایی که از طریق خاصیت های QueryString، Form، Cookie ها به آن ها دسترسی صورت گرفته را اعتبار سنجی می کند.

## شیء Response

**Response object** نشانگر پاسخ (**response**) سرور به درخواست سرویس گیرنده (**client request**) می باشد. این شیء نمونه ای از کلاس **System.Web.HttpResponse** می باشد.

در **ASP.NET**، شیء **response** هیچ نقش اساسی در ارسال متن **HTML** به سرویس گیرنده ایفا نمی کند، زیرا که کنترل های سمت سرور (**server-side controls**) دارای متدهای تودرتو (**nested**) و شیء گرا ویژه ی **render** کردن خود هستند.

با این وجود، شیء **HttpResponse** قابلیت ها و امکانات مهمی ارائه می دهد که از جمله آن ویژگی **cookie** و تابع **Redirect()** می باشد. متد **Response.Redirect()** امکان انتقال کاربر را هم داخل برنامه و هم خارج از آن به یک صفحه ی دیگر فراهم می کند.

## خاصیت ها و توابع شیء Response

جدولی که زیر مشاهده می کنید برخی خواص (**property**) کارآمد **response object** را فهرست کرده است :

خاصیت	شرح
Buffer	این خاصیت تعیین می کند آیا خروجی صفحه موقت ذخیره (buffer) شود یا خیر. چنانچه روی <b>true</b> تنظیم شده باشد، در آن صورت خروجی صفحه تا زمانی که اسرکیپت روی آن صفحه پردازش نشده باشد به کلاینت ارسال نمی گردد (و یا تا زمانی که متدهای <b>End</b> و <b>Flush</b> فراخوانی نشده اند (خروجی) فرستاده نمی شود).
BufferOutput	این خاصیت مقداری را <b>get</b> و <b>set</b> می کند که مشخص می کند آیا خروجی را موقت ذخیره (buffer) کرده و آن را پس از اینکه کل صفحه کاملاً پردازش شد ارسال کند یا خیر.

Charset	مجموعه کاراکترهای http stream خروجی را خوانده (get) و (set) مقداردهی می کند.
ContentEncoding	مجموعه کاراکتر http stream خروجی را get یا set می کند.
ContentType	HTTP MIME type استریم (stream) خروجی را get یا set می کند.
Cookies	Gets the response cookies collection Cookie collection پاسخ (response) را برمی گرداند.
Expires	تعداد دقیقه هایی که پس از سپری شدن آن یک صفحه موقت ذخیره شده در مرورگر باید منقضی شود را تنظیم می کند.
ExpiresAbsolute	تاریخ و زمان دقیق و مطلق که اطلاعات موقت ذخیره شده (cached) باید از حافظه ی پنهان (cache) حذف شوند را مقدار دهی می کند.
HeaderEncoding	encoding object را که بیانگر encoding برای header output stream است را مقداردهی می کند.
Headers	مجموعه سرایندهای (header) response را برمی گرداند.
IsClientConnected	مقداری را برمی گرداند که نشان می دهد آیا سرویس گیرنده هنوز به سرویس دهنده (server) متصل هست یا خیر.
Output	متن بازگشتی را به http response stream می دهد.
OutputStream	خروجی دودویی (binary output) را به بدنه ی محتوا (content body) http خروجی (خارج شونده) می دهد.
RedirectLocation	مقدار سرایند http Location header را تعیین می کند
Status	Status line ای که به سرویس گیرنده ارسال می گردد را تنظیم (set) می کند.



StatusCode	http status code خروجی (output) که به سرویس گیرنده برگردانده می شود را گرفته (get) و تنظیم (set) می کند.
StatusDescription	این خاصیت، http Status string خروجی (output) که به سرویس گیرنده ارسال می گردد را مقداردهی (get و set) می کند.
SubStatusCode	مقداری را خوانده یا گرفته که این مقدار status code (کد وضعیت) را تعریف می کند.
SuppressContent	مقداری را خوانده و یا گرفته، که این مقدار تعیین می کند آیا http content به سرویس گیرنده ارسال گردد یا خیر.

جدول زیر برخی از متدهای مهم را فهرست می کند :

Method	Description
AddHeader	یک سرآیند (header) http به output stream اضافه می کند. این متد به منظور ایجاد سازگاری با ویرایش های پیشین ASP ارائه شد.
AppendCookie	زیرساخت (infrastructure)، یک cookie به cookie collection درونی (intrinsic) اضافه می کند.
AppendHeader	یک سرآیند (header) http را به output stream اضافه می کند.
AppendToLog	اطلاعات ثبت وقایع (log information) سفارشی را به فایل گزارشات ( log file) سرویس های اطلاعات اینترنتی (IIS) اضافه می کند.
BinaryWrite	رشته ای از کاراکترهای دودویی (binary characters) را در http output stream می نویسد.
ClearContent	تمام محتوا (content) خروجی را از buffer stream پاک می کند.
Close	اتصال سمت سرویس گیرنده را می بندد (قطع می کند).

End	کلیه ی خروجی هایی که در حال حاضر به طور موقت ذخیره شده اند را به سرویس گیرنده ارسال کرده، اجرای صفحه را متوقف کرده سپس رخداد EndRequest را اجرا (فعال) می کند.
Equals(Object)	بررسی می کند آیا شیء مشخص شده با شیء جاری یکسان (برابر) هستند یا خیر.
Flush	تمامی خروجی هایی که در حال حاضر موقتاً ذخیره (buffer) شده اند را به سرویس گیرنده ارسال می کند.
GetType	این تابع نوع نمونه ی جاری را برمی گرداند.
Pics	این متد سرایند (header) HTTP PICS-Label را به stream خروجی الحاق (append) می کند.
Redirect(String)	درخواست (request) را به یک url جدید تغییر مسیر (redirect) داده و url جدید را مشخص می کند.
Redirect(String, Boolean)	سرویس گیرنده را به url جدید ارجاع (redirect) می دهد. این متد url جدید را مشخص کرده و همچنین تعیین می کند آیا اجرای صفحه ی جاری باید خاتمه داده شود یا خیر.
SetCookie	یک کوکی موجود را در cookie collection (مجموعه کوکی ها) بروز رسانی می کند.
ToString	شیء فعلی را به صورت رشته نمایش می دهد (برمی گرداند).
TransmitFile(String)	فایل مشخص شده را مستقیماً و بدون اینکه آن را در حافظه موقتاً ذخیره کند به HTTP response output stream می دهد .
Write(Char)	کاراکتر را در HTTP response stream قرار می دهد(می نویسد).
Write(Object)	یک شیء را در HTTP response stream قرار می دهد(می نویسد).
Write(String)	یک رشته را در HTTP response stream قرار می دهد(می نویسد).

WriteFile(String)	محتوا (content) فایل مشخص شده را مستقیماً داخل HTTP response output stream به صورت یک file block قرار می دهد (می نویسد).
WriteFile(String, Boolean)	محتوای فایل مشخص شده را مستقیماً در HTTP response output stream به صورت یک memory block (بلوک حافظه) می نویسد.

مثال :

مثال بسیار ساده ی زیر دارای یک کنترل **textbox** می باشد که کاربر می تواند در آن اسم خود را وارد کند، همچنین یک دکمه که اطلاعات را به سرور ارسال می کند و یک کنترل **label** که **url** کامپیوتر سرویس گیرنده (**server**) را نمایش می دهد.

: Content file

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="Asp.Net_ServerSide.Default" %>
    Inherits="server_side._Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
    <head runat="server">
        <title>Untitled Page</title>
    </head>
    <body>
        <form id="form1" runat="server">
            <div>
                Enter your name:
                <br />
                <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
                <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Submit" />
                <br />
                <asp:Label ID="Label1" runat="server"/>
            </div>
        </form>
    </body>
</html>
```

: code behind Button1\_Click

```
protected void Button1_Click(object sender, EventArgs e)
{
    if (!String.IsNullOrEmpty(TextBox1.Text))
    {
        // Access the HttpServerUtility methods through
        // the intrinsic Server object.
        Label1.Text = "Welcome, " + Server.HtmlEncode(TextBox1.Text) + ". <br/> The url is " +
            Server.UrlEncode(Request.Url.ToString());
    }
}
```

حال صفحه را اجرا کنید تا نتیجه زیر نمایش داده شود :

Enter your name:

Welcome, Steve.

The url is `http%3a%2f%2flocalhost%3a1478%2fDefault.aspx`