



# **ĐỔI TƯỢNG CỦA ASP.NET**

## **Thời lượng: 120 phút**

**GV: ThS. PHAN LONG**

**Email: [phanlong92@gmail.com](mailto:phanlong92@gmail.com)**

**Phone: 090 3333 080**

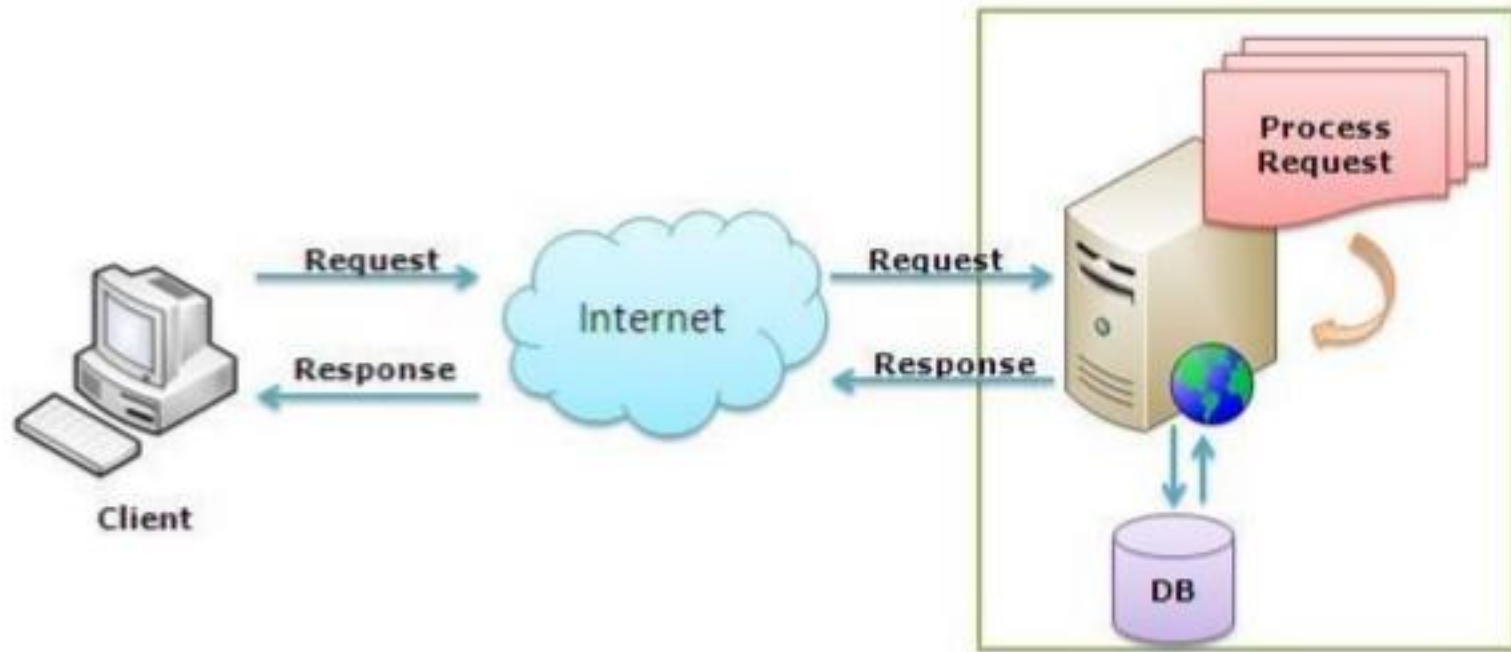


- ❖ *Đối tượng Response*
- ❖ *Đối Request*
- ❖ *Đối tượng Session*
- ❖ *Đối tượng Cookie*
- ❖ *Đối tượng Application*
- ❖ *Đối tượng Server*



# ĐỐI TƯỢNG REQUEST & RESPONSE

- Cho phép server gửi dữ liệu trở về trình duyệt như một đáp ứng, trả lời hay thông tin với client.
- Thuộc Lớp System.Web



Quá trình Request – Response của Http



# ĐỐI TƯỢNG REQUEST

- Nhận thông tin từ trình duyệt, thông tin này bao gồm các thông số của Form được gửi lên thông qua phương thức POST (mặc định)/GET hay các thông số gửi kèm trang ASP.NET được gọi.
- Đối tượng Request còn dùng để trao đổi các thông tin giữa các trang ASP.NET trong cùng website thông qua các giá trị cookies trên máy client.
- Bảng thuộc tính và phương thức của Request



# ĐỐI TƯỢNG REQUEST

- Lấy dữ liệu từ người dùng gửi lên:
  - ❖ Request.QueryString["Tên\_Phần\_tử\_cần\_đọc"]:  
Đề đọc giá trị của một phần tử được gửi theo phương thức Get (Method = "Get").
  - ❖ Phương thức Request.Form ["Tên\_Phần\_tử\_cần\_đọc"] : Đề đọc giá trị của một phần tử được gửi theo phương thức Post (Method = "Post").



# ĐỐI TƯỢNG REQUEST

- Các hàm thường sử dụng
  - Request.Form["param-name"] => String
  - Request.QueryString["param-name"] => String
  - Request.Cookies["cookie-name"] => HttpCookie
  - Request.Params["mix-name"] => String
  - Request.Browser
  - Request.UserHostAddress



# ĐỐI TƯỢNG REQUEST

- Ví dụ:

- ❖ Đọc tham số
- ❖ Kiểm soát trình duyệt
- ❖ IP của client
- ❖ Đọc Cookie
- ❖ Đăng nhập có Cookie



# ĐỐI TƯỢNG RESPONSE

- Đối tượng Response được sử dụng để giao tiếp với client, nó quản lý và điều phối thông tin từ Web Server đến các trình duyệt của người dùng.
- Sử dụng để chuyển dữ liệu về client. Dữ liệu gồm nội dung trang web, Cookie và thông tin điều khiển
- Các hàm thường dùng:
  - ❖ `Response.Write("HTML")`
  - ❖ `Response.Redirect("url");`
  - ❖ `Response.Cookies.Add(HttpCookie)`





# ĐỐI TƯỢNG RESPONSE

- Trong đó, hai hàm chính thường dùng: **Write** và **Redirect**.

❖ Ví dụ:

Chuyển tới trang sanpham.aspx trong cùng web site:

```
Response.Redirect("sanpham.aspx");
```

❖ Chuyển tới trang [www.amazon.com](http://www.amazon.com):

```
Response.Redirect("http:// www.amazon.com ");
```



# ĐỐI TƯỢNG RESPONSE

Ví dụ minh họa Request, Response

- ❖ Bước 1 : Tạo 2 trang ASP.NET gồm trang NhậpTen.aspx và trang XemChiTiet.aspx. Bảng mô tả các thuộc tính của các controls trang NhậpTen.aspx.

Control	Tên thuộc tính	Giá trị thuộc tính
<b>Label</b>	Text	Nhập tên
<b>Label</b>	Text	Ngày sinh
<b>TextBox</b>	ID	txtTen
<b>TextBox</b>	ID	txtNgaySinh
<b>Button</b>	Text	Xem chi tiết
	ID	btnXemChiTiet



# ĐỐI TƯỢNG RESPONSE

minh họa Nhập tên

localhost:52666/Chương\_0 x

localhost:52666/Chương\_08/NhapTen.aspx

**MINH HỌA REQUEST - RESPONSE**

Nhập tên

Ngày sinh

Kết quả trả về

localhost:52666/Chương\_0 x

localhost:52666/Chương\_08/XemChiTiet.aspx?Ten=admin&NgaySinh=08/06/2010

Xin chào bạn : **admin**

Ngày sinh của bạn là : **08/06/2010**



# ĐỐI TƯỢNG RESPONSE

❖ Bước 2 : Viết code xử lý cho các trang.

code behide trang nhapten.aspx.cs

```
protected void btnXemChiTiet_Click(object sender, EventArgs e)
{
    //Lấy các giá trị đã nhập vào các TextBox

    string strHoTen = txtTen.Text;

    string strNgaySinh = txtNgaySinh.Text;

    //Chuyển quan trang XemChiTiet.aspx

    Response.Redirect("XemChiTiet.aspx?Ten=" + strHoTen +
"&NgaySinh=" + strNgaySinh);
}
```



# ĐỐI TƯỢNG RESPONSE

## code behide trang Xemchitiet.aspx.cs

```
public partial class XemChiTiet : System.Web.UI.Page

{

    protected void Page_Load(object sender, EventArgs e)

    {

        //Lấy các giá trị từ trang NhapTen.aspx

        string strHoTen = Request.QueryString["Ten"];

        string strNgaySinh = Request.QueryString["NgaySinh"];

        string s = "Xin chào bạn : <b>" + strHoTen + "</b><br>" +

            "Ngày sinh của bạn là : <b>" + strNgaySinh + "</b>";

        //In xuống trình duyệt

        Response.Write(s);

    }

}
```



Sinh viên tự thiết kế một webform lấy dữ liệu từ webform khác => đối chiếu kết quả



# ĐỐI TƯỢNG SERVER

- Đối tượng Server được sử dụng để cung cấp thông tin của Server cho ứng dụng.
- Chứa các hàm tiện ích hỗ trợ các thao tác lập trình web như
  - ❖ Ánh xạ đường dẫn từ ảo sang vật lý
  - ❖ Thực thi một trang ASP.NET khác
  - ❖ Chuyển sang trang ASP.NET khác
  - ❖ Mã hóa và giải mã chuỗi HTML và URL



# ĐỐI TƯỢNG SERVER

- Danh sách Thuộc tính/Phương thức:

Tên thuộc tính	Diễn giải
MachineName	Lấy tên của Web Server
ScriptTimeout	Thiết lập thời gian xử lý tối đa 1 file





# ĐỐI TƯỢNG SERVER

Phương thức	Diễn giải
Excute	Thực thi một trang ASP.NET khác
HtmlDecode	Giải mã chuỗi thành các thẻ HTML tương ứng (ngược với HtmlEncode)
HtmlEncode	Mã hóa HTML thành chuỗi
MapPath	Lấy đường dẫn vật lý hoặc đường dẫn ảo đến một thư mục trên Server
Transfer	Ngừng thi hành trang hiện hành, gửi yêu cầu mới đến trang được gọi thực hiện
UrlDecode	Ngược lại của UrlEncode, giải mã chuỗi mã hóa URL
UrlEncode	Mã hóa URL thành chuỗi (khoảng trắng thành dấu "+", ký tự không thuộc chữ và số sẽ chuyển thành số hexadecimal).



# ĐỐI TƯỢNG SERVER

Phân biệt Response.Redirect với Server.Transfer:

- Response.Redirect: Chuyển đến 1 trang mới giống như gõ địa chỉ trang đó trên trình duyệt và request đến server.
  - Server.Transfer: Cũng chuyển đến trang mới nhưng ở phía Server, làm giảm request đến server, giữ nguyên URL và có thể chuyển cả các query string và biến n trên Form đến địa chỉ mới, chỉ thực hiện giữa các trang trên cùng 1 host.
- \* Ví dụ đối tượng Server: Minh họa Upload file lên server.



# ĐỐI TƯỢNG SESSION

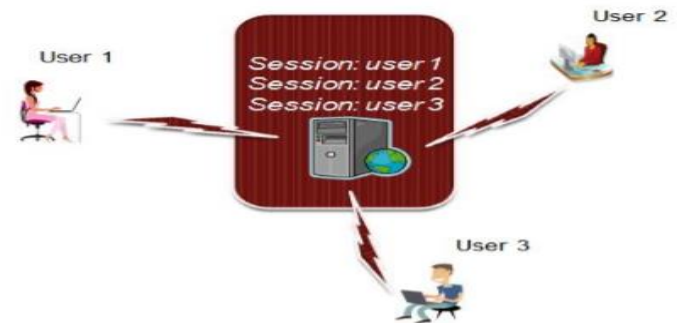
## 1. Khái niệm:

- Là đối tượng của lớp **HttpSessionState** thuộc không gian tên **System.Web.SessionState**.
- Lấy nội dung trang và thực hiện một số tương tác với ứng dụng rồi đóng kết nối
- Dùng để lưu trữ thông tin về phiên làm việc của người dùng.
- Session hết hạn khi gọi Website khác hay đóng cơ sở trình duyệt, hay ta có thể quy định thời gian hết hạn của session



# ĐỐI TƯỢNG SESSION

- Lưu giữ thông tin của riêng người sử dụng trong suốt phiên làm việc.
- Các thao tác thường dùng
  - Cất giữ /cập nhật/ truy xuất biến session
  - Xóa một biến session
  - Xóa sạch các biến session
  - Cập nhật thời gian chờ của một phiên



Hình 9: Session trong ASP.NET



# ĐỐI TƯỢNG SESSION

Các hàm thường dùng

- **Session.Add("name")** tạo biến session hiện tại
- **Session["name"] [= value]=>object**
- **Session.Clear()** xóa sách các biến session
- **Session.Abandon()** hủy session hiện tại
- **Session.Remove("name")** xóa biến session
- **Session.Timeout = minutes; //** thời gian chờ tối đa



## 2. Cơ chế làm việc của Session:

- User yêu cầu hiển thị 1 trang web
- ASP.NET sẽ dò từng liên kết trong mã HTML, cuối mỗi liên kết, ASP.NET sẽ gắn thêm Session ID đã được mã hoá, như  
`<a href="sessionwithoutcookies.aspx?SessionID=...">`
- Khi user kích vào một liên kết, ASP.NET sẽ giải mã và chuyển Session đó đến trang web user yêu cầu



## 3. Kiểm soát đối tượng Session:

- Session hết hạn khi gọi Website khác hay đóng cửa sổ trình duyệt, hay ta có thể quy định thời gian hết hạn của session

**Session.Timeout = 500**

- hay có thể kết thúc session ngay tức khắc bằng cách hủy bỏ session:

**Session.Abandon**



# ĐỐI TƯỢNG SESSION

## 4. Lưu và nhận giá trị vào Session:

- Lưu trữ giá trị value vào session với tên khóa name hay tại vị trí index:

*Session[string name] = value;*

*Session[int index] = value;*

Có thể sử dụng đối tượng Session.Contents thay cho đối tượng Session

*Session.Contents[string name] = value*

*Session.Contents[int index] = value*

Ví dụ: **Session ["username"] = "hoa"**





# ĐỐI TƯỢNG SESSION

- Nhận giá trị từ session với tên khóa name:

*object Session[string name]*

Ví dụ: `Response.Write(Session["username"])`

- Xoá giá trị từ session với tên khóa name hay tại vị trí index

*void Session.Remove(string name)*

*void Session.RemoveAt(int index)*



# ĐỐI TƯỢNG SESSION

- Đếm số giá trị từ session:

*int Session.Count*

Ví dụ: Có thể duyệt tất cả các giá trị từ session

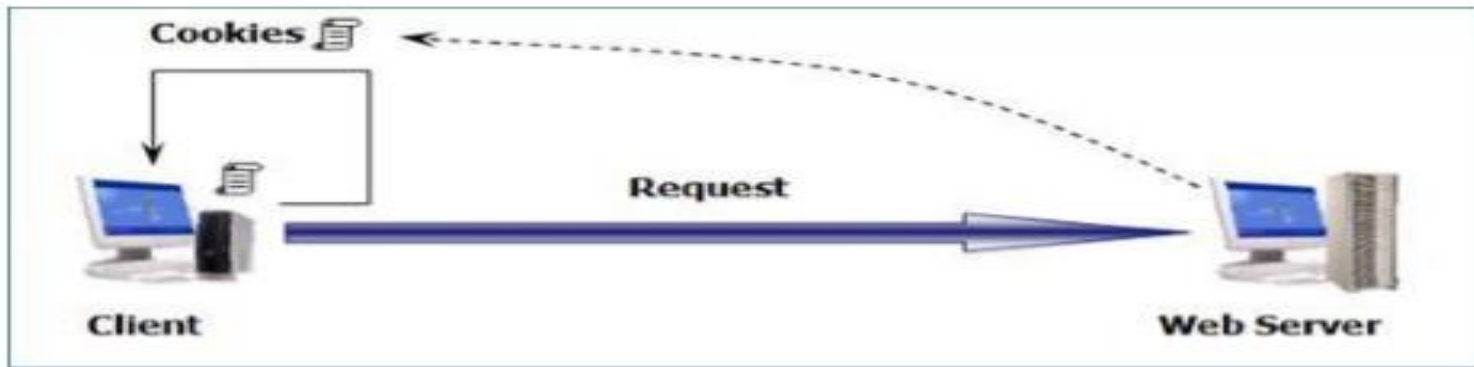
```
for (int i = 0; i < Session.Count; i++)  
Response.Write(Session[i].ToString() +  
"<br>");
```



# ĐỐI TƯỢNG COOKIE

## 1. Khái niệm:

Session là cơ chế lưu thông tin một phiên kết nối phía trình chủ. Còn cookies lưu thông tin một phiên kết nối trên tập tin ở máy khách. Cookies được lưu trên máy khách bởi trình duyệt, và được gửi lên máy chủ khi yêu cầu hiển thị trang web. Thường cookies dùng để lưu các thông tin liên quan đến user như tên người sử dụng đã ghé thăm web site. mật khẩu





# ĐỐI TƯỢNG COOKIE

## 2. Lưu và nhận đối tượng từ Cookie:

- Tạo Cookie: có 2 phương pháp

- \* Tạo nhiều Cookies, mỗi Cookie kèm với một giá trị value (Cookie 1 giá trị)

*Response.Cookies[string name].Value = value*

*Response.Cookies[int index].Value = value*

- \* Tạo một Cookie với nhiều cặp key/value (Cookie nhiều giá trị)

*Response.Cookies[string name][string key] = value*



# ĐỐI TƯỢNG COOKIE

Có thể tạo đối tượng Cookie, rồi sử dụng hàm Add để bổ sung Cookie vào tập hợp Cookie

*HttpCookie c = new HttpCookie(string name)*

*HttpCookie c = new HttpCookie(string name, string value)*

*Response.Cookies.Add(HttpCookie cookie)*

- Cập nhật giá trị của đối tượng Cookie trong tập hợp Cookie

*Response.Cookies.Set(HttpCookie cookie)*

- Truy xuất giá trị cookies do trình khách gửi lên

*Request.Cookies[string name].Value*

*Request.Cookies[string name][string key]*



# ĐỐI TƯỢNG COOKIE

- Xóa đối tượng Cookie với tên khóa name

*Response.Cookies.Remove(string name)*

- Xóa tất cả các đối tượng Cookie

*Response.Cookies.Clear()*

- Tổng của các cặp khóa/ giá trị Cookie trong tập hợp Cookie nhận được bởi biến Count:

*Response.Cookies.Count*



# ĐỐI TƯỢNG COOKIE

## 3. Kiểm soát giá trị Cookie:

- Thiết lập thời gian hết hạn của giá trị Cookie

*Response.Cookies["user"].Expires = DateTime.Parse("02/20/2008")*

Thiết lập thời gian hết hạn của giá trị Cookie là sau một tháng kể từ lúc thực hiện:

*Response.Cookies["user"].Expires = DateTime.Now.AddMonths(1)*

- Để xóa tất cả giá trị Cookie ở máy client, ta đơn giản thiết lập giá trị của Expires thành 0 hay giá trị thời gian thuộc về quá khứ, Cookie sẽ biến mất khi user đóng trình duyệt



Link tham khảo: phân biệt session và cookie trong asp  
<https://www.youtube.com/watch?v=VrrtAvLK-b4>





# ĐỐI TƯỢNG APPLICATION

- Được sử dụng để lưu thông tin chia sẻ cho tất cả mọi người truy cập website.
- Chú ý sự khác biệt so với Session là phạm vi chia sẻ của Application là toàn cục, trong khi đó Session chỉ dành riêng cho người tạo ra nó



# ĐỐI TƯỢNG APPLICATION

- Các hàm thường dùng
  - ❖ **Application.Add("name")** thêm một biến Application
  - ❖ **Application["name"][=value;]** đọc/cập nhật biến
  - ❖ **Application.Clear()** xóa sạch các biến
  - ❖ **Application.Remove("name")** xóa biến
  - ❖ **Application.Lock()** khóa Application để tránh xung đột
  - ❖ **Application.Unlock()** mở khóa Application



# ĐỐI TƯỢNG GLOBAL.ASPX

- Chứa các sự kiện xảy ra liên quan đến vòng đời của ứng dụng và người truy xuất. Sau đây là mô tả các sự kiện xảy ra ngay
  - ❖ **Application\_Start()** sau khi ứng dụng khởi động
  - ❖ **Application\_End()** trước khi ứng dụng kết thúc
  - ❖ **Application\_Error()** sau khi ứng dụng gặp lỗi
  - ❖ **Session\_Start()** sau khi người dùng truy cập vào site
  - ❖ **Session\_End()** trước khi session bị time out



# BÀI TẬP

1. Kể vài đặc tính và ứng dụng dùng đối tượng Cookie?
2. So sánh đối tượng Cookie và Session
3. So sánh đối tượng Session và Application
4. Phát triển trang web sử dụng đối tượng Application để hiển thị số người dùng truy cập Website