

## CHƯƠNG 2: CÁC ĐỐI TƯỢNG TRONG ASP.NET

Trong bất kỳ ứng dụng nào, dù là winform based hay webform based thì việc giao tiếp (tương tác) với người dùng và giữa các webform với nhau là điều bắt buộc. Ví dụ ta cần phải lấy thông tin đặt hàng do người dùng nhập vào và hiển thị trở lại người dùng một số thông tin hữu ích khác, như kết quả thanh toán... hay một trang chuyển tiếp kết quả cho một trang khác để xử lý v.v...

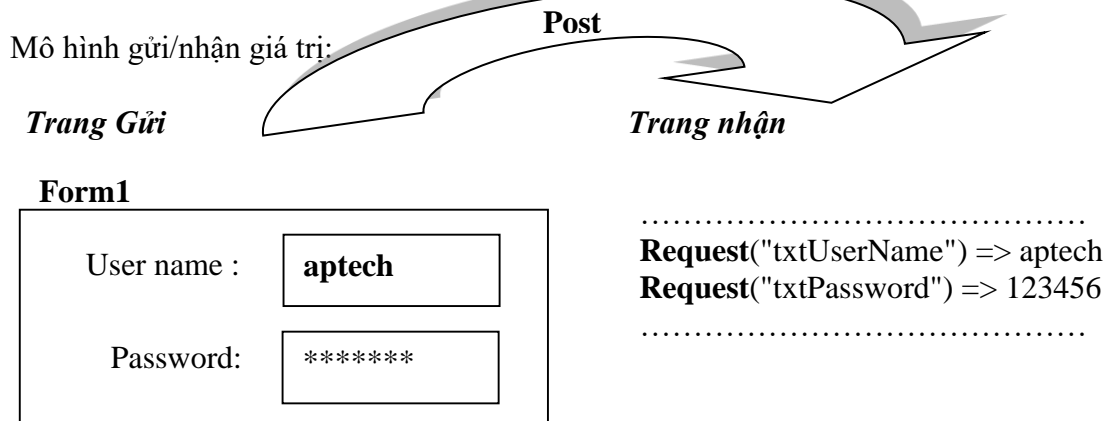
Ở bài trước, để làm điều này chúng ta thực hiện dễ dàng thông qua các server controls như textbox, listbox, dropdownlist, label,... Tuy nhiên những điều khiển này chỉ có tác dụng trong một Page còn các trang khác thì hoàn toàn không thể đọc/ghi giá trị nằm trong các điều khiển này.

Để thực hiện việc giao tiếp (truyền dữ liệu) giữa các webform ASP.NET cung cấp một tập các điều khiển giúp ta làm việc đó một cách dễ dàng, đó là: Đối tượng Request và đối tượng Response. Trong bài học này, chúng ta cũng tìm hiểu thêm một số đối tượng khác cũng rất hay dùng khi xây dựng ứng dụng là đối tượng Server, Application và Session.

### 2.1. Request Object

#### 2.1.1 Đối tượng Request dùng để làm gì ?

Request là một đối tượng của ASP.NET, dùng để nhận các thông tin từ Client gửi về cho Web server.



#### 2.1.2 Các thành phần (thuộc tính và phương thức) chính

- Phương thức **Request.QueryString.Get("Tên\_Phần tử cần đọc")**: Để đọc giá trị của một phần tử được gửi theo phương thức Get (Method = "Get")
- Phương thức **Request.Form.Get("Tên\_Phần tử cần đọc")**: Để đọc giá trị của một phần tử được gửi theo phương thức Post (Method = "Post").
- Thuộc tính **QueryString**: cho phép nhận các giá trị truyền qua chuỗi tham số  
**Cú pháp**: `Biến=Request.QueryString["Tên_tham_số"];`

#### Ví dụ 1:

Xây dựng 2 trang web : trang **Webform1.aspx**, trong đó có 2 textbox chứa tên và mật khẩu. Khi người dùng click vào nút **submit** thì gửi tên và mật khẩu sang trang **Webform2.aspx** để hiển thị.

User name:

Password:

Trang nguồn (gửi): Webform1.aspx

Username: quynm  
Password: 123456

Kết quả nhận về: trang Webform2.aspx

Code của 2 trang sẽ như sau:

### Webform1.aspx

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Trang gửi</title>
</head>
<body>
  <form id="form1" runat="server" method="post" >
  <div>
    User name:
    <asp:TextBox runat="server" ID="txtUserName"></asp:TextBox><br />
    Password:
    <asp:TextBox runat="server" ID="txtPassword"></asp:TextBox><br />
    <asp:Button runat="server" ID="cmdSubmit" Text="Submit"
   PostBackUrl="~/Bai1/WebForm2.aspx" />
  </div>
</form>
</body>
</html>
```

### WebForm2.aspx

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Trang nhận</title>
</head>
<body>
  <form id="form1" runat="server">
    <asp:Label ID="lblKetQua" runat="server"></asp:Label>
  </form>
</body>
</html>
```

### WebForm2.aspx.cs

```
protected void Page_Load(object sender, EventArgs e)
{
  lblKetQua.Text = "Username: " + Request.Form.Get("txtUserName")
+ "<br>Password: " + Request.Form.Get("txtPassword");
}
```

## Ví dụ 2: (tự làm)

Xây dựng 2 trang web : trang **TienDien.aspx**, trong đó có 2 textbox chứa chỉ số điện cũ và mới. Khi người dùng click vào nút **Tính** thì gửi chỉ số điện cũ và mới sang trang **XL\_TienDien.aspx** để hiển thị số tiền phải trả. Biết rằng:

- 100 kw đầu giá là: 2000đ/kw
- 50 kw tiếp giá là: 2500đ/kw
- 50 kw tiếp giá là: 2800đ/kw
- >200 kw giá là: 3500đ/kw

Vào chỉ số điện cũ:	<input type="text" value="100"/>
Vào chỉ số điện mới:	<input type="text" value="300"/>
<input type="button" value="Tính tiền"/>	

\***XL\_TienDien.aspx.cs**

## 2.2. Response Object

### 2.2.1 Đối tượng Response dùng để làm gì ?

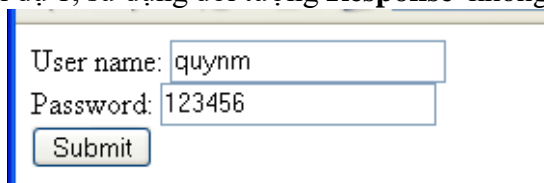
Đối tượng này được dùng để gửi nội dung (một chuỗi) bất kỳ về cho trình duyệt. Quản lý và điều phối thông tin từ Web Server đến trình duyệt của người dùng.

### 2.2.2 Các thành phần (thuộc tính và phương thức) chính

- Phương thức: `Response.write(<Biểu thức>)` dùng để gửi giá trị biểu thức truyền vào cho phía trình duyệt.
- Phương thức `Response.Redirect("địa chỉ URL")`: Chuyển tới một trang khác.

## Ví dụ1:

Tạo một trang Login như ví dụ 1, sử dụng đối tượng **Response** không dùng Method = "Post/Get".



User name:   
Password:

Code của 2 trang sẽ như sau:

### **Webform1.aspx**

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Trang gửi</title>
</head>
<body>
  <form id="form1" runat="server" >
    <div>
      User name:
      <asp:TextBox runat="server" ID="txtUserName" runat="server"></asp:TextBox><br />
      Password:
      <asp:TextBox runat="server" ID="txtPassword"></asp:TextBox><br />
```

```
<asp:Button runat="server" ID="cmdSubmit" Text="Submit" />
```

```
</div>
</form>
</body>
</html>

Webform1.aspx.cs
protected void cmdSubmit_Click(object sender, EventArgs e)
{
    string sName = txtUserName.Text;
    string sPass = txtPassword.Text;
    Response.Redirect("WebForm2.aspx?Name=" + sName + "&Pass=" + sPass);
}

WebForm2.aspx
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Trang nhận</title>
</head>
<body>
    <form id="form1" runat="server">
        <asp:Label ID="lblKetQua" runat="server"></asp:Label>
    </form>
</body>
</html>

WebForm2.aspx.cs
protected void Page_Load(object sender, EventArgs e)
{
    lblKetQua.Text = "Username: " + Request.QueryString["Name"].ToString() +
        " <br> Password: " + Request.QueryString["Pass"].ToString();
}
```

## Ví dụ 2: (tự làm theo 2 cách: Request và Response)

Xây dựng 2 trang web : trang **TienLuong.aspx**.

Mã NV:	NV04	
Bậc lương:	4	
Ngày công:	27	
Chức vụ:	Trưởng phòng ▼	
		<input type="button" value="TÍNH"/> <input type="button" value="XÓA"/>

Khi người dùng click vào nút **Tính** thì gửi kết quả sang trang **XL\_TienLuong.aspx**

Mã NV:	NV04	
Có ngày công:	27	
Tiền được lĩnh là:	79500000	

TienLinh= BacLuong \* 650000 \* NCTL + PhuCap

Với: + NCTL = NgayCong nếu NgayCong < 25

= (NgayCong - 25) \* 2 + 25 nếu NgayCong >= 25

- + PhuCap=500000 nếu chức vụ là Trưởng phòng
- =300000 nếu chức vụ là Phó phòng
- =100000 nếu chức vụ là Nhân viên

## 2.3. Server Object

- Dùng để cung cấp thông tin của Web Server cho ứng dụng
- Phương thức:
  - Transfer(“URL”): ngừng thi hành trang hiện hành, gửi yêu cầu mới tới trang khác
  - MapPath(): trả về đường dẫn vật lý tương ứng với đường dẫn ảo trên Web Server

### Ví dụ

- In ra tên của máy chủ hiện hành: **Response.Write(Server.MachineName);**
- Cho biết đường dẫn thực sự trên ổ cứng (thư mục vật lý) của trang hiện hành (trang default.aspx) : **Server.MapPath(“default.aspx”);**
- Cho biết đường dẫn vật lý ứng với tệp QLCB.Mdb, biết rằng tệp này nằm trong một thư mục con là “App\_Data”: **Server.MapPath(“App\_Data/QLDB.MDB”);**

## 2.4. Session Object

- Dùng để lưu trữ thông tin trong một phiên làm việc cụ thể.
- Được tạo ra khi người dùng kết nối tới Web Server lần đầu tiên.

Nói cách khác, đối tượng (biến) **session** là một biến mà mọi trang trong một phiên (Session) đều có thể truy xuất

+ Cú pháp để **tạo biến** Session:

***Session[“Tên\_Biến”]=Giá\_trị;***

*Ví dụ :* Tạo một biến tên là MaNguoiDung và gán giá trị là TK34

***Session[“MaNguoiDung”]=“TK34”;***

+ Cú pháp để **lấy giá trị** từ biến session:

***Biến=Session[“Tên\_Biến”];***

Riêng với đối tượng **Session**, nó còn có các sự kiện. Các sự kiện này tự động được gọi mỗi khi một phiên làm việc được tạo ra. Các sự kiện này có tên là **\_Start** và **\_End**. Các sự kiện này được đặt trong file **Global.asax**.

**Lưu ý :** Tập tin **Global.asax** được dùng để:

- + Khai báo và khởi tạo giá trị cho các biến Application, Session.
- + Viết xử lý cho các sự kiện của 2 đối tượng Application và Session.

## 2.5. Application Object

- Đối tượng toàn cục, quản lý toàn bộ ứng dụng Web
- Thông tin được lưu trong đối tượng (biến) Application được “**hiểu**” ở tất cả các trang aspx trong suốt thời gian “**sống**” của ứng dụng.
- Chỉ bị đóng/hủy khi tắt Web Server.
- Tạo biến Application:

***Application[“tên\_biến”]=giá\_trị;***

- Lấy giá trị:

***<biến>=Application[“tên\_biến”];***

+ Ví dụ: Tạo biến So\_Nguoi\_Truy\_Cap

***Application[“So\_Nguoi\_Truy\_Cap”]= 0;***

Ngoài ra, đối tượng Application còn có 2 phương thức thường dùng là **Application.Lock()**: Để khóa không cho người khác sửa đổi các biến toàn cục và **Application.UnLock()** để mở khóa .

Đối tượng Application cũng có 2 sự kiện đó là **Application\_Start** và **Application\_END**. Sự kiện **Start** chỉ được kích hoạt duy nhất một lần khi yêu cầu đầu tiên phát sinh. Sự kiện **END** được kích hoạt khi dịch vụ web dừng (unload).

Đối tượng Application có 2 phương thức là Lock và Unlock. Khi gọi phương thức Lock (khóa) thì tất cả các ứng dụng không được phép thay đổi các giá trị Application. Để các ứng dụng khác được phép thay đổi các biến Application thì gọi phương thức Unlock.

Mã lệnh viết cho 2 sự kiện này cũng được đặt trong file **Global.asax**.

**Ví dụ 1:** Tạo một trang **Login**, nếu người dùng nhập user name và mật khẩu tương ứng là **son** và **123** thì được phép truy cập các trang **Home.aspx**, trái lại mỗi lần người dùng truy cập đến trang **Home.aspx** thì đều được chuyển tới trang **Login.aspx**.

**Cần tạo 3 trang là Home.aspx/cs, Login.aspx/cs và Global.asax** như sau:

**Global.asax**

```
...
Void Session_Start(object sender, EventArgs e)
    { Session["TrangThai"]=""; }
...
```

**Home.aspx**

...

```
<body>
<form id="form1" runat="server">
  <h1><asp:Label ID="lblThongbao" runat="server"> </asp:Label> </h1>
</form>
</body>
</html>
```

**Home.aspx.cs**

```
protected void Page_Load(object sender, EventArgs e)
{
    if (Session["TrangThai"] == "roi")
        lblThongbao.Text = "Chào mừng bạn đến Website của chúng tôi!";
    else
    {
        lblThongbao.Text = "Bạn chưa đăng nhập!";
        Response.Write("<a href='Login.aspx'> Login </a>");
    }
}
```

**Login.aspx**

...

```
<body>
<form id="form1" runat="server">
<table align="center">
  <tr>
    <td>User name (son): </td>
    <td>
      <asp:TextBox runat="server" ID="txtUserID" >
      </asp:TextBox>
    </td>
  </tr>
  <tr>
    <td>Password (123): </td>
    <td><asp:TextBox runat="server" ID="txtPassword"
TextMode="Password"></asp:TextBox>
    </td>
  </tr>
</table>
```

```

<td colspan="2" align="center"> <asp:Button runat="server" Text="Login"
ID="cmdLogin" onclick="cmdLogin_Click"/> </td>
</tr>
</table>
</form>
</body>
</html>

```

### Login.aspx.cs

```

protected void cmdLogin_Click(object sender, EventArgs e)
{
    if (txtUserID.Text == "son" && txtPassword.Text == "123")
    {
        Session["TrangThai"] = "roi";
        Response.Redirect("Home.aspx");
    }
    else
    {
        Session["TrangThai"] = "chua";
        Response.Write("Bạn đăng nhập sai!");
    }
}

```

**Ví dụ 2:** Tạo một trang đếm số lượng người truy cập. Dùng biến tệp text để lưu.

- Tạo file **SL.txt** nhập giá trị 0
- Tạo 2 trang là Index.aspx/cs và Global.asax với nội dung sau:

#### Trang Index.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Index.aspx.cs"
Inherits="Index" %>

<!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>Home Page - Hit counter</title>
</head>
<body>
    <form id="form1" runat="server">
        <h1>Chào mừng bạn đã đến website của chúng tôi</h1>
        <asp:Label runat="server" ID="lblSLKhach"></asp:Label>
    </form>
</body>
</html>

```

#### Trang Index.aspx.cs

```

using System;
public partial class Index : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        lblSLKhach.Text="Bạn là vị khách thứ: " +
            Application["SLTruyCap"].ToString();
    }
}

```



```
<%@ Application Language="C#" %>
```

```
<script runat="server">
    void Application_Start(object sender, EventArgs e)
    { //Đọc file SL.txt lấy giá trị gán cho biến S
        Application.Lock ();
        System.IO.StreamReader sr;
        sr = new System.IO.StreamReader (Server.MapPath ("SL.txt"));
        string S = sr.ReadLine ();
        sr.Close ();
        Application.UnLock ();
        //Tạo một biến Application là SLTruyCap và khởi tạo giá trị lấy từ biến S
        Application["SLTruyCap"]=S;
    }
    ...
    void Session_Start(object sender, EventArgs e)
    {
        //Tăng số lượng người truy cập lên 1 khi có một người mới thăm
        Application["SLTruyCap"] =
            int.Parse (Application["SLTruyCap"].ToString ()) + 1;
        //Lưu vào file SL.txt (mở và ghi đè)
        System.IO.StreamWriter sw;
        sw = new System.IO.StreamWriter (Server.MapPath ("SL.txt"));
        sw.Write (Application["SLTruyCap"].ToString ());
        sw.Close ();
    }
</script>
```

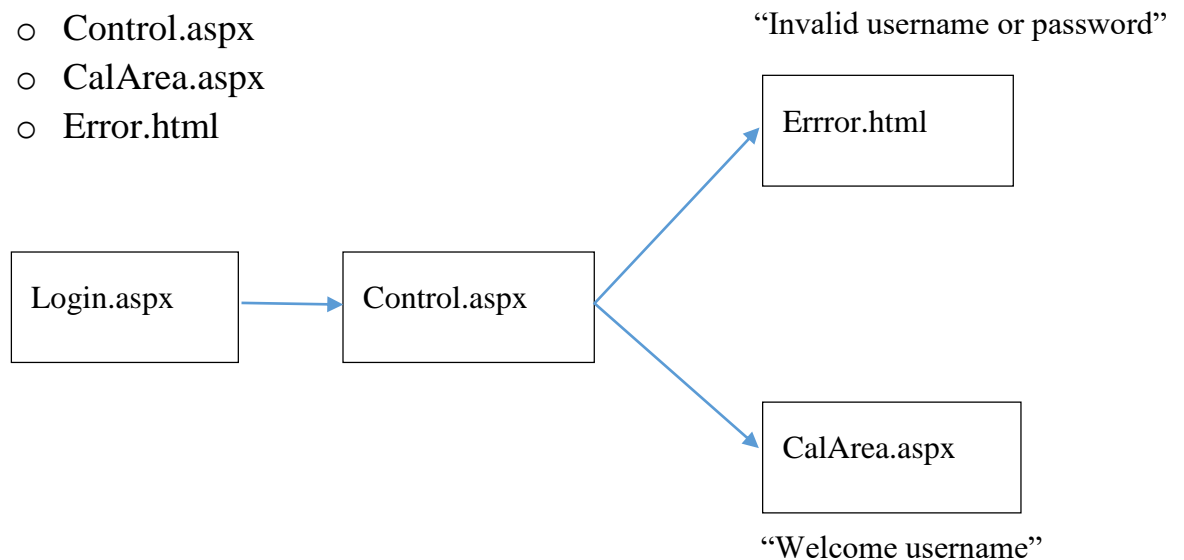
Sau khi tạo, chạy file **Index.aspx** để kiểm chứng sẽ thấy rằng số lượng người truy cập luôn luôn tăng lên bất kể là server có tắt hay máy tính bị trục trặc.



## BÀI TẬP

tạo 4 trang web theo yêu cầu dưới đây:

- Login.aspx
- Control.aspx
- CalArea.aspx
- Error.html



1. Trang Login.aspx có các điều khiển label, textbox, button để đăng nhập vào hệ thống sau đó chuyển hướng tới trang Control.aspx.
2. Trang Control.aspx sẽ kiểm tra xem username và password có bằng nhau không, nếu có thì chuyển tới trang CalArea.aspx.
3. Nếu username và password là khác nhau sẽ chuyển sang trang Error.html và hiển thị thông báo “Invalid username or password.
4. Trang CalArea sẽ thực hiện hiển thị lời chào “Welcome” + username nhập vào từ trang login. Trang CalArea.aspx cũng cho phép tính toán diện tích của hình chữ nhật khi người dùng nhập vào chiều dài và chiều rộng của nó.
5. Nếu người dùng chạy trực tiếp trang CalArea.aspx thì sẽ có thông báo lỗi “Bạn chưa đăng nhập”.