



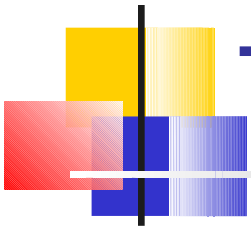
Chương 5

Application, Server và Session



Mục tiêu

- *Tập tin Global.asax*
- *Các sự kiện trong Global.asax*
- *Sử dụng đối tượng Application*
- *Sử dụng đối tượng Server*
- *Sử dụng đối tượng Session*

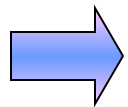


Tập Global.asax

G
-
b
o
-
a
-
s
a
s
a
x

Lưu trữ trong thư mục gốc của ứng dụng

Định nghĩa phạm vi của ứng dụng



Khởi tạo các biến mức application hay session

Nối kết cơ sở dữ liệu

Gửi cookie



Các sự kiện trong Global.asax

Sự kiện	Mô tả
Application_Start	Phát sinh lúc trang asp.net trong ứng dụng được gọi
Application_End	Phát sinh lúc session cuối cùng của ứng dụng kết thúc hay ứng dụng dừng sử dụng Internet Services Manger snap-in.
Application_BeginRequest	Phát sinh lúc mỗi lần trang bắt đầu yêu cầu(trang được nạp hay refresh).



Events in Global.asax

Contd...

Sự kiện	Mô tả
Application_EndRequest	Phát sinh mỗi lần một yêu cầu của trang kết thúc
Session_Start	Phát sinh mỗi lần session bắt đầu.
Session_End	Phát sinh mỗi lần session kết thúc.



Global.asax Ví dụ

Global.asax

```
<script language="C#" runat="server">
    protected void Application_Start(Object sender,
    EventArgs e)
    {
    }

    protected void Session_Start(Object sender,
    EventArgs e)
    {
        Response.Write( "Session Started <br>");
    }
</script>
```



Global.asax Example

```
protected void Application_BeginRequest(Object
sender, EventArgs e)
{
    Response.Write("<h1>Application
Begins</h1>");
    Response.Write ("Application request begins
<br>");
}
protected void Application_EndRequest(Object
sender, EventArgs e)
{
    Response.Write ("Application request ends
<br>");
}
```



Global.asax Ví dụ

```
protected void Session_End(Object sender, EventArgs e)
{
    Response.Write("Session ended");
}
protected void Application_End (Object sender,
EventArgs e)
{
}
</script>
```

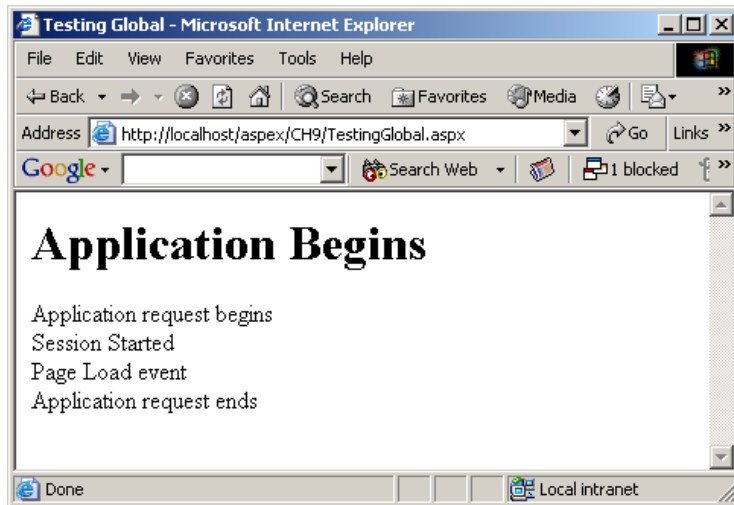



Test Global.aspx

```
<html>
<title>Testing Global</title>
  <script language="C#" runat="server">
    void Page_Load(Object sender, EventArgs e)
    {
      Response.Write ("Page Load event <br>");
    }
  </script>
</html>
```

Global.asax Kết xuất

Refresh





Đối tượng Application

Biểu diễn một thể hiện của một ứng dụng ASP.NET.



Syntax

```
Object [varName]
```

Application Level
Session Level



```
Application ["greeting"] = "Welcome to our sites";
```



Đối tượng Application Ví dụ

```
void Application_Start(Object sender, EventArgs E)
{
    Application ["sessioncount"] = 0;
}
```

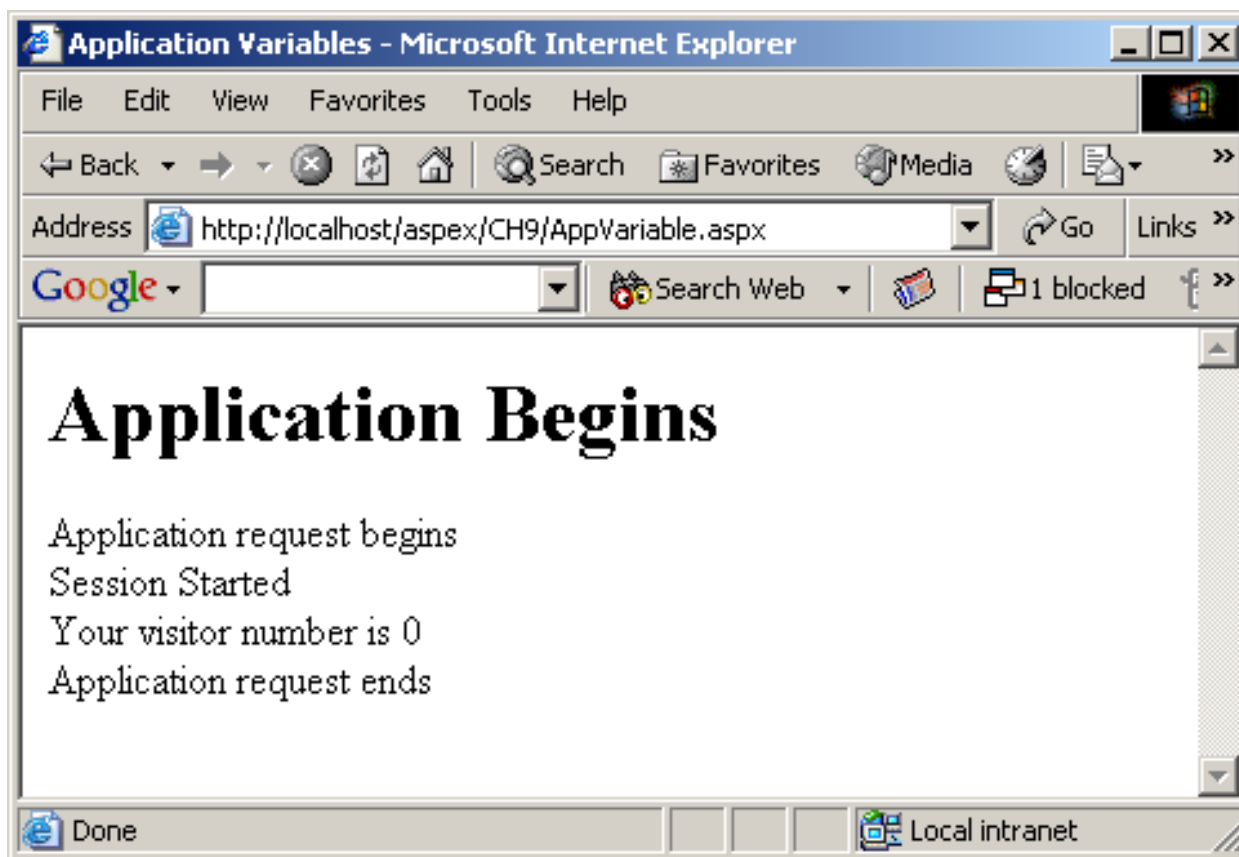
Global.asax

```
<HTML>
<script Language = "C#" runat = "server" Debug = "true">
void Page_Load(Object Src, EventArgs E)
{
    Response.Write ("Your visitor number is " +
    Application ["sessioncount"]);
}
</script>
<form runat= "server" >
</form>
</HTML>
```

ApplicationVariable.aspx



Đối tượng Application Ví dụ





Đối tượng Application Ví dụ

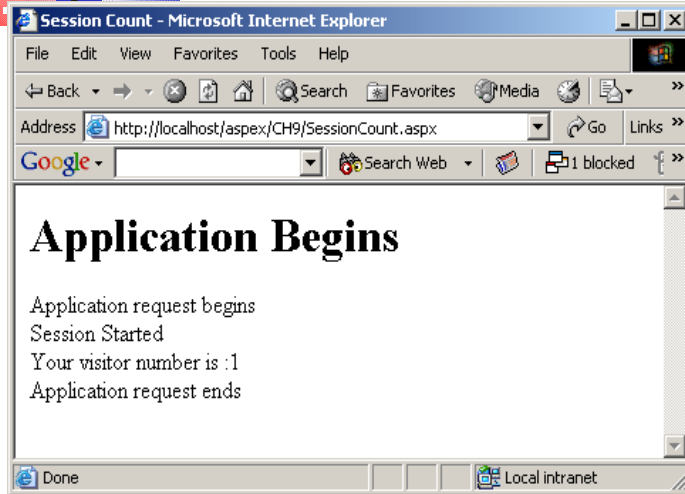
```
void Session_Start(Object sender, EventArgs e)
{
    Application["sessioncount"]=(Int32)Application["sessioncount"] + 1;
}
```

Global.asax

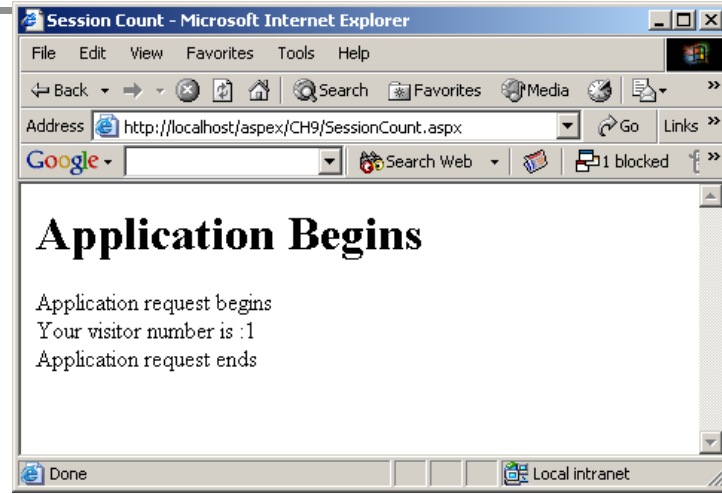
```
<html>
  <script Language = "C#" runat = "server" Debug = "true">
    void Page_Load(Object Src, EventArgs E)
    {
      Response.Write ("Your visitor number is :" + Application["sessioncount"]);
    }
  </script>
  <form runat= "server" > </form>
</html>
```

SessionCount.aspx

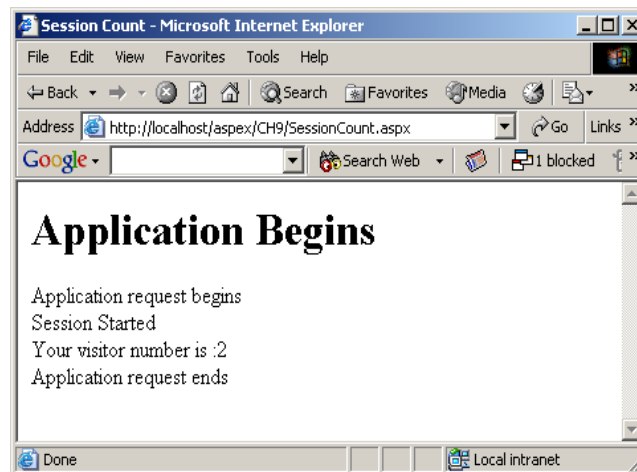
Đối tượng Application kết xuất



Output Initially



Output after refreshing



Output after reopening the browser



Kiểm soát truy xuất

- Biến ứng dụng có thể được truy xuất và cập nhật bởi tất cả các trang trong ứng dụng ASP.NET
- Bởi vì biến chỉ duy trì một giá trị, nên nếu người dùng thay đổi một giá trị của biến nó sẽ thay đổi đến tất cả người dùng trong ứng dụng
- Để đảm bảo các biến mức ứng dụng không được cập nhật một cách đồng thời, đối tượng Application sử dụng phương thức Lock() và Unlock().



CoKiểm soát truy xuất

```
Application.Lock();  
//...code to change the value of the application  
variables  
.....  
.....  
Application.Unlock();
```

- Phương thức Lock khóa tất cả các biến trong khối mã
- Đảm bảo chỉ một người dùng hiện tại có điều khiển qua trang.
- Khi phương thức Unlock được gọi, người dùng hiện tại mất kiểm soát



Mảng

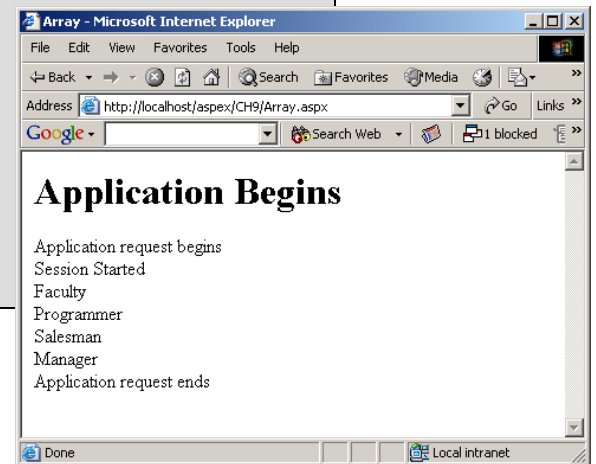
- Mảng mức ứng dụng có thể chia xẻ nhóm các thông tin chung qua toàn bộ ứng dụng
- Thông thường các mảng mức ứng dụng sử dụng thông tin tĩnh.
- Khi sử dụng mảng trong đối tượng ứng dụng các phần tử trong mảng không nên thay đổi trực tiếp

```
String [] job = new String [4];  
    job[0]= "Faculty";  
    job[1]= "Programmer";  
    job[2]= "Salesman";  
    job[3]= "Manager";  
Application ["j"] = job;
```



Array.aspx

```
<HTML>
  <script Language = "C#" runat = "server" >
    void Page_Load(Object Src, EventArgs E)
    { int i = 0; String[] k;
      k = (String[])Application["j"];
      for (i = 0; i < k.Length; i++)
      {
        Response.Write(k[i] + "<br>");
      }
    }
  </script>
</HTML>
```



Đối tượng Server

Cho phép web server có thể được kiểm soát và hoạt động như là một giao tiếp với HTTP service

Syntax \rightarrow **Server.property | method**

Properties

Thuộc tính	Mô tả
ScriptTimeout	Chỉ ra khoảng thời gian một script được chạy trên server trước khi nó kết thúc
MachineName	Dùng để lấy tên máy của server

Methods



- Execute and Transfer
- HTML Encode
- URL Encode
- Map Path

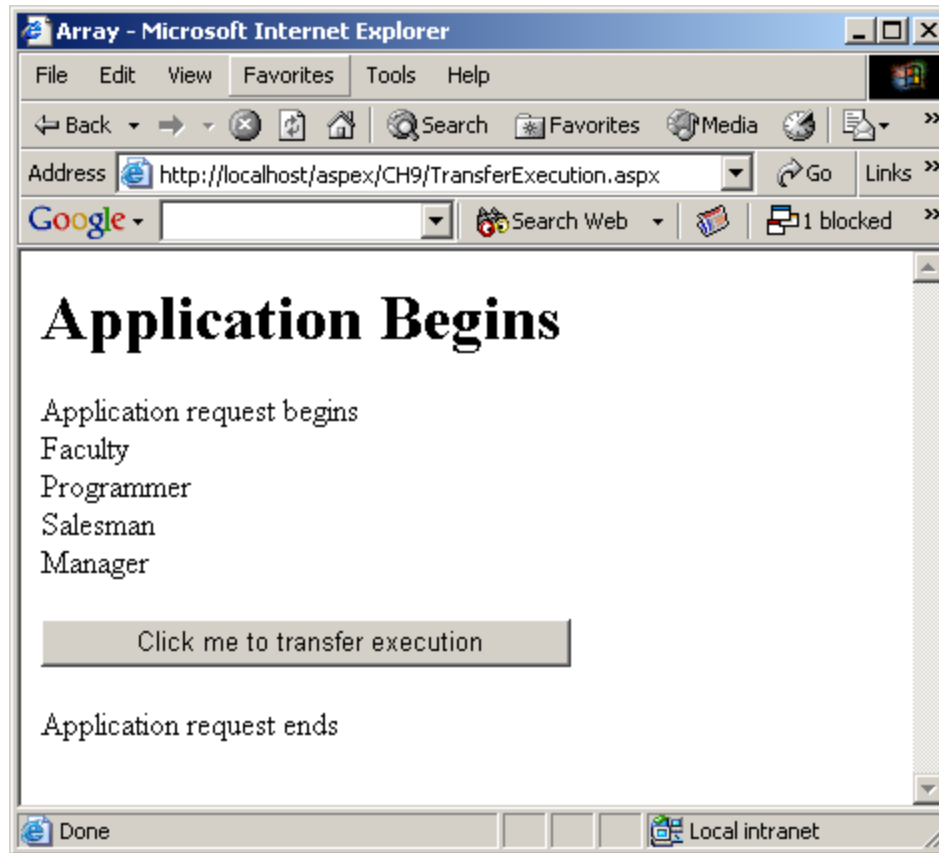


Phương thức Execute

Phương thức Execute() dùng để truyền thực thi từ trang hiện tại đến một trang khác, và trả lại thực thi đến trang hiện tại

```
<%@ Page Debug = "true"%><html>
  <script language="C#" runat="server">
    void clicked (Object Src, EventArgs E)
    {Server.Execute ("Array.aspx");}
  </script>
  <form runat = "server">
    <asp:button id = "btnClick" onclick = "clicked"
Text = " Click me to transfer execution" runat =
"server" />
  </form>
</html>
```

Phương thức Execute kết xuất

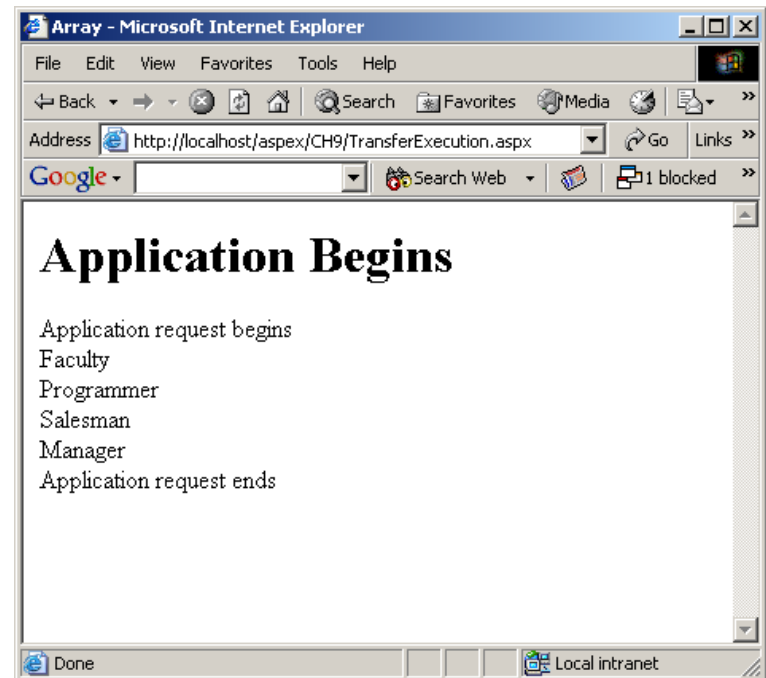


Phương thức Transfer

Phương thức này dùng để truyền toàn bộ thực thi đến một trang nào đó
Khác với phương thức Execute(), kiểm soát bị mất từ trang gọi phương thức thực thi

Syntax:

Server.Transfer ("Array.aspx");





Phương thức HTML Encode

- `HtmlEncode()` dùng để mã hóa một chuỗi HTML nào đó.
- Trong trường hợp không gọi phương thức `HtmlEncode()` `method is not specified`, `<H1>` được trình hiểu là một thẻ HTML, và thẻ được hiển thị theo.
- Cú pháp phương thức `HtmlEncode()` :

`Server.HTMLEncode (string)`



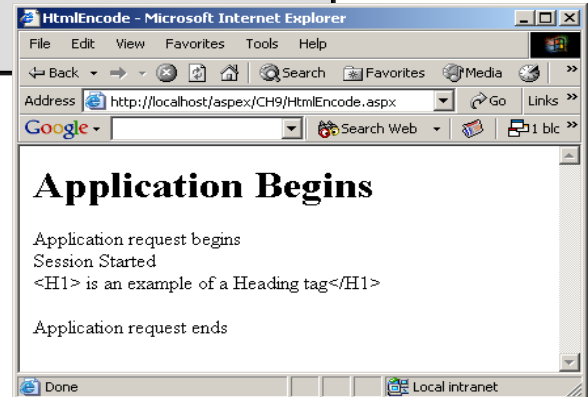
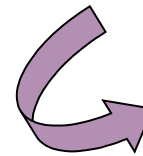
Ví dụ HTML Encode

```
<%@ Page Debug = "true"%>
<html>
  <title>HtmlEncode</title>
  <script language="C#" runat="server">
void Page_Load (Object Src, EventArgs E)
{
    Response.Write (Server.HtmlEncode("<H1> is an
example of a Heading tag</H1>"));
}
  </script>
  <form runat = "server"></form>
</html>
```

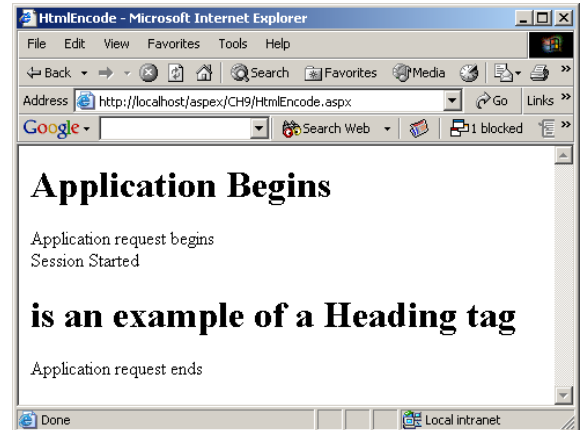
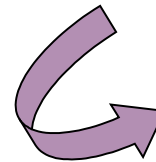


HTMLEncode kết xuất

```
Response.Write( Server.HtmlEncode("<H1> is an  
example of a Heading tag</H1>"));
```



```
Response.Write("<br><H1> is an  
example of a Heading tag</H1>");
```





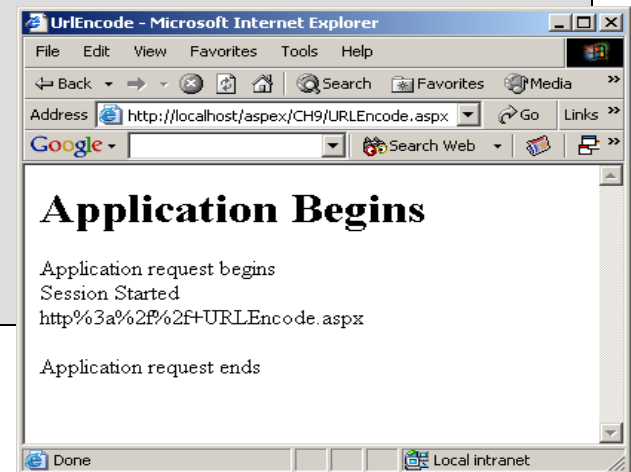
Phương thức URLEncode

- Một URL có thể dùng để gửi dữ liệu từ client đến một tập tin trên server
- Nếu dữ liệu không phải kí tự alphanumeric nó cần được mã hóa trước khi gửi đến server.
- Cú pháp:

```
Server.URLEncode(string);
```

URLEncode Ví dụ

```
<%@ Page Debug = "true"%>
<html>
  <title>Ur1Encode</title>
  <script language="C#" runat="server">
void Page_Load(Object Src, EventArgs e)
{
    Response.Write(Server.UrlEncode ("http://
URLEncode.aspx" ));
}
</script>
<form runat = "server"></form>
</html>
```





Phương thức URLEncode

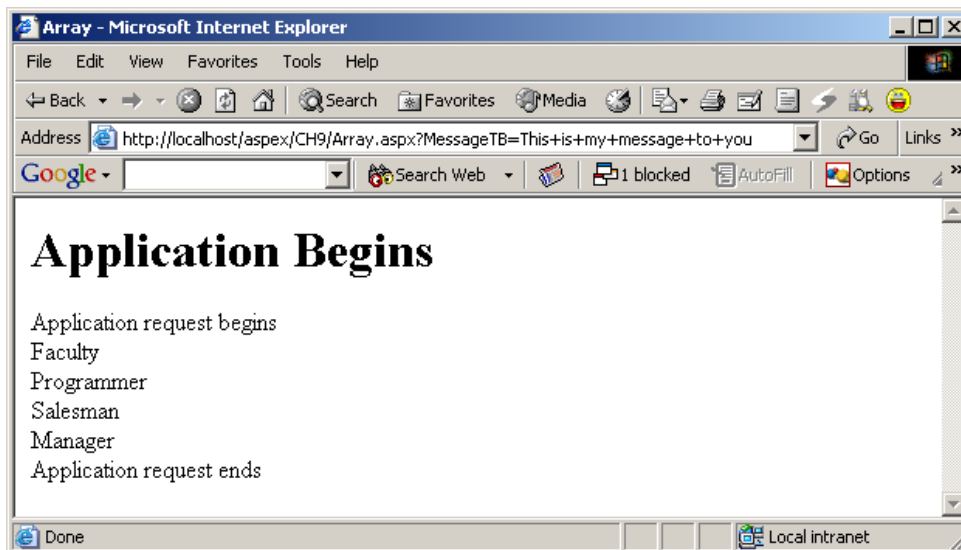
```
<html>
  <script language="C#" runat="server">
    void Meclicked(Object Src, EventArgs e)
    {
        string MyMessage = Server.UrlEncode ("This is my
message to you");
        Response.Redirect ("Array.aspx?MessageTB=" +
MyMessage);
    }
  </script>
  <form runat = "server">
    <asp:button id = "btnClick" onclick = "Meclicked"
Text = "Click me to send messages" runat = "server"/>
  </form>
</html>
```

Phương thức URLEncode

Khi mã được thực thi, trang 'Array.aspx' được gọi từ Web Server để xử lý thông tin được gửi trong URL.

Tập tin 'Array.aspx' có thể lấy dữ liệu trong trường MyMessage.

Kết xuất





Phương thức MapPath

- Dùng bởi server để ánh xạ đường dẫn đến thông tin trên server
- Phương thức này là một giao tiếp giữa thư mục ảo và thư mục vật lý trên Web Server
- Cú pháp:

Server.MapPath (path)



MapPath Ví dụ 1

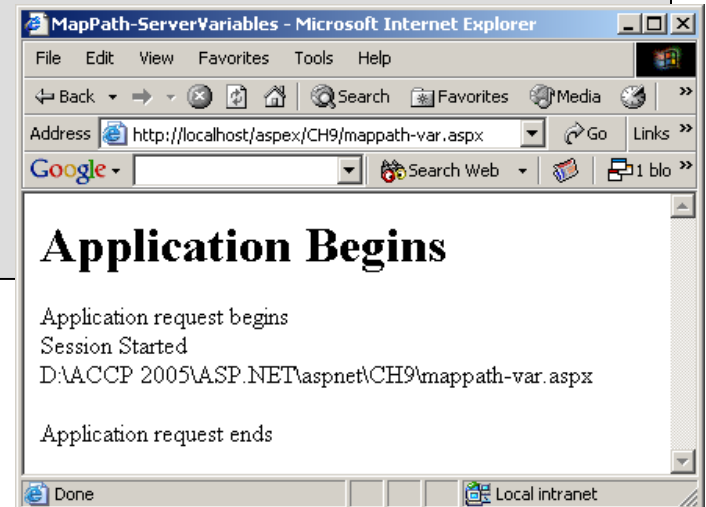
```
<%@ Page Debug ="true"%>
<html>
  <script language="C#" runat="server">
    void Page_Load (Object Src, EventArgs E)
    {
      Response.Write (Server.MapPath("/Array.aspx"));
    }
  </script>
  <form runat ="server">
  </form>
</html>
```


MapPath Kết xuất



MapPath Ví dụ 2

```
<%@ Page Debug = "true"%>
<html>
  <title>MapPath-ServerVariables </title>
  <script language="C#" runat="server">
    void Page_Load(Object Src, EventArgs e)
    {
      Response.Write (Server.MapPath (Request.ServerVariables.Get
("PATH_INFO")));
    }
  </script>
  <form runat = "server"> </form>
</html>
```

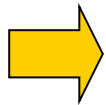




Đối tượng Session

- Dùng để lưu trữ thông tin về người dùng
- Thông tin này được duy trì trong suốt phiên làm việc của người dùng
- Biến lưu trữ trong đối tượng Session không bị loại bỏ khi trình duyệt người dùng chuyển từ trang này đến trang khác bên trong cùng một ứng dụng
- Đối tượng bị hủy khi người dùng kết thúc session hay hết hạn thời gian tồn tại cho một session

THỰC HIỆN



■ **Chứa thông tin người dùng đặc biệt**

■ **Theo dõi và giám sát thông tin người dùng trong một session**

■ **Hủy đối tượng sau khi session hết hạn**



Sử dụng Variable

- Một biến session khác biến ứng dụng
- Biến session chỉ dành cho một người dùng nào đó bên trong một session
- Biến ứng dụng có thể truy xuất hay thay đổi bởi các người dùng khác trong ứng dụng
- Biến session chỉ được truy xuất qua phiên làm việc của người dùng đó.



Sử dụng Variable – ví dụ 1

```
<html>
<title>Session Variable</title>
  <script language="C#" runat="server">
    void Page_Load(Object Sender, EventArgs e)
    {
        Response.Write("<u><b><center>Logon
authentication using Session
variables</center></b></u><br><br>");
    }
    void WritingSesVar(Object Sender, EventArgs e)
    {
        Session["Name"] = txtUserName.Text;
        Session["Password"] = txtPassword.Value;
        lblMessage1.Text = "Session variables
stored";
    }
  </script>
</html>
```



Sử dụng Variable – ví dụ 1

```
        lblMessage2.Text = "";
    }

    void ReadingSesVar(Object Sender, EventArgs e)
    {
        lblMessage1.Text = "The value of name is " + Session
["Name"];
        lblMessage2.Text = "The value of password is " +
Session ["Password"];
    }
    void Clear(Object Sender, EventArgs e)
    {
        lblMessage1.Text = "";
        lblMessage2.Text = "";
    }
}
```

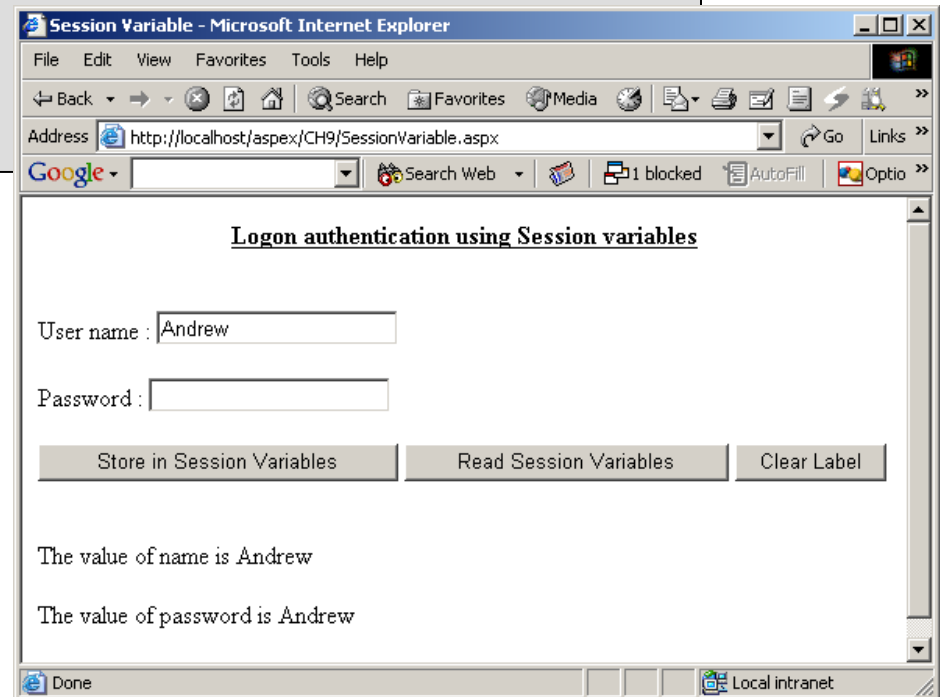


Sử dụng Variable – ví dụ 1

```
</script>
<form runat="server">
  User name :
  <asp:TextBox id = "txtUserName" runat="server" />
  <br><br>
  Password :
  <input id="txtPassword" type="password" runat="server">
  <br><br>
  <asp:button id="btnStoreVar" text="Store in Session
Variables" onclick="WritingSesVar" runat="server" />
  <asp:button id="btnReadVar" text="Read Session
Variables" onclick="ReadingSesVar" runat="server" />
  <asp:button id="btnClearLabel" text="Clear Label"
onclick="Clear" runat="server"/>
```

Sử dụng Variable – Kết xuất

```
<br><br>
  <asp:Label id = "lblMessage1" runat="server" />
  <br><br>
  <asp:Label id = "lblMessage2" runat="server" />
  <br><br>
</form>
</html>
```



Session Variable - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print Mail News RSS Feeds

Address http://localhost/aspex/CH9/SessionVariable.aspx Go Links

Google Search Web 1 blocked AutoFill Optio

Logon authentication using Session variables

User name :

Password :

Store in Session Variables Read Session Variables Clear Label

The value of name is Andrew

The value of password is Andrew

Done Local intranet



Thay đổi biến

Giá trị trong một biến session cũng có thể được đọc từ các trang web khác

SessionVariable1.aspx

```
<html>
  <script language="C#" runat="server">
    void Page_Load(Object Sender, EventArgs e)
    {
      Response.Write("<u><b><center>Logon authentication
using Session variables</center></b></u><br><br>");
    }
  </script>
</html>
```



SessionVariable1.aspx ...

```
void WritingSesVar(Object Sender, EventArgs e)
{
    Session["Name"] = txtUserName.Text;
    Session["Password"] = txtPassword.Value;
    lblMessage.Text = "Session variables stored";
}
void ReadingSesVar(Object Sender, EventArgs E)
{
    Response.Redirect("SessionVariableRedirected.aspx");
}
</script>
<form runat="server">
    User name :
    <asp:TextBox id = "txtUserName" runat="server" /><br><br>
    Password :
    <input id="txtPassword" type="password" runat="server"><br><br>
```

SessionVariableRedirected.as



px

```
<asp:button id="btnStoreVar" text="Store in Session
Variables" onclick="WritingSesVar" runat="server" />
  <asp:button id="btnReadVar" text="Read Session
Variables" onclick="ReadingSesVar" runat="server" />
  <br><br>
  <asp:Label id = "lblMessage" runat="server"
/><br><br>
</form>
</html>
```

SessionVariableRedirected.aspx

```
<html>
  <title>Redirected Page</title>
  <script language="C#" runat="server">
```

SessionVariableRedirected.asp

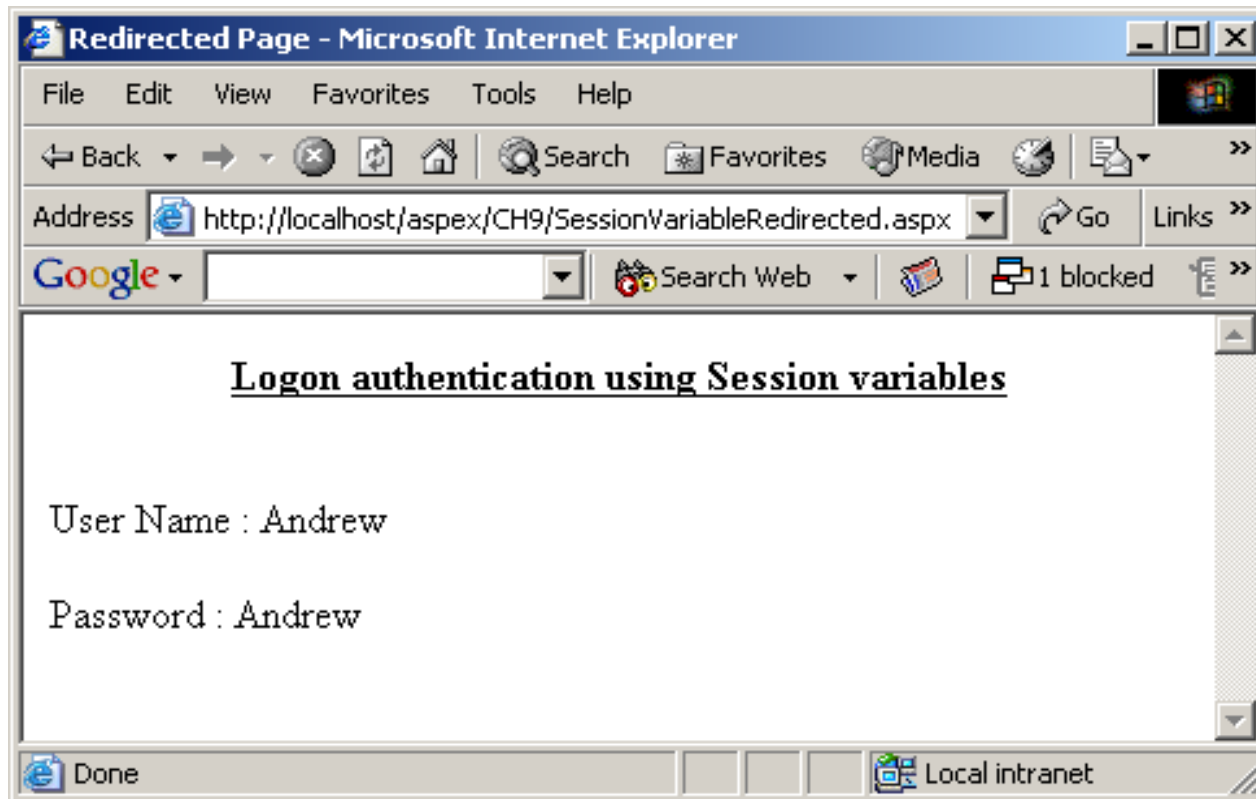


X

```
void Page_Load(Object Sender, EventArgs E)
{
    Response.Write("<u><b><center>Logon authentication
using Session variables</center></b></u><br><br>");
    lblMessage1.Text = "User Name : " +
(String)Session ["Name"];
    lblMessage2.Text = "Password : " +
(String)Session["Password"];
}
</script>
<form runat="server">
<asp:Label id = "lblMessage1" runat="server" /> <br><br>
<asp:Label id = "lblMessage2" runat="server" /> <br><br>
</form>
</html>
```

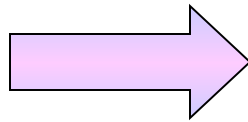
SessionVariableRedirected.asp

x kết xuất



Sự kiện và thuộc tính của Session

Events



Session_Start
Session_End

Session.collection | property | method

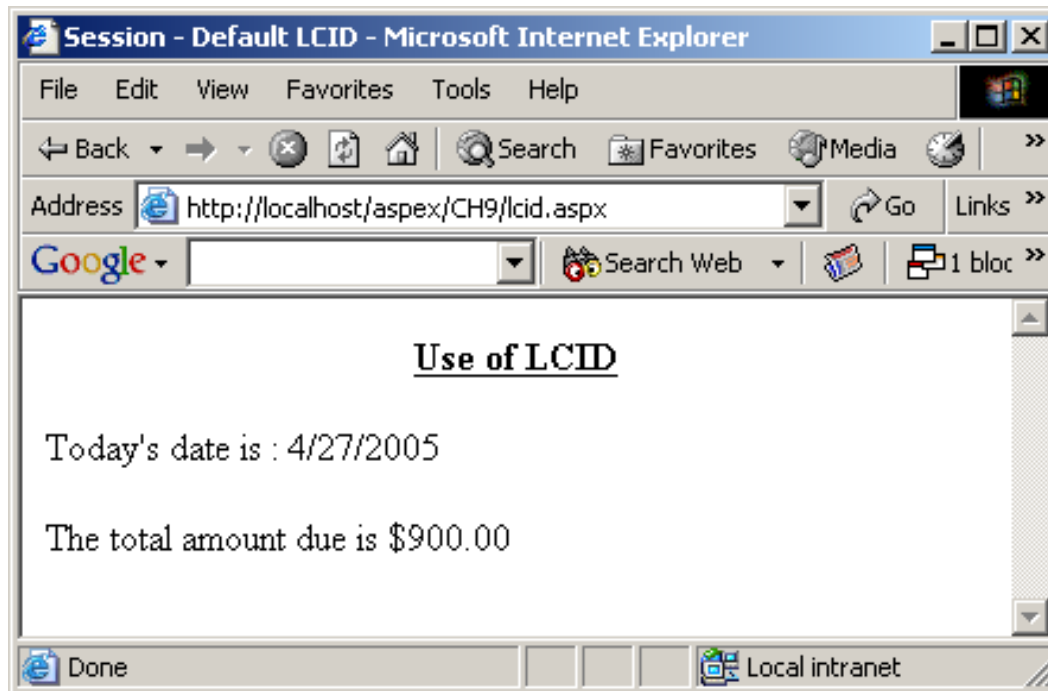
Thuộc tính	Mô tả
SessionID	Xác định tính duy nhất của người dùng
TimeOut	User timeout
LCID	Local identifier
IsNewSession	Trả về true nếu session được tạo trong yêu cầu hiện tại
Item	Tên của giá trị Session
Count	Số các mục trong tập hợp trạng thái session



Session ví dụ

```
<HTML>
<script language="C#" runat="server">
void Page_Load (Object Src, EventArgs E)
{
Response.Write("<b><u><center>Use of
LCID</b></u></center><br>");
DateTime dt;
dt = DateTime.Now;
int due = 500;
Response.Write ("Today's date is : " +
dt.ToShortDateString() + "<br><br>");
Response.Write ("The total amount due is " + due.ToString
("C") ); }
</script>
</HTML>
```

Session kết xuất





Session ví dụ

```
<HTML>
<title>Session - LCID French</title>
  <script language="C#" runat="server">
    void Page_Load(Object Src, EventArgs e)
    {
      Response.Write("<b><u><center>Change of LCID </b>
</u> </center> <br>");
      Session.LCID =0x040C;
      DateTime dt;
      dt = DateTime.Now;
      int due = 500;
      Response.Write ("Today's date is : " +
dt.ToShortDateString() + "<br>");
    }
  </script>
</HTML>
```

Session kết xuất

```
Response.Write ("The total amount due is : " +  
due.ToString ("C")) ;  
}  
</script>  
</HTML>
```

