

Departament de Llenguatges i Sistemes Informàtics

Tema 13. Aplicacions Web (IV)

Herramientas Avanzadas para el Desarrollo de Aplicaciones

Escola Politècnica Superior
Universitat d'Alacant

Indice

1. Cookies
2. Enviament d'email
3. Fitxers
4. Controls d'usuari

1

Cookies

Cookies

- Per emmagatzemar dades relatives a un usuari es pot utilitzar l'objecte Session
- Problema:
 - Les dades s'esborren quan l'usuari tanca la finestra del navegador
- Solució:
 - Per emmagatzemar dades i que aquests es preservin és necessari utilitzar les cookies

Cookies

- Les cookies són extractes de dades que una aplicació ASP.NET pot emmagatzemar en el navegador del client per a la seva posterior recuperació
- Les cookies no es perden quan es tanca el navegador (tret que l'usuari les esborri)

Cookies en ASP.NET

- Una cookie es representa per la classe `HttpCookie`
- Les cookies de l'usuari es llegeixen a través de la propietat `Cookies` de l'objecte `Request`
- Les cookies de l'usuari es modifiquen a través de la propietat `Cookies` de l'objecte `Response`
- Per defecte les cookies expiren quan es tanca el navegador
 - Es poden alterar els punts d'expiració (posar una data determinada d'expiració)

Cookies en ASP.NET

Página default.aspx.cs

```
protected void Page_Load(object sender, EventArgs e)
{
    HttpCookie userCookie;
    userCookie = Request.Cookies["UserID"];
    if (userCookie == null)
    {
        Label1.Text = "No existe la cookie, creando cookie ahora";
        userCookie = new HttpCookie("UserID", "Ana López");
        userCookie.Expires = DateTime.Now.AddMonths(1);
        Response.Cookies.Add(userCookie);
    }
    else
    {
        Label1.Text = "Bienvenida otra vez, " + userCookie.Value;
    }
}
```

Cookies en ASP.NET

- La variable `userCookie` s'inicialitza com una instància de la classe `HttpCookie` i se li assigna el valor de la cookie cridada `UserID`
- Es comprova l'existència de la cookie
 - Cas de no existir es mostra un missatge
 - Se li dóna el valor “Ana López”
 - Se li assigna una data d'expiració
- La cookie es transfereix al navegador usant el mètode `Response.Cookies.Add`
- Si la cookie existeix es mostra un missatge de benvinguda

Cookies en ASP.NET

- La primera vegada que es carrega la pàgina es mostrarà el missatge
 - No existeix la cookie, creant cookie ara
- Si es recarrega de nou la pàgina, la cookie ja existirà i es mostrarà el missatge
 - Bienvenida una altra vegada, Ana López
- Cuidat!
 - Tingues en compte que els usuaris poden rebutjar les cookies
 - No pots deixar que recaiguin en elles aspectes importants de l'aplicació

Cookies en ASP.NET

- Esborrar cookies
 - L'única forma és reemplaçar-la per una cookie amb una data d'expiració que ja ha passat

```
HttpCookie userCookie= new HttpCookie("UserID");  
userCookie.Expires=DateTime.Now.AddDays(-1);  
Response.Cookies.Add(userCookie);
```

2

Email

Email

- Suposem que tenim una tenda online i volem enviar un email de confirmació de comanda a cada client
- En lloc d'escriure manualment cada email ASP.NET permet automatitzar aquest procés

Email

- System.Net.Mail
 - **SmtpClient**
 - **MailMessage**
 - **Attachment**
 - **AttachmentCollection**
 - **MailAddress**
 - **MailAddressCollection**

Email

- **MailMessage**
 - From
 - To
 - CC
 - Bcc
 - Attachments
 - Subject
 - Body
 - IsBodyHTML

Email

```
SsmtpClient smtpClient = new SsmtpClient("smtp.gmail.com",587);
MailMessage message = new MailMessage();
try
{
    MailAddress fromAddress = new MailAddress("irene@dlsi.ua.es", "Alias remitente");
    MailAddress toAddress = new MailAddress("correo@gmail.com", "Alias destinatario");
    message.Attachments.Add(new Attachment("C:\\imagen1.gif"));
    message.Attachments.Add(new Attachment("C:\\imagen2.jpg"));
    message.From = fromAddress;
    message.To.Add(toAddress);
    message.Subject = "Probando el envío!";
    message.Body = "Este es el cuerpo del mensaje";
    smtpClient.EnableSsl = true;

    smtpClient.Credentials = new System.Net.NetworkCredential("usuario", "password");
    smtpClient.Send(message);
    Label1.Text = "Mensaje enviado.";
}
catch (Exception ex)
{
    Label1.Text = "No se pudo enviar el mensaje!";
}
```

3

Fitxers

Fitxers

- En ASP.NET existeixen un conjunt de classes per treballar amb fitxers
 - System.IO
- Ens permeten llegir, escriure i actualitzar el contingut de fitxers i directoris
- Tres tipus de classes para:
 - Treballar amb fitxers
 - Treballar amb streams
 - Treballar amb directoris

Fitxers

- Classes per treballar amb fitxers i streams:
 - **File**: Mètodes per treballar amb fitxers
 - **FileStream**: Representa un stream per llegir i escriure en fitxers
 - **StreamReader**: Llegeix caràcters d'un fitxer de text
 - **StreamWriter**: Escriu caràcters en un fitxer de text
 - **Path**: Mètodes per manipular un fitxer o directori

Fitxers

- Escrivint en un arxiu:

Pàgina WriteFile.aspx

```
<asp:TextBox ID="TextBox1" runat="server" TextMode="MultiLine" Rows="10"
    Width="220px"></asp:TextBox>
<asp:Button ID="Button1" runat="server" Text="Escribir"
    onclick="Button1_Click" />
```

Pàgina Writefile.aspx.cs

```
protected void Button1_Click(object sender, EventArgs e)
{
    using (StreamWriter streamwriter = File.CreateText(@"C:\Dpaa\Mitexto.txt"))
    {
        streamwriter.WriteLine(TextBox1.Text);
    }
}
```

Fitxers

- El mètode CreateText sobreescriu l'arxiu cada vegada que s'invoca
- Alternativa:
 - AppendText: Agrega texto a continuació
- La directiva using permet alliberar els recursos utilitzats en acabar l'escriptura

Fitxers

- Llegint d'un arxiu

```
string inputString;  
TextBox1.Text = "";  
using (StreamReader streamReader =  
    File.OpenText(@"C:\Dpaa\Mitexto.txt"))  
{  
    inputString = streamReader.ReadLine();  
    while (inputString != null)  
    {  
        TextBox1.Text += inputString + "\n";  
        inputString = streamReader.ReadLine();  
    }  
}
```

Fitxers

- Pujant arxius
 - Els usuaris poden pujar arxius al servidor
 - Fotos
 - Presentacions
 - Vídeos
- En ASP.NET
 - Control FileUpload
 - Proporciona un TextBox i un botó per seleccionar els arxius de l'ordinador de l'usuari



Fitxers

- Propietats de FileUpload:
 - **HasFile:** True si l'usuari ha pujat un arxiu
 - **FileName:** El nom del fitxer com string
 - **FileContent:** Stream que pot ser utilitzat per llegir el contingut
 - **FileBytes:** Array de bytes que pot ser utilitzat per llegir el contingut
 - **PostedFile:** Un objeto HttpPostedFile
 - **ContentLength:** La grandària del fitxer en bytes
 - **ContentType:** El tipo MIME (image/gif)
 - **SaveAs:** Se li passa un string que conté la ruta i el nom de l'arxiu a guardar

Fitxers

Arxiu UploadFile.aspx

```
<asp:FileUpload ID="FileUpload1" runat="server" />
<asp:Button ID="Button2" runat="server" onclick="Button2_Click" Text="Upload" />
<br />
<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
```

Arxiu Uploadfile.aspx.cs

entArgs e)

```
{ if (FileUpload1.HasFile)
{
    string fileName = FileUpload1.FileName;
    FileUpload1.SaveAs(@"C:\Dpaa\"+fileName);
    Label1.Text = "Fichero " + fileName + " adjuntado con éxito.";
}
else
    Label1.Text = "No existe fichero adjunto!";
}
```


4

Controls d'usuari.

Què són?

- Es poden crear controls personalitzats i reutilitzables en diferents pàgines : CONTROLS D'USUARI
- Són arxius amb extensió .ascx que inclourem a les nostres pàgines web aspx. Els canvis realitzats en el ascx seran reflectits allà on l'hàgim situat.
- Contenen un o diversos controls ASP.NET juntament amb el codi necessari perquè els controls realitzin la funcionalitat desitjada.

Diferències amb webforms

- L'extensió de nom d'arxiu per al control d'usuari és .ascx.
- En lloc d'una directiva **@Page** el control d'usuari conté una directiva **@Control** que defineix la configuració i altres propietats.
- Els controls d'usuari no es poden executar com a arxius independents. En el seu lloc, ha d'agregar-los a les pàgines ASP.NET, com faria amb qualsevol altre control.
- El control d'usuari no conté elements **html**, **bodi** o **form**. Aquests elements han d'estar a la pàgina d'allotjament.

Creació

- Agregar un nou element al projecte, de tipus user control
- Una vegada creat, treballarem com si d'una pàgina .aspx normal es tractés
- Per afegir el control a qualsevol lloc de la nostra Web, no tenim més que arrossegar-ho al lloc que vulguem..

Incloure un control d'usuari en una pàgina Web ASP.NET

- Es pot fer de dues maneres:
 1. Manualment afegim la següent directiva en el aspx
 - **Atribut src:** defineix la ruta al control d'usuari
 - **Atribut tagprefix:** permet associar un prefix al control d'usuari
 - **Atribut Tagname:** associem un nom al control d'usuari
 2. Arrosseguem el control des del nostre explorador de solucions a la pàgina on vulguem incloure-ho i automàticament s'afegirà aquesta directiva.

Example

```
<%@ Page Language="C#" %>
<%@ Register TagPrefix="uc" TagName="Spinner"
    Src="~/Controls/Spinner.ascx" %>
<html>
<body>
<form runat="server">
    <uc:Spinner id="Spinner1"
        runat="server"
        MinValue="1"
        MaxValue="10" />
</form>
</body>
```

Links

- [http://msdn.microsoft.com/es-es/library/26db8ysc\(v=vs.80\).aspx](http://msdn.microsoft.com/es-es/library/26db8ysc(v=vs.80).aspx)
- <http://www.subgurim.net/Articulos/asp-net-general/121/como-hacer-un-control-de-usuario-ascx.aspx>
- Ejemplo un buscador.
- <http://csharp-experience.blogspot.com.es/2009/12/aspnet-eventos-de-controles-de-usuarios.html>