

Boletín de Servicios Web

HelloWorldWS

El objetivo de este primer ejercicio es hacer nuestro primer servicio Web. La primera operación es HelloWorld() que devolverá un saludo para todo el mundo:

```
[WebMethod]
public string HelloWorld()
{
    return "Hola a todos";
}
```

Para personalizar este saludo vamos a incorporar dos operaciones nuevas: SetName(), que sirve para personalizar el saludo, y Hello() que sirve para devolver el saludo personalizado. Para ello tenemos que hacer uso del estado de sesión:

```
[WebMethod(true)]
public void SetName(string name)
{
    Session["name"] = name + " (" + System.DateTime.Now + ")" + "--" + nameCount;
    nameCount++;
}

[WebMethod(true)]
public string Hello()
{
    return "Hola " + (string)Session["name"];
}
```

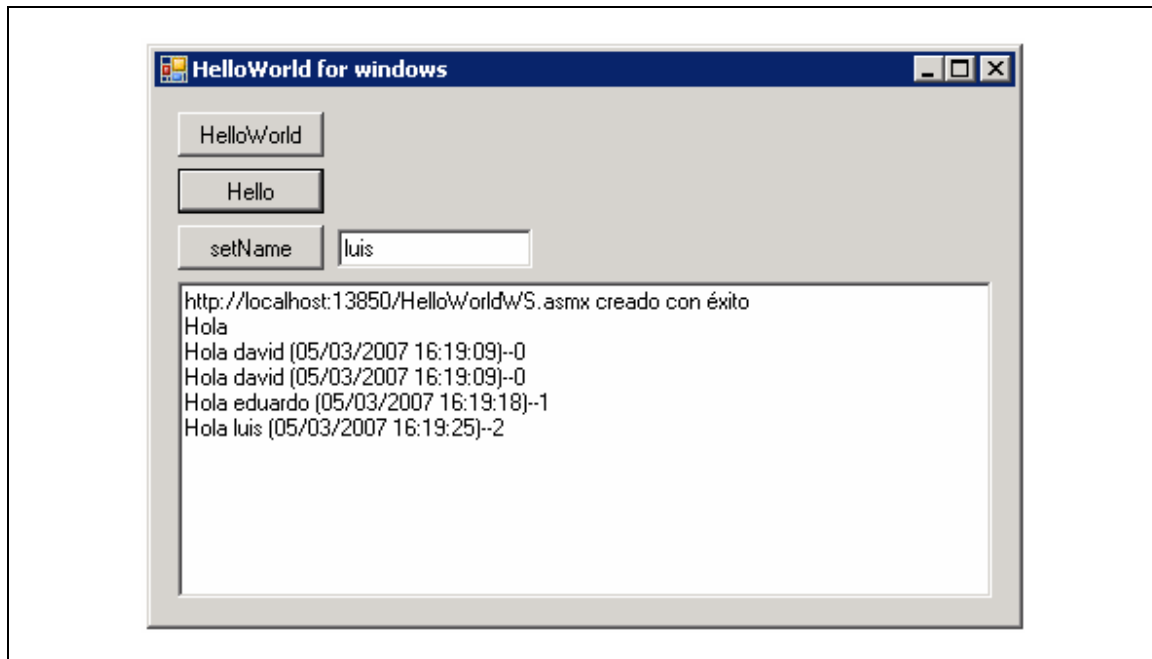
El formato del saludo personalizado es “Nombre (fecha) – número”, donde fecha nuestra el instante en el que se cambió por ultima vez el nombre y número indica el número de veces que se ha cambiado el nombre.

Se puede hacer uso de una propiedad privada para almacenar el número de veces que cambia un nombre con el siguiente Idiom:

```
private int nameCount
{
    get
    {
        if ( Session["nameCount"] == null)
            Session["nameCount"] = 0;
        return (int)Session["nameCount"];
    }
    set
    {
        Session["nameCount"] = value;
    }
}
```

HelloWorldWin

El objetivo de este ejercicio es hacer una aplicación Windows que use el servicio Web anterior. El interfaz es el siguiente:



Para hacer esto tenemos que agregar una referencia Web al proyecto de la aplicación Windows, al que se puede llamar localhost, esta referencia se usa para crear un objeto dentro de nuestra aplicación que apunte al servicio Web:

```
Private void mainFrm_Load(object sender, System.EventArgs e)
{
    ws = new localhost.HelloWorldWS();
    //credenciales del usuario que está ejecutando esta aplicación
    //ws.Credentials = System.Net.CredentialCache.DefaultCredentials;
    //si no usamos credenciales hay que dar acceso al usuario anónimo del IIS
    ws.CookieContainer = new System.Net.CookieContainer();
    resultTbx.Text += ws.Url + " creado con éxito";
}
private void helloWorldBtn_Click(object sender, System.EventArgs e)
{
    resultTbx.Text += "\r\n" + ws.HelloWorld();
}
private void helloBtn_Click(object sender, System.EventArgs e)
{
    resultTbx.Text += "\r\n" + ws.Hello();
}
private void setNameBtn_Click(object sender, System.EventArgs e)
{
    ws.SetName(setNameTbx.Text);
}
```

CalculatorWS

El objetivo de ejercicio es afianzar los conceptos anteriores. Para ello vamos a realizar un Servicio Web que implemente una calculadora con las siguientes operaciones:

- public float Add(float a, float b)
- public float Subtract(float a, float b)
- public float Multiply(float a, float b)
- public float Divide(float a, float b)
- public void SetMemory(float n)
- public float GetMemory()

Además vamos a utilizar una cookie para saber si este servicio ha sido utilizado alguna vez por el mismo cliente, para ello utilizaremos las operaciones:

- public Boolean HasBeenUsed()
- public DateTime LastUsed()

Para enviar una cookie en la respuesta HTTP que da IIS es necesario hacer lo siguiente:

```
HttpCookie c;  
c = new HttpCookie("lastUse", "" + System.DateTime.Now );  
c.Expires = DateTime.Now.AddYears(50);  
Context.Response.Cookies.Add(c);
```

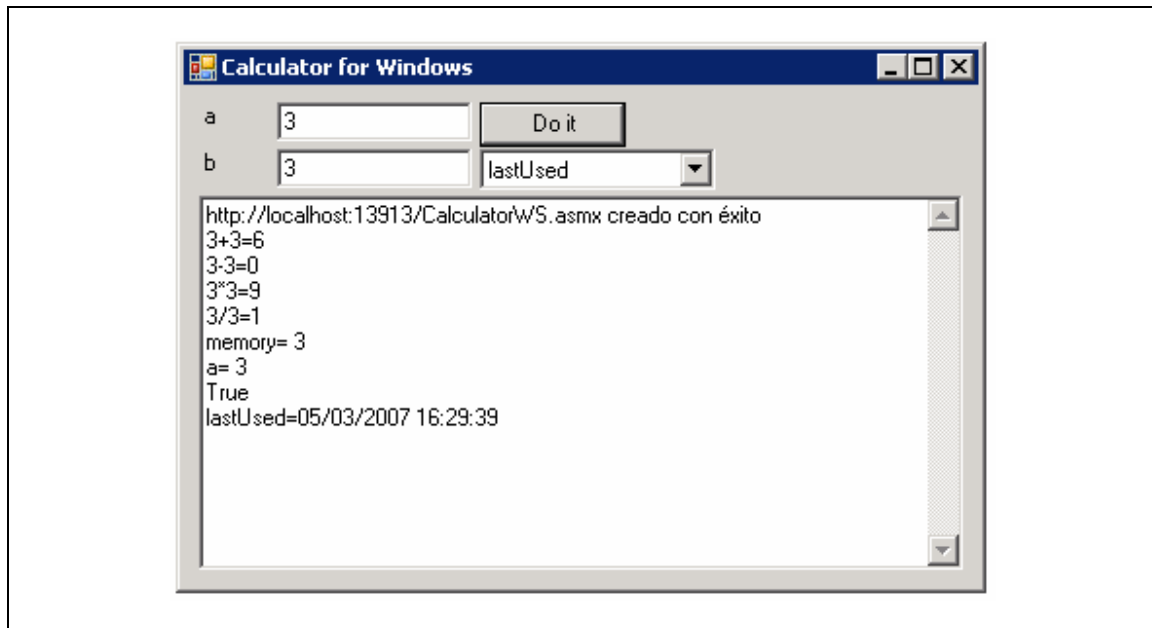
Por el contrario para obtener una cookie por parte del servicio Web lo que tenemos que hacer es:

```
HttpCookie c = Context.Request.Cookies["lastUse"];
```

Pruebe este servicio Web en Internet Explorer y busque el directorio en el que se guardan las cookies. Pruebe a borrar manualmente la cookie y observe el comportamiento.

CalculatorWin

El objetivo de este ejercicio es hacer una aplicación Windows que use el servicio Web anterior. El interfaz es el siguiente:

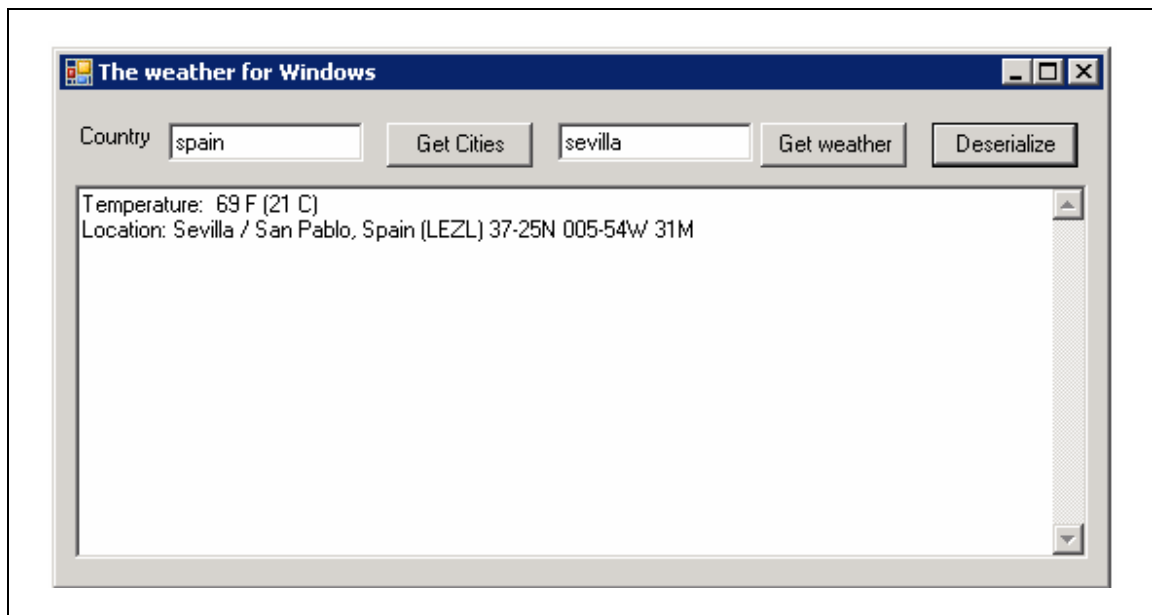


¿Dónde se almacenan ahora las cookies?

TheWeatherWin

El objetivo de este ejercicio es hacer una aplicación Windows que consuma un servicio Web proveído por un tercero. La descripción WSDL de servicio Web que tomaremos como ejemplo se encuentra en la siguiente URL <http://www.webserviceex.net/globalweather.asmx?WSDL>.

El interfaz de la aplicación a desarrollar es la siguiente:



En la caja de texto se mostrará el resultado de la llamada a los servicios `getCitiesByCountry` y `getWeather`, respectivamente. Además, deserializaremos el XML obtenido en la segunda llamada para crear un objeto y poder acceder a sus propiedades, la clase de objeto sería:

```
public class CurrentWeather
{
    public CurrentWeather(){}
    public string Location;
    public string Time;
    public string Wind;
    public string Visibility;
    public string Temperature;
    public string DewPoint;
    public string RelativeHumidity;
    public string Pressure;
    public string Status;
}
```

Y la forma de deserializar una cadena en formato XML en un objeto concreto sería:

```
CurrentWeather w;  
XmlSerializer serializer = new XmlSerializer(typeof(CurrentWeather));  
w = (CurrentWeather) serializer.Deserialize( new StringReader(resultTbx.Text));  
  
//uso del objeto  
resultTbx.Text = "Temperature: " + w.Temperature;  
resultTbx.Text += "\r\nLocation: " + w.Location;
```