

Capítulo 14

Gestión y administración del estado de páginas Web

Objetivo

Al finalizar el capítulo, el alumno:

- Identificar los principales elementos de una aplicación Web ASP.NET.
- Utilizar los objetos del modelo de objetos de ASP.NET.
- Identificar las diferentes formas de almacenamiento de sesiones.

Temas

1. Eventos de la aplicación
 - A nivel de aplicación
 - A nivel de sesión
2. Modelo de objetos ASP.NET
 - Application
 - Session
 - Caché
 - Server
 - Response
 - Request

1. Eventos de la aplicación

Eventos de la aplicación

Eventos a nivel de aplicación

- Application_Start
- Application_End
- Application_Error

Eventos a nivel de sesión

- Session_Start
- Session_End

Los objetos **Application** y **Session** permiten mantener información para ser usada en cualquier página de una aplicación ASP.NET. Esta información es mantenida por el servidor web.

El **objeto Application** expone tres importantes eventos:

- **Application_Start:** este evento es lanzado cuando el primer usuario realiza la primera solicitud al servidor. Si el servidor se reinicia, este evento se disparará en la próxima solicitud. La aplicación también se reinicia si se vuelve a realizar un despliegue, a través del Visual Studio.
- **Application_End:** este evento se dispara cuando el servidor, servicio web o website es detenido o reiniciado. Es un buen lugar para liberar los recursos que la aplicación ha usado. Tener en cuenta que no será disparado si ocurre una falla en la aplicación.
- **Application_Error:** este evento se dispara si una excepción no manejada ocurre en la aplicación. No obstante, si el error se manipula a través de un bloque try...catch, o a nivel de página, este evento no se disparará. Es un buen lugar para colocar código que se debe ejecutar ante errores inesperados que pueden ocurrir en la aplicación.

El **objeto Session** expone dos importantes eventos:

- **Session_Start:** este evento se dispara cuando un usuario inicia una sesión en la aplicación al momento de realizar una solicitud de página. Es un buen lugar para inicializar valores que son específicos a un usuario.
- **Session_End:** este evento se dispara cuando una sesión de usuario expira o cuando se invoca al método `Session.Abandon`.

2. Modelo de objetos ASP.NET

Modelo de objetos ASP.NET

- Application
- Session
- Caché
- Server
- Response
- Request



Objeto Application

Una aplicación ASP .NET está compuesta por páginas, archivos, manejadores de eventos y código que se encuentran en un determinado directorio virtual y subdirectorios. Asimismo, ASP .NET mantiene un único objeto Application para cada aplicación en el servidor Web. Este objeto es creado en la memoria cuando el primer usuario realiza la primera solicitud a la aplicación, y permanece en memoria hasta que la aplicación se detiene. Una aplicación se detiene cuando el servidor se apaga, el servidor Web es detenido o el sitio Web es detenido. Una aplicación también se puede detener por una falla en la aplicación o cuando se despliega una nueva versión de la aplicación.

Cuando un objeto es almacenado en el objeto Application, cualquier página en la aplicación Web puede acceder a este, sin importar qué usuario hizo la solicitud a la aplicación. Sin embargo, como la aplicación se puede reiniciar sin previo aviso, no se debería usar el objeto Application para almacenar información crítica y siempre se debería verificar que un valor existe en el caché antes de intentar usarla.

La siguiente línea de código muestra cómo guardar información dentro del objeto Application.

```
Application["Variable1"] = "Valor1";
```

Objeto Session

Es un objeto que representa una visita de un usuario a la aplicación Web. Una sesión empieza cuando un usuario realiza la primera solicitud de una página o archivo. Las sesiones expiran normalmente cuando ha transcurrido un tiempo determinado desde la última solicitud. Se puede configurar este tiempo cambiando el valor de la propiedad Session.Timeout. También se puede terminar una sesión invocando al método **Session.Abandon**.

Vía código:

```
Session.Timeout = 60;
```

Vía archivo de configuración:

```
<configuration>  
  <system.web>  
    <sessionState  
      mode="InProc"  
      cookieless="true"  
      timeout="30" />  
  </system.web>  
</configuration>
```

El objeto Session, así como el objeto Application, permiten compartir valores entre páginas ASP .NET. Sin embargo, cada usuario tiene su propio objeto Session; por lo tanto, se debería usar el objeto Session solo para almacenar información específica del usuario.

La siguiente línea de código muestra cómo guardar información en un objeto Session:

```
Session["Usuario"] = "Nombre de Usuario";
```

Objeto Caché

ASP .NET proporciona un objeto Caché (almacenamiento de determinados datos en memoria, los datos que se almacenan usualmente toman tiempo en crearse/conseguirse, por lo que este almacenamiento se hace con el propósito de no tener que volver a crear/conseguir esta información), a través del cual se accede al área caché de la aplicación. Se pueden agregar valores a la memoria caché de la aplicación, tal como se muestra a continuación:

```
Cache["value1"] = "Valor de ejemplo";
```

Si la memoria del servidor Web comienza a llenarse, el motor de ejecución de ASP .NET remueve los elementos ya utilizados de la caché. Si se intenta acceder a un objeto que ya fue removido de la caché, entonces se obtendrá un valor nulo; por lo tanto, siempre se debería verificar la existencia de un valor en la caché antes de utilizarlo, de lo contrario, puede ocurrir una excepción en tiempo de ejecución.

Objeto Server

El objeto Server proporciona métodos y propiedades del servidor IIS. Server es una clase de utilidad con varios métodos de aplicaciones auxiliares muy útiles, incluidos **Server.Execute()**, **Server.MapPath()** y **Server.HtmlEncode()**. Server es un objeto del tipo **System.Web.HttpServerUtility**.

Algunos de los métodos y propiedades de este objeto son:

- CreateObject (Crea la instancia de un objeto)
- MapPath (Mapea una ruta específica a una ruta física)
- URLEncode (Aplica reglas de encoding a una cadena específica)
- HTMLEncode (Aplica encoding de HTML a una cadena específica)

Objeto Response

El objeto Response contiene los métodos y propiedades necesarios para la creación de una respuesta HTTP. Incluye variables HTTP, cookies que serán almacenados por el navegador del cliente e información adicional acerca del contenido que se está enviando. El objeto Response es del tipo **System.Web.HttpResponse**.

Algunos de los métodos y propiedades de este objeto son:

- Expires (Configura cuánto tiempo, en minutos, una página permanecerá en el caché del browser).
- ExpiresAbsolute (Configura la fecha y hora en la que una página que está en el caché de un browser va a expirar).
- CacheControl (Configura si un servidor proxy puede colocar en caché el resultado de la respuesta que se está generando o no)
- Redirect (Redirige al usuario a una URL diferente)

El siguiente ejemplo muestra el uso del objeto **Response** para escribir un mensaje en la página que va a visualizar un usuario.

```
Response.Write("Esta línea se visualiza en la página web")
```

Objeto Request

El objeto Request contiene información sobre una solicitud HTTP incluyendo datos del explorador, cookies y valores pasados en un formulario o en un querystring. El objeto Request es del tipo **System.Web.HttpRequest**.

Algunos de las propiedades de este objeto son:

- Form (contiene todos los valores de entrada que son enviados por un formulario a través de un request)
- QueryString (contiene los valores de las variables del request)
- Cookies (Contiene todas los cookies enviados en el request)
- ClientCertificate (contiene todos los valores almacenados en el certificado del cliente)

El siguiente ejemplo muestra cómo obtener un valor de un QueryString enviado, a través del navegador Web del usuario y mostrarlo en la página.

URL: <http://localhost/AspEg/login.asp?nombre=John>

```
Valor1 = Request.QueryString("nombre").ToString()  
Response.Write(Valor1)
```