ASP.NET
Reputation
CRM
Quality
Goals
Innovation
Strategy
Solution
Analysis
Marketing
Management
Service
Team
Finance
Customers
Costs
Income
Plan

Jump2Learn
The Online Learning Place

ASP.NET

CHAPTER 4 : CLIENT -SERVER - COMMUNICATION

AUTHORS

Dr. Ashoksinh V. Solanki
M.C.A., Ph.D.
COLLEGE OF APPLIED SCIENCE AND PROFESSIONAL STUDIES, CHIKHLI

Mr. Yatin Solanki
M.C.A., Ph.D.(Pursuing)
DOLAT USHA INSTITUTE OF APPLIED SCIENCES, VALSAD

Mr. Chetan Parmar
M.C.A., NET, SET
NARMADA COLLEGE OF SCIENCE & COMMERCE, ZADESHWAR

❖ RESPONSE OBJECT

The Response object represents a valid HTTP response that was received from the server. The response header properties are read-only.

PROPERTIES OF THE OBJECT ARE AS FOLLOWS:

NAME	DESCRIPTION
Body:	Gets the body of the HTTP response. Only the portion of the body stored in the response buffer is returned
BytesRecv:	Gets the number of bytes the client received in the response
BytesSent:	Gets the number of bytes send in the HTTP request
CodePage:	Gets or sets the code page used for setting the body of the HTTP response
ContentLength:	Gets the size, in bytes, of the response body
Headers:	Gets a collection of headers in the response
HeaderSize:	Gets the combined size, in bytes, of all the response headers
HTTPVersion:	Gets the HTTP version used by the server for this response
Path:	Gets the path that was requested
Port:	Gets the server port used for the request
ResultCode:	Gets the server's response status code
Server:	Gets the name of the server that sent the response
TTFB:	Gets the number of milliseconds that have passed before the first byte of the response was received
TTLB:	Gets the number of milliseconds that passed before the last byte of the response was received
UseSSL:	Checks whether the server and client used an SSL connection for the request and response

SAMPLE CODE

The following sample code assumes that you have a Button control and four RadioButtoin controls with a common GroupName property on a Web Form. The user will be sent to a new Web Page if he makes a selection from one of the radio button's and hits the Submit button.

```
Private Sub Page_Load(ByVal sender As System.Object, ByVal e
As_System.EventArgs) Handles MyBase.Load

Dim ccc As String = "Response Object"

Response.Write(ccc)
Response.Write("Using Response object")

'using the response object's write method to write some text
```

```
End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As_System.EventArgs) Handles Button1.Click
If RadioButton1.Checked = True Then
Response.Redirect("http://www.google.com")
'using the response object's redirect method to redirect the user to another web page
ElseIf RadioButton2.Checked = True Then

Response.Redirect("http://www.amazon.com")
ElseIf RadioButton3.Checked = True Then

Response.Redirect("http://www.yahoo.com")
ElseIf RadioButton4.Checked = True Then

Response.Redirect("http://www.startvbdotnet.com")
End If

End Sub
```

The ASP.NET provides a class called ***HttpResponse*** which is defined in the namespace ***System.Web***. This class provides various methods and properties which help you use various information related to a web response.

An instance of this class is created by default in all the pages, so that you can use this object without creating again each time in all the pages. The name of this object is ***Response***.

FREQUENTLY USED METHODS AND PROPERTIES OF RESPONSE OBJECT:

RESPONSE.WRITE

This method is used to write dynamic text to the web page.

```
Response.Write(DateTime.Now.ToString())
```

The above code will generate the current time as text and display to the user. But note that the text will be displayed on the top of the page. There is no way you can specify the location and format the text.
If you want to move the location of the text, you may have to do something like below:

```
Response.Write("<table width=500>")

Response.Write("<tr>")
Response.Write("<td align=right>")

Response.Write("<BR>")
Response.Write("<BR>")
Response.Write("<BR>")
Response.Write(DateTime.Now.ToString())
Response.Write("</td>")
Response.Write("</tr>")
Response.Write("</table>")
```

ASP.NET provides several web controls including the Label control which allow you to specify the exact location where you want the control to be displayed. Due to this, the Response.Write() method is not widely used now.

RESPONSE.COOKIES

Cookies are used to store small pieces of information in the client computer. In ASP.NET, the Response object is used to send cookies to the client browser. If you select the 'Remember Me' option at the time of login, we use the following code to store your user id in a cookie in your computer. When you come back to this site later, we retrieve the user id from the Request and load your information automatically, without asking you to login again.

```
Response.Cookies("UserId").Value = "your user id"

Response.Cookies("UserId").Expires = DateTime.Now.AddDays(7)
```

We use the above code to store your user id in a cookie. The name of the cookie used is "UserId". The cookie will expire after 7 days. This means, if you come back to our site after 7 days, we will not remember you.

RESPONSE.REDIRECT()

This may be the most frequently used method of the **Response** object. **Response.Redirect()** is used to redirect user from one page to another page or website.

```
Response.Redirect("Login.aspx")
```

When the above line of code is executed, the page will be redirected to the login page.

❖ STATE MANAGEMENT

- Generally, web applications are based on stateless HTTP protocol which does not retain any information about user requests.
- In typical client and server communication using HTTP protocol, page is created each time the page is requested.
- Web pages are stateless and are HTTP-based, which means that it does not automatically indicate whether a sequence of request is all from the same client or a

new client. Every time a client request a page, a new instance of the page is returned and after each roundtrip the page is destroyed.

- State management is defined as the management of the state of one or more user interface controls such as textboxes, buttons, etc.. in a graphical user interface.
- In UI programming technique, the state of UI control depends on the state of other UI controls.
- Developer is forced to implement various state management technique when developing applications which provide customized content and which "remembers" the user.
- A new instance of the web page class is created each time the page is posted to the server.
- In traditional web programming, this would typically mean that all the information associated with the page and the controls on the page would be lost with each round trip.
- **Round trip:-** Whenever a user interaction require processing on the server, the web page is posted back to the server processed and is returned back to the browser. This sequence is called round trip.

- **THE PROCESSING CYCLE FOR AN ASP.NET WEB PAGE:**

  1. The user request the page
  2. The page dynamically renders markup to the browser, which the user sees as a web page similar to any other page.
  3. The user types information or selects from available choices and then clicks a button.
  4. The page is posted to the web server. Specifically, the page is posted back to itself.
  5. on the web server, the page runs again. The information that the user typed or selected is available to the page.
  6. The page performs the processing that you have programmed it to do.
  7. The page renders itself back to the browser.

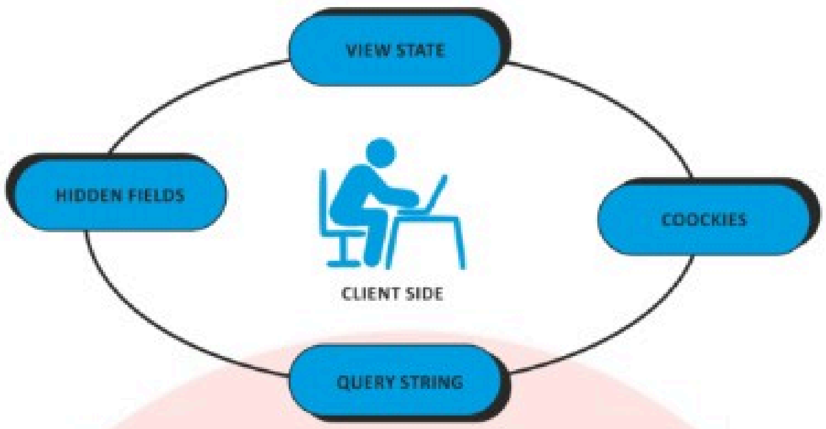
This cycle continues as long as the user is working in the page. Each time user clicks a button, the information in the page is posted to the web server and the page runs again.

Each cycle is referred to as a round trip.

STATE MANAGEMENT

- State management can be divided into two categories:
  1. Client-side management.
  2. Server-side management.

1. CLIENT SIDE MANAGEMENT:



- The following sections describe options for state management that involve storing information either in the page or on the client computer.
- For these options, no information is maintained on the server between round trips.

VIEWSTATE:-

- This is the default method that the page uses to preserve page and control property values between round trips.
- View State can be used to store state information for a single user.
- We can store page specific information.
- View state is a built in feature in web controls to persist data between page post backs.
- Each web page and controls on the page have the EnableViewState property. Using this property you can set View state on/off for each controls.
- By default, EnableViewState property will be set to true.
- ASP.NET framework uses the ViewState property to automatically
- save the values of the web page and each control on the web page prior to rendering the page.
- You can also disable View state for the entire page by adding EnableViewState=false to @page directive.
- The View State is implemented with a hidden form field called _VIEWSTATE, which is automatically created in every web page.
- When page is created on web server this hidden control is populated with state of the controls and when page is posted back to server this information is retrieved and assigned to controls.
- When the page is posted back to the server, the page parses the view-state string at page initialization and restores property information in the page.
- View state mechanism poses performance overhead.
- You can store values in view state as well.

HIDDEN FIELD:-

- Hidden fields are used to store data at the page level.
- As its name says, these fields are not rendered by the browser.
- A hidden field stores a single variable in its value property and must be explicitly added to the page.
- It's just like a standard control for which you can set its properties.
- Whenever a page is submitted to server, hidden fields values are also posted to server along with other controls on the page.
- All the asp.net web controls have built in view state property for state of the control, hidden fields functionality seems to be redundant.
- We can still use it to store insignificant data.
- Do not store any information in a hidden field that is sensitive.
- HiddenField have less number of properties compare to other web server control.

COOKIES:-

THERE ARE TWO TYPES OF COOKIES PERSISTENT AND NON PERSISTENT COOKIES

- persistent cookies are permanent, example of persistent is remember me option used in most login pages. the user name and password are stored for long time
- non persistent cookies are temporarily used for storage, for short span of time
- persistent cookies and non-persistent cookies otherwise called as memorizable cookies and non-memorizable cookies.
- Persistent cookie means the cookie will be expired as soon as the application is closed
- Non-persistent cookie means even the application is closed the data will be remained as per the cookie timeout value.

WHAT ARE COOKIES?

Cookies are small pieces of information stored in the client computer. They are limited to 4K in size. Session Cookies and Persistent Cookies are two types of cookies.

Session Cookies are stored in memory during the client browser session. When the browser shutdown the session cookies are lost.

Example #1:

```
        Dim objCookie As New HttpCookie ("Username", "Jim")
        Response.Cookies.Add (objCookie)
```

READ COOKIE

```
        Request. Cookies ("Username"). Value
```

Persistent Cookies work the same way as session cookies. The difference between the two is that persistent cookies have an expiration date.

Example #2:

```
        Dim objCookie As New HttpCookie ("Username"," Jim")
        ObjCookie.Expires "#12/31/2005#"
        Response.Cookies.Add (objcookie)
```

Read Cookie

```
Request. Cookies ("Username"). Value
```

How to Create a Cookie?

The "Response.Cookies" command is used to create cookies.

Example #3:

```
Response.Cookies("myName")("sarav") = "Saravanan"
        Response.Cookies("friends")("selva") = "Selvakumar"
        Response.Cookies("friends")("venkat") = "Venkatesh"
        Response.Cookies("friends")("hari") = "Hariraam"
        Response.Write("Cookies Created!")
```

HOW TO RETRIEVE A COOKIE VALUE?

The "Request.Cookies" command is used to retrieve a cookie value.

Example #4:

```
        Response.Write("My Name is:" & Request.Cookies("myName")("sarav"))
```

Output:

```
My Name is: Saravanan
```

HOW TO READ ALL COOKIES?

If a cookie contains a collection of multiple values, we say that the cookie has Keys. In the example below, we will create a cookie collection named "friends". The "friends" cookie has Keys that contains information about a friends:

Example #5:

```
Dim fnd As String
For Each fnd in Response.Cookies("friends").Values
Response.Write(Request.Cookies("friends")(fnd) & "")
Next
```

Output:

```
Selvakumar
Venkatesh
Hariraam
```

ADVANTAGES:

i) It can be used to store information about a particular client.
ii) It is simple to use.
iii) It is quite secure.
iv) No server resources.

DISADVANTAGES:

i) It has limited size.
ii) No security.

LIMITATIONS OF COOKIES:

(1) it can't store complex data (like dataset ..)

(2) cookies stored in client machine only so there is no SECURITY then what about the data of very securable (like passwords,creditcard details etc..)

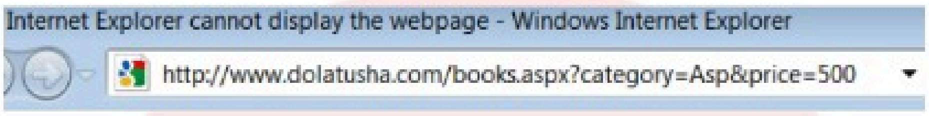
To overcome HTTPcookies limitations, SESSIONS are solution

Persistent cookies are stored on your computer hard disk. They stay on your hard disk and can be accessed by web servers until they are deleted or have expired. Persistent cookies are not affected by your browser setting that deletes temporary files when you close your browser

.Non-persistent cookies are saved only while your web browser is running. They can be used by a web server only until you close your browser. They are not saved on your disk. Microsoft Internet Explorer 5.5 can be configured to accept non-persistent cookies but reject persistent cookies

QUERYSTRING:-

- A query string is information that is appended to the end of a page URL.
- Query strings are usually used to send information from one page to another page. They are passed along with URL in clear text.
- This information is passed in URL of the request.
- A typical URL with a query string looks like:


Internet Explorer cannot display the webpage - Windows Internet Explorer

http://www.dolatusha.com/books.aspx?category=Asp&price=500

- In the URL path above, the query string starts with a question mark (?) and includes two attribute/value pairs, one called "category" and the other called "price".
- We can only pass smaller amounts of data using query strings.
- Most browsers impose a limit of 255 characters on URL length.
- Since Query strings are sent in clear text, we can also encrypt query values.
- Syntax:-

SENDING ONLY ONE QUERYSTRING VALUE.

Sender page

```
Response.Redirect("PageName?KeyName=Value")
```

Reciever page

```
Str=Request.QueryString("KeyName")
```

SENDING ONLY ONE QUERYSTRING VALUE.

Sender page

```
Response.Redirect("PageName?KeyName1=Value1&
KeyName2=Value2")
```

Reciever page

```
Str1=Request.QueryString("KeyName1")

Str2=Request.QueryString("KeyName2")
```

SUMMARY OF CLIENT SIDE STATE MANAGEMENT

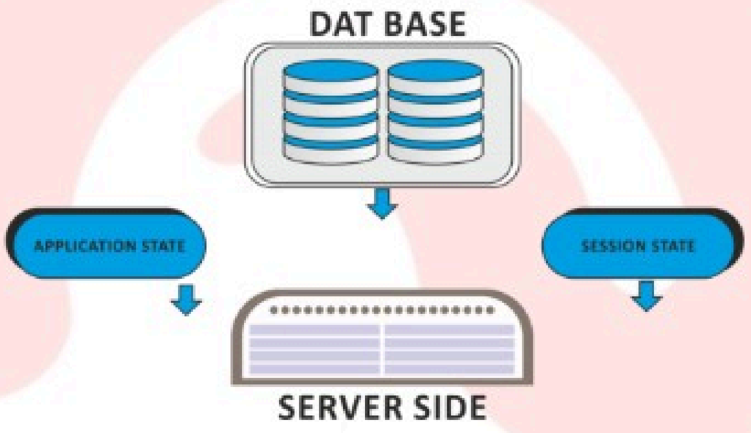
View State:- Use when you need to store small amounts of information for a page that will post back to itself.

Hidden Fields:- As above and security is not an issue that time use.

Cookies:- Use when you need to store small amounts of information on the client PC and security is not an issue.

Query String :- Use when you are transferring small amounts of information from one page to another and security is not an issue.

2. SERVER SIDE MANAGEMENT:



- With server-based state management, you can decrease the amount of information sent to the client in order to preserve state, however it can use costly resources on the server.

APPLICATION:-

- Application object provided a great place to store frequently used pieces of data that changed infrequently, such as the contents of menus or other reference data.
- ASP.NET implements application state using the System.Web.HttpApplicationState class.
- It provides methods for storing information which can be accessed globally.
- Information stored on application state will be available for all the users using the website.
- Application object is used to store data which is visible across entire application and shared across multiple user sessions.
- The application object provides a mechanism for storing that is accessible to all code running within the web application.

- The information that is global to the application may be stored in application object.
- For efficiency, this state is typically stored once and then read many times.
- A collection of user-defined variables which are shared by an ASP.NET application are termed as application state.
- When the Application_OnStart event fires at the time of loading of the first instance of the applications these variables are set and initialized and are available till the last instance of the application exits.
- Application state is stored in the memory of the windows process which is processing user requests on the web server.
- Application state is useful in storing small amount of often-used data.
- Unlike session state, which is specific to a single user session, application state applies to all users and all sessions.
- Syntax:- Application("KeyName") = Value
- Ex:- Application("Name") = "Yatin"

SESSION:-

- ASP.NET allows you to save values by using session state, which is an instance of the HttpSessionState class, for each active web-application session.
- A cookie is very simple and is not suitable for sophisticated storage requirements.
- Session state gives a method to keep more complex objects securely.
- ASP.NET allows programmers to keep any type of objects in session.
- Like Querystring we can use session to store value and pass it to another page, unlike Querystring, session will not display on the end of the URL.
- Session state is similar to application state, except that it is scoped to the current browser session.
- A session is the time for which a particular user interacts with a web application.
- Every client that uses the application will have separate sessions.
- During session the unique identity of the user is maintained internally.
- Session state is ideal for storing user specific information.
- If different users are using your application, each user session will have a different session state.
- If a user leaves your application and then returns later, the second user session will have a different session state from the first.
- Do not store large quantities of information in session state.
- A collection of user-defined session variables are termed as Session State, which are persevered during a user session.
- These variables are accessed using a session collection. These variables have a unique instance to different instances of a user session for the application.
- Session variables can be set to automatically destroyed after a definite period of inactive time, irrespective of the session state whether session ends or not.
- Syntax:- Session("KeyName") = Value
- Ex:- Session("Name") = "Chetan"

- **IMPORTANT METHODS/PROPERTIES OF SESSION.**

- **Session.Abandon :-** cancels the current session.
- **Session.Remove :-** Delete an item from the session-state collection.
  Ex- Session.Remove(Uname)
- **Session.RemoveAll :-** Deletes all session state items.
- **Session.SessionID :-** Get the session Id read only property of a session for the current session.
- **Session.Timeout :-** if a user does not request a page of the ASP.NET application within certain minutes then the session expires.
  Ex- Session.Timeout = 30

DATABASE:-

- Database enables you to store large amounts of information pertaining to state in your web application.
- Some times users continually query the database by using the unique ID, you can save it in the database for use across multiple request for the pages in your site.
- You can store as much information as you like in a database.
- Access to databases requires authentication and authorization.

SUMMARY OF SERVER SIDE STATE MANAGEMENT

Application :- Use when you are storing infrequently changed, global information that is used by many users, and security is not an issue.

Session :- Use when you are storing short-lived information that is specific to an individual session and security is an issue.

Database :- Use when you are storing large amount of information, managing transaction, or the information must survive application and session restarts.