

---

# Asp.Net

---

## Ch. 02

## State Management

# What is State management ?

- In the web world, for transferring the information from one place to another we are using **HTTP** (Hyper Text Transfer Protocol) protocol.
- HTTP is a communication protocol which is used by **World Wide Web**.
- Clients will send request to server and server responds to these requests.

# What is State management ?

- HTTP uses **TCP** (Transmission Control Protocol) a protocol for communication between client and server.
- These all communications are done via ports.
- Once the client receives the response, client will be disconnected from the server.
- For any new request the same process is followed again and again.

# What is State management ?

- ASP.Net files are like text files that are located on server.
- On each request for a page, the server locates the requested file. Then the ASP.Net engine produces the response for that file by processing the server tags and generate HTML format file for the client.

# What is State management ?

- As HTTP is a stateless protocol, once the clients gets the response it terminates the connection with server and does not save any information on HTTP. It means a new instance of web page is created each time the page is posted to the server.
- It also means that all information associated with page and the controls on the page are lost with each page request response.
- To overcome the limitations of Web programming, State management is used.

# What is State management ?

- The current value of all the controls and variables for current user in the current session is called state of web page.
- State management techniques are used to maintain state and page information of web page through the application for same or multiple requests.

# Types of State Management

- In ASP.Net there are two types of state management
  - Client Side State Management
  - Server Side State Management
- Client Side State Management
  - In Client Side State Management information is stored on client's computer by embedding the information into a web page or into a URL or into a cookie.
  - To store this information some techniques are used like,
    - View State, Control State, Hidden Files, Cookies, Query String

# Client Side State Management

## ■ View State

- ❑ ASP.NET uses view state to store state of the page and all its controls.
- ❑ View State is automatically maintained by ASP.NET framework.
- ❑ When a page is sent to the client as per request, the changes in properties and controls of pages are determined and stored in the value of hidden input field.
- ❑ This file is name as `_VIEWSTATE`.
- ❑ When the page is post back the `_VIEWSTATE` field is sent to the server with HTTP request.



# Client Side State Management

## ■ Control State

- ❑ It is new mechanism implemented by ASP.NET 2.0.
- ❑ It is used to store critical, private information across post backs.
- ❑ It is another type of state container which is reserved for controls core behavioral functionality.
- ❑ It is address some of the shortcomings of View State. It shares some memory data structures with View State.
- ❑ It is done even through the View State control is disabled.

# Client Side State Management

## ■ Hidden Fields

- ❑ They are used to store data at Page Level.
- ❑ This field are not rendered by web browser.
- ❑ This data is available only when the form is processed.
- ❑ We can set properties of these controls.

# Client Side State Management

## ■ Cookies

- It is small piece of text stored on user's computer. The browser sends this information with every page request to the same server. It is used by websites to keep track of visitors.

## ■ Query Strings

- They are used to store values in the URL that are visible to user.
- Query String is information sent from one page to another page.

# Server Side State Management

- In Server Side State Management information is stored on server's computer using.
  - Application State
  - Session State
- **Application State**
  - Application state allows us to save values which is an instance of HTTP Application State class for each active web application.
  - It is a global storage mechanism which is accessible from all pages of Web Application.

# Server Side State Management

- ❑ Application state is useful for storing information between server round trips and between requests for pages.

## ■ Session State

- ❑ Session state is quite opposite to Cookies.
- ❑ Cookies are used to store the data on client side while session are used to store the data on Server Side.

# Client Side State management Control

- **View State management :**
  - ❑ View State is used to store information of single user.
  - ❑ It is a built in feature of ASP.Net that stores data of controls and page information in form of variable named `_VIEWSTATE`.
  - ❑ `_VIEWSTATE` is a hidden variable created by ASP.NET.
  - ❑ `_VIEWSTATE` will be automatically created and is assigned an encoded value.

# Client Side State management Control

- The View state can be enabled or disable :
  - Enable or Disable View State for Entire App
    - By setting the `EnableViewState` property of Page Attributes in `Web.Config` file.
  - Enable or Disable View State for A Page
    - By setting the `EnableViewState` attribute for Page directive, as

```
<%@Page Language="C#"
```

```
    EnableViewState="false" %>
```

# Client Side State management Control

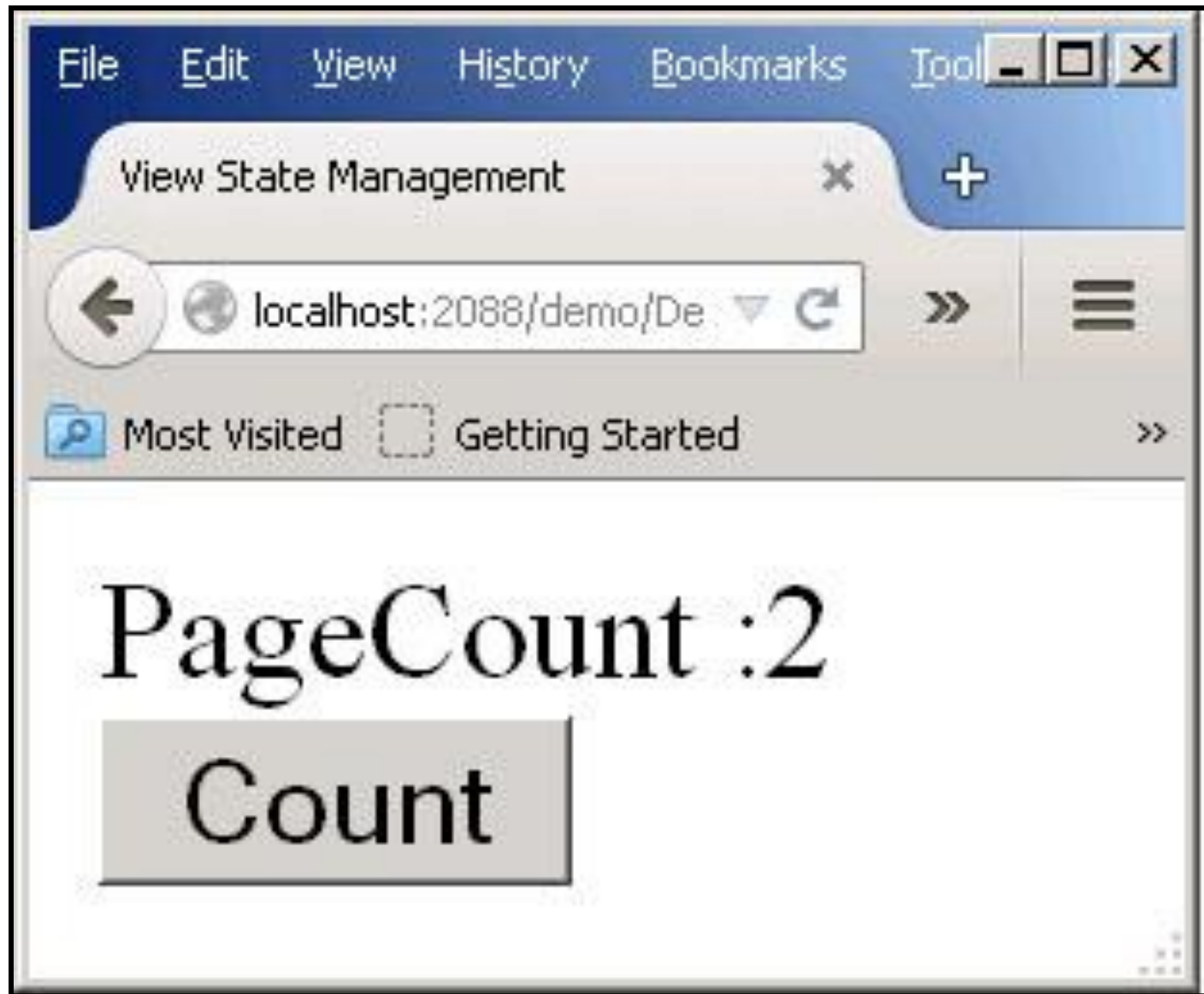
- Enable or Disable View State for A control
  - By setting the EnableViewState property of individual control.
  - View state is defined by State Bag Class which defines a collection of view state items.
  - The State Bag is a data structure which contains attribute/value pairs.
  - These pairs are stored as strings which are associated with the objects.



# Advantages & Disadvantages

- **Advantages of View State Management :**
  - ❑ View State management is quite simple and it is easily applied at Page Level Data.
  - ❑ All fields and controls are encrypted and then stored in View State Variable.
  - ❑ View State can be set at control level to.
- **Disadvantages of View State Management :**
  - ❑ View State management is overhead for ASP.Net as each time it has to encoded View State Value.
  - ❑ This variable makes the pages quite heavier.

# Def. \*\* Demonstrate for ViewState



# Step to demonstrate view state.

## ■ Step-1

- ❑ Create an empty website and add a web form in it.

## ■ Step-2

- ❑ Add a label "Page Count" and a button "Add Count" on the web page.

## ■ Step-3

- ❑ Goto CS code file.

```
public partial class _Default :  
    System.Web.UI.Page  
{
```

# Step to demonstrate view state.

```
public int counter
{
    get
    { if (ViewState["lblAddCount"] != null)
      { return ((int)ViewState["lblAddCount"]);
      }
      else
      { return 0;
      }
    }
}
```

# Step to demonstrate view state.

```
set
```

```
{
```

```
    ViewState["lblAddCount"] = value;
```

```
}
```

```
}
```

```
protected void btnAddCount_Click(object sender, EventArgs e)
```

```
{    lblAddCount.Text = "Page Count : " +  
        counter.ToString();
```

```
    counter++;
```

```
} }
```

# Disable View State :

- When we click on button control it will add 1 to current ViewState value each time.
- To disable View State
  - Add EnableViewState property as false in the page attribute as given below.

```
<%@ Page Language="C#"
    AutoEventWireup="true"
    CodeFile="Default.aspx.cs"
    Inherits="_Default"
    EnableViewState="false" %>
```

# Hidden Fields :

- Hidden fields are the data which is not visible to the user at runtime.
- Hidden fields are sent back to server when the user submits a form.
- However, the information stored in hidden fields is never displayed to the user unless the user views the page source of the webpage.
- Hidden field control stores a single variable in its value and property.
- Hidden fields are used to store data at page level.
- If you use hidden fields, you must submit your page to the server using HTTP Post method rather HTTP Get method.

# Hidden Fields :

- Property of Hidden fields :
  - Value
    - It is used to set the value of given Hidden Fields control.
- Advantages of Hidden Field :
  - Hidden field is quite simple to implement at page specific data level.
  - It is used to store small amount of data so they take less size.
- Disadvantages of Hidden Field :
  - Hidden field is quite inappropriate (अयोग्य) for sensitive data.
  - Hidden field values can be visible when only passed over a network.



# Query String Control :

- Query String are commonly used to store variables that are specific to given page.
- A query string is information that is added to the end of a page URL.
- A typical query string *or* URL may look like :  
`http://www.micro.com/Default.aspx?k=444`
- The query string starts with "?" mark.

# Query String Control :

- Advantages of Query String :
  - ❑ Query String is quite simple to Implement.
- Disadvantages of Query String :
  - ❑ Query string is always available in Human Readable format.
  - ❑ It has URL length limitation as length of Client browser is limited.
  - ❑ Cross paging functionality can makes it unusual (अन्यथा ).
  - ❑ It can be easily modified by end users.

# Cookies :

- Web application store small piece of information in Client's web browser using cookies.
- A cookie is a small file that is stored on client side.
- A cookie can have a maximum size is 4KB.
- Cookies can be stored either in a text file on the Client Machine or in memory in the Client Browser session.
- The location where the cookies are stored in controlled by the browser.

# Cookies :

- The web server create a cookie then attaches an additional HTTP header to the response and sends it to the browser.
- The browser would create cookie in Client's computer and include this cookie for all requests made to the same client.
- Server can read value of cookie from the request and maintain that state.
- For each different web applications different cookies are maintained.

# Cookies :

- Cookies are used to keep the track of visitors on a website.
- All the cookies would be cleared by browser immediately when we close Client's browser.
- Cookies can be also created by using `HttpCookie` class.

- Syntax :

```
HttpCookie userInfo=new HttpCookie();  
userInfo["UserName"]="ABC";
```

- Here, **userInfo** is the name of `HttpCookie` object created, **userName** is cookie variable and **ABC is** the value of the cookie.

# Cookies :

## ■ Some of the properties of Cookies :

Property	Meaning
Domain	It is used to specify the domain associated with the cookie. Its default value is current domain.
Expires	It is used to specify the expiry time of the cookie. It is specify by using DateTime object.
HttpOnly	It is used to enable or disable the access of cookies through javascript
Secure	It is set when the cookies are transmitted over SSL.

# Cookies :

- Some of the properties of Cookies :

Property	Meaning
Name	It is used to set the name of the cookie.
Value	It is used to set the value of the cookie.
Path	It allows us to set the value of the cookie.
TimeStamp	It provides us data and time of cookie when it was created.

# Cookies :

- We can write a cookie by using the page's response property.
- The Response object supports a collection named cookies, to which we can add the cookies you want to write to the browser.
- Example

```
Response.Cookies["userName"].Value="abc"
```

- Here, **userName** is the name of the cookie variable and **"abc"** is the value of assigned to that cookie variable.



# Cookies :

- We can retrieve cookie value from response object by using the request object.
- Example

```
string userName=
```

```
Request.Cookies["userName"].Value
```

# Def. Demonstrate Use Of Single Cookie.

## ■ Step-1

- Add a new Web Site & Web Page.

## ■ Step-2

- Add a Label : ID=lblCookie
- Add a Button : ID=btnShowCookie

## ■ Step-3

```
protected void Page_Load(object sender, EventArgs e)
{
    HttpCookie c = new
        HttpCookie("UserName", "MONARCH
        Computer");
    Response.Cookies.Add(c); }
}
```

# Def. Demonstrate User Of Cookie.

- protected void btnCookie\_Click(object sender, EventArgs e)  
{ lblCookie.Text=Request.Cookies  
["UserName"].Value;  
}

# MultiValued Cookies :

- Multivalued cookies are nothing but a single cookie which can store multiple values.
- In ASP.NET we can create multivalued cookies in same way as we create a cookie.

# Def. Demonstrate use of MultiValued Cookies :

```
protected void Page_Load(object sender, EventArgs e)
{
    HttpCookie c = new
        HttpCookie("userName","Monarch");
    Response.Cookies.Add(c);
    Response.Cookies["userName"]["age"] = "28";
    Response.Cookies["userName"]["city"] =
        "Jamnagar";
}
```

# Def. Demonstrate use of MultiValued Cookies :

```
protected void btnShow_Click(object sender, EventArgs e)
{
    Response.Write("<p>User Name : " +
        Request.Cookies["userName"].Value);
    Response.Write("<p>User Age : " +
        Request.Cookies["userName"]["age"]);
    Response.Write("<p>User City : " +
        Request.Cookies["userName"]["city"]);
    Response.Write("<p>User Value Count : " +
        Request.Cookies["userName"].Values.Count);
    Response.Write("<p>User Values:" + Request.
        Cookies["userName"].Values.ToString(); }
}
```

# Advantages of Cookie :

- Cookie is quite transparent, so user can easily read data from it.
- Cookie is used to store user preference information on the client machine.
- Cookie is used quite easy way to maintain cookies.
- Cookie is quite fast accessing.

# Disadvantages of Cookie :

- If any user remove or clears cookie information then we cannot get it back.
- No security provided by cookie.
- Each request would generate cookie information with page much over load on client machine.



# Server Side State Management

- For server side State Management we can use two state.
  - Session State
  - Application State

# Session State :

- Cookies are very simple and are not suitable for complex storage requirements.
- Session State is used to store complex object securely.
- ASP.Net allows programmers to save any type of objects in Session.

# Session State :

- Data stored in session is kept in server memory and it is protected as it would never be transmitted to client.
- Whenever a user connects to any ASP.Net website, a new session transmitted to client.
- For each request from the Client a new session state object will be created.

# Session State :

- Session state stores information about the user and keeps the track of pending operations of the users.
- Each session is identified and tracked with 120-bit Session ID.
- This SessionID is passed from client to server and is sent back as a cookie called ASP.NET\_SessionID.

# How Session works ?

- ASP.NET maintains unique id for each session called Session ID.
- This session ID is generated using a custom algorithm.
- Session is present on client as a cookie.
- The browser resends this unique id each time a new request is generated.

# How Session works ?

- To get session ID we can use...

```
string sessionID = Session.SessionID;
```

- If we don't store anything in session then ASP.NET will generate a different session id for each request else use same session ID.

# Application State Management

- ASP.NET provides saving values using application state.
- It is a global storage in application key value dictionary.
- Once application specific information is added to application state, then server manages it, and it is never exposed to the client.

# Application State Management

- Adding data to application state makes the data accessible to all pages from a single location in memory.
- Data stored in application state object is not permanent, it gets lost any time when the application is restated.
- ASP.NET creates an application state object for each application by using the `HTTPApplicationState` class and stores this object in server memory.



# Application State Management

- Application state is used mostly to store
  - Hit counter and other Statistical data,
  - Global application data like
    - tax rate, discount rates etc
  - It also used to keep track of users visiting the web application.