

[ASK A QUESTION](#)[CONTRIBUTE](#)

## State Management Concept In ASP.NET

In this article I will explain the state management concept in ASP.NET. State management is a preserve state control and object in an application because ASP.NET web applications are stateless.



Rakesh

Mar 19 2019

6

18

92.7k

[Download Free .NET & JAVA Files API](#)

[PLAY THE SECURE CODING GAME & WIN A FREE T-SHIRT](#)

State management is very important and useful in ASP.NET. It is also asked in many interviews to fresher and experienced developers.

State management is a preserve state control and object in an application because ASP.NET web applications are stateless. A new instance of the Web page class is created each time the page is posted to the server. If a user enters information into a web application, that information would be lost in the round trip from the browser (MSDN).

In a single line, State management maintains and stores the information of any user till the end of the user session.

Two types of State Management techniques are available in ASP.NET as in the following figure,



Figure 1: State Management Part

## Server side

### Session

Session is a very important technique to maintain state. Normally session is used to store information and identity. The server stores information using Sessionid.

#### Set User Session

```
01. protected void btnSubmit_Click(object sender, EventArgs e)
02. {
03.     Session["UserName"] = txtName.Text;
04.
05.     Response.Redirect("Home.aspx");
06. }
```

#### Session Event

Session event can be seen in project *Global.asax* file.

[ASK A QUESTION](#)[CONTRIBUTE](#)

```
01. void Session_Start(object sender, EventArgs e)
02. {
03.     Session["master"] = "~/Master.master";
04. }
```

## Session\_End

The Session\_End event is raised when session is ended by a user or a time out using Session end method.

```
01. void Session_End(object sender, EventArgs e)
02. {
03.     Response.Write("Session_End");
04. }
```

The session is stored in the following for ways in ASP.NET.

- *InProcMode*  
It is a default session mode and a value store in web server memory (IIS). In this the session value stored with server start and it ends when the server is restarted.
- *State Server Mode*  
In this mode session data is stored in separate server.
- *SQL Server Mode*  
In this session is stored in the database. It is a secure mode.
- *Custom Mode*  
Generally under session data is stored in InProc, Sql Server, State server, etc. If you store session data with other new

[ASK A QUESTION](#)[CONTRIBUTE](#)

There are three types of applications in ASP.NET.

### Application\_Start

This event begins with domain start.

```
01. | Void Application_Start(object sender, EventArgs e)
02. | {
03. |     Application["AppstartMessage"] = "Welcome to CSharp Corner Developer Community";
04. | }
```

### Application\_Error

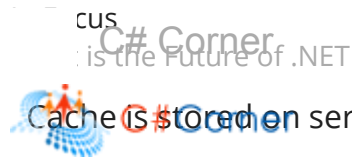
In this section manage unhandled exception error.

```
01. | void Application_Error(object sender, EventArgs e)
02. | {
03. |     // Write an unhandled error code exception
04. | }
```

### Application\_End

This ends with domain or restarts IIS.

```
01. | Void Application_End(object sender, EventArgs e)
02. | {
03. |     Application["AppEndMessage"] = "Application Closed";
04. | }
```

[ASK A QUESTION](#)[or](#)[CONTRIBUTE](#)

# Client Side

Now here I am explaining client side state management one by one:

Also state management has the following four important parts available on the client side,

## Cookie

Cookie is a small and an important part of ASP.NET. In this store user information, session and application. It can be created constant and temporary and they work with browser request. Cookies are store on client side. The server can read cookies and abstract data.

Two types of cookies are available,

### Persistence

This type of cookie works with Date and time.

```
01. Response.Cookies["CookieName"].Value = "Test Cookies";  
02. //set expire time  
03. Response.Cookies["CookieName"].Expires = DateTime.Today.AddHours(1);
```

### Non-Persistence

This is a temporary cookie. It is created with access application and discards the close application.

```
01. Response.Cookies["CookieName"].Value = "Test Cookies";
```



Control state technique is developed to maintain data work properly in order. We disabled by the user, the control will not work as expected. For expected results of the control we have to use Control State. In

```
01.  if (!IsPostBack)
02.  {
03.      lblmsg1.Text = "Welcome to C# corner";
04.      lblmsg2.Text = "Welcome to C# corner community";
05.  }
```

When two messages are displayed on a Postback event, then control which one is displayed by using customized control state.

## Hidden Field

Hidden fields are used to store value to client side. Hidden field is not displayed on the browser, but it works on a request.

```
01.  if (HiddenField1.Value != null)
02.  {
03.      int val = Convert.ToInt32(HiddenField1.Value) + 1;
04.      HiddenField1.Value = val.ToString();
05.      Label1.Text = val.ToString();
06.  }
```

## Viewstate

Viewstate is a very useful client side property. It is used for page level state management. Viewstate stores any type of data and used for sending and receiving information,

### Example 1 - Count.

```
01.  ViewState Demo Count: <asp:Label runat="server" id="lblcount" />
02.
03.  <asp:Button runat="server" id="Submit" onclick="Submit_Click" text="show" />
04.
05.  protected void Page_Load(object sender, EventArgs e)
```

[ASK A QUESTION](#)[CONTRIBUTE](#)

```
08. if (ViewState["count"] != null)
09. {
10.     lblcount.Text = ViewStateVal.ToString();
11.     ViewState["count"] = ViewStateVal.ToString();
12. }
13. else
14. {
15.     ViewState["count"] = "1";
16. }
17. }
18. }
19. }
20. }
21. protected void Submit_Click(object sender, EventArgs e)
22. {
23.     lblcount.Text = ViewState["count"].ToString();
```

### Example 2 - Set/Get user.

```
01. if (ViewState["UserName"] != null)
02. {
    lblName.Text = ViewState["UserName"].ToString();
```

OR,

```
01. ViewState["UserName"] = txtUserName.Text;
```

Viewstate is easy to apply and does not need access to any server resources. In a Viewstate, do not store big data, only store small values. Viewstate enables and disables on page level control. It also supports Encryption and Decryption and data/value is stored in hashed format. So we are not storing important data such as password, account information, etc. When more data is stored in this, then the page becomes heavy.

## Query String

Query string stores the value in URL.



It is visible to all the users in url as in the following link,

[ASK A QUESTION](#)[CONTRIBUTE](#)

I hope you enjoyed the clear state management concept in ASP.NET.

[ASP.NET](#)[Client side State Management](#)[Server side State Management](#)[state management](#)

Rakesh *TOP 500*

<https://www.c-sharpcorner.com/members/rakesh-chavda2>

261

2.2m

1

[View Previous Comments](#)

6

18



Type your comment here and press Enter Key (Minimum 10 characters)



Good one

[Santhakumar Munuswamy](#)

36 32.8k 901.2k

Oct 18, 2015

1

0

[Reply](#)





ASK A QUESTION

CONTRIBUTE


**80** **20.2k** **2.9m**

1 1 Reply



thanks Gowtham K

[Rakesh](#)**261** **6.9k** **2.2m**

Oct 13, 2015

0



Good work

[Gopi Chand](#)**159** **11.9k** **818.2k**

Oct 13, 2015

1 0 Reply



thanks Nilesh Jadave

[Rakesh](#)**261** **6.9k** **2.2m**

Oct 13, 2015

0 0 Reply



thanks Mukesh Kumar

[Rakesh](#)**261** **6.9k** **2.2m**

Oct 13, 2015

0 0 Reply



Thanks Harshad Pansuriya

[Rakesh](#)**261** **6.9k** **2.2m**

Oct 13, 2015

0 0 Reply



good one....

[Debendra Dash](#)**140** **13.2k** **4.2m**

Oct 13, 2015

1 1 Reply




thanks Debendra Dash

[Rakesh](#)**261** **6.9k** **2.2m**

Oct 13, 2015

0


Informative One..



CUS

C# Corner

is the Future of .NET




thanks buddy

ASK A QUESTION

CONTRIBUTE

Oct 13, 2015



Good share

Nilesh Jadav

63 23k 8.5m

Oct 13, 2015

1 0 Reply



ASK A QUESTION

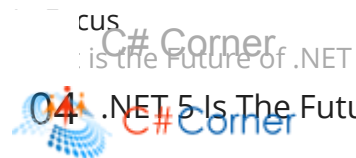
CONTRIBUTE

#### TRENDING UP

01 Angular App With ASP.NET Core And Cosmos DB

02 The Future of .NET

Implement Microsoft And Twitter Based Authentication In ASP.NET Core 2.2



04 .NET 5 Is The Future Of .NET - What Every .NET Developer Must Know

[ASK A QUESTION](#)

[CONTRIBUTE](#)

06 Performance Of Loops In C#

07 What Is .NET Core

08 What Is Scrum? Who Is Scrum Master In Agile Process?

09 Candidate Features For C# 9

10 Angular 8.0 - What's New And How To Upgrade

[View All](#) 



[ASK A QUESTION](#)

[CONTRIBUTE](#)



[ASK A QUESTION](#)

#

[CONTRIBUTE](#)

©2019 C# Corner. All contents are copyright of their authors.