

Chapter - 3

State management in ASP.NET

State क्या होता है ? (What is State)

जैसा कि हम सभी जानते हैं, ब्राउज़र आमतौर पर stateless होते हैं। अब यहां सवाल उठता है कि stateless का मतलब असल में क्या होता है?

Stateless का मतलब है, जब भी हम किसी वेबसाइट पर जाते हैं, तो हमारा ब्राउज़र हमारी अनुरोधित कार्यक्षमता (requested functionality) या अनुरोध (request) के आधार पर संबंधित सर्वर से संचार (communicate) करता है। Browser HTTP या HTTPS protocol का उपयोग करके संबंधित सर्वर से संचार (communicate) करता है।

लेकिन उस प्रतिक्रिया (response) के बाद, आगे क्या होगा? या जब हम अपना वेब ब्राउज़र (web browser) बंद करने के बाद फिर से उस वेबसाइट पर जाएंगे तब क्या होगा ?

इस स्थिति में HTTP/HTTPS को यह याद नहीं रहता कि हम किस वेबसाइट या URL पर गए थे, या दूसरे शब्दों में हम कह सकते हैं कि यह पिछली वेबसाइट की state नहीं रखता है, जिसे हमने अपने ब्राउज़र को बंद करने से पहले देखा था, जिसे stateless कहा जाता है।

अब शायद आपको समझ आ गया होगा की State और stateless का क्या मतलब होता है | तो हम यह कह सकते हैं की हमारा वेब ब्राउज़र stateless होते हैं।

State की रूपरेखा (State Outline)

जैसा कि मैंने शुरुआत में कहा, HTTP एक stateless प्रोटोकॉल है। तो हम कह सकते हैं कि यह पहले प्रयोग किये गए सभी अनुरोध (request) को सर्विस (service) देने वाले सभी संसाधनों/संदर्भों (resources/references) को हटा देता है।

ये संसाधन (resources) कई प्रकार के हो सकते हैं, जैसे-

- ऑब्जेक्ट्स (Objects)
- आवंटित मेमोरी (Allocated Memory)

- Sessions ID's
- कुछ URL जानकारी (some URL info)

State Management के प्रकार (Types of State Management)

ASP.NET में निम्नलिखित 2 State Management पद्धतियाँ हैं:



क्लाइंट-साइड स्टेट मैनेजमेंट (Client-Side State Management)

जब भी हम क्लाइंट-साइड स्टेट मैनेजमेंट (Client-Side State Management) का उपयोग करते हैं, तो state से संबंधित जानकारी सीधे क्लाइंट-साइड पर स्टोर हो जाती है। यह जानकारी यूजर द्वारा किये गए प्रत्येक अनुरोध (request) के साथ कम्यूनिकेट करेगी और फिर सर्वर साइड से कम्यूनिकेशन करने के बाद यूजर को प्रतिक्रिया (response) प्रदान करेगी |

इसका आर्किटेक्चर कुछ इस प्रकार है:-



सर्वर-साइड स्टेट मैनेजमेंट (Server-Side State Management)

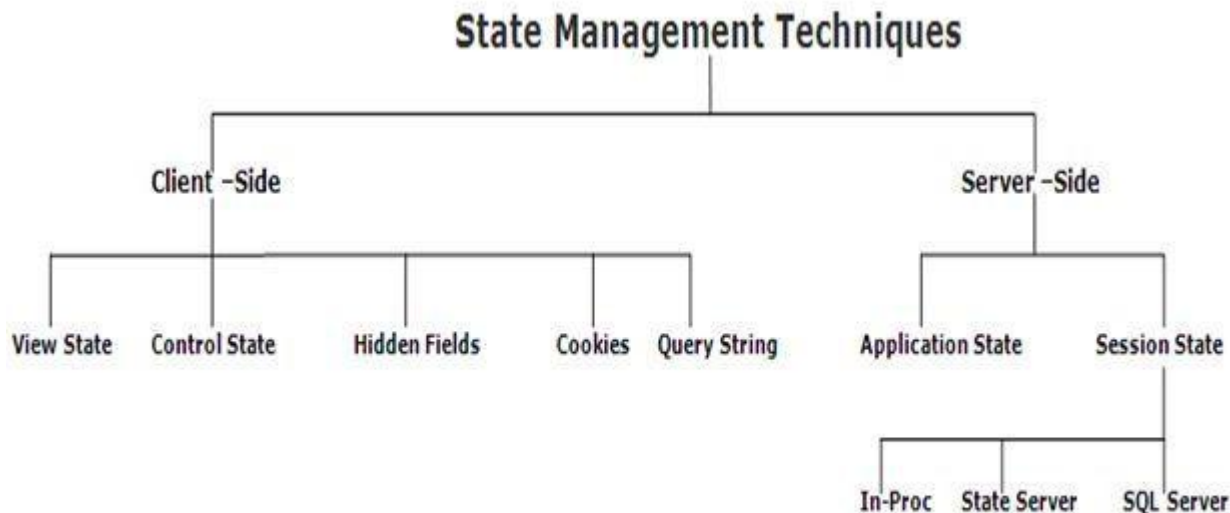
सर्वर-साइड स्टेट मैनेजमेंट क्लाइंट-साइड स्टेट मैनेजमेंट से अलग है लेकिन संचालन (operations) और कार्यप्रणाली (working) कुछ हद तक कार्यक्षमता (functionality) में समान है। सर्वर-साइड स्टेट मैनेजमेंट में सारी जानकारी Web Server की मेमोरी में संग्रहीत होती है। इस विशेष कार्यक्षमता (functionality) के कारण क्लाइंट-साइड स्टेट मैनेजमेंट की तुलना में सर्वर साइड पर अधिक सुरक्षित डोमेन होते हैं।

स्टेट मैनेजमेंट परिदृश्य (State Management Scenario)

इस बात का सीधे पता लगा पाना थोड़ा मुश्किल है कि हमारी एप्लीकेशन के लिए क्लाइंट-साइड और सर्वर-साइड आर्किटेक्चर में से कोन प्रयोग करना बेहतर होगा। हम सीधे तौर पर यह नहीं कह सकते कि हम स्टेट मैनेजमेंट के क्लाइंट-साइड या सर्वर-साइड में से किस आर्किटेक्चर का उपयोग करेंगे।

स्टेट मैनेजमेंट तकनीक (State Management Techniques)

स्टेट मैनेजमेंट तकनीक (State Management Techniques) क्लाइंट साइड और सर्वर साइड पर आधारित होती है। उनकी कार्यक्षमता (functionality) स्टेट में परिवर्तन के आधार पर अलग-अलग होती है, इसलिए यहाँ hierarchy दी गई है



क्लाइंट-साइड तकनीक (Client-side Techniques)

क्लाइंट-साइड स्टेट मैनेजमेंट तकनीकें निम्नलिखित हैं,

- View State
- Hidden field
- Cookies
- Control State
- Query Strings

सर्वर-साइड तकनीक (Server-side Technique)

सर्वर-साइड स्टेट मैनेजमेंट तकनीकें निम्नलिखित हैं,

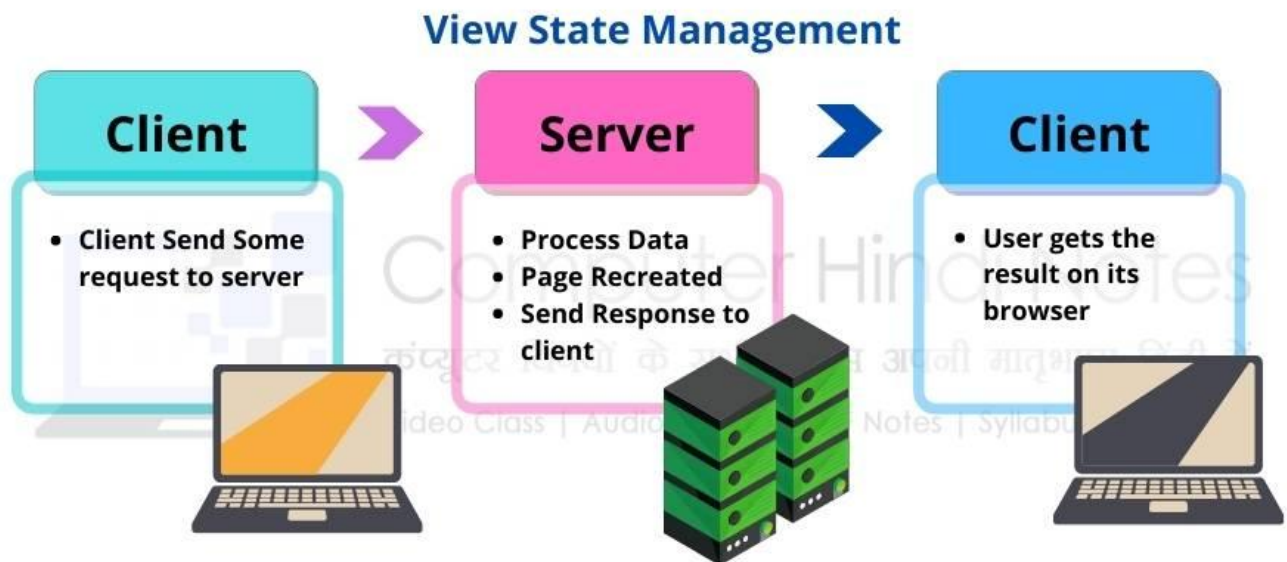
- Session State
- Application State

अब हम जानेंगे की प्रत्येक तकनीक क्या होती है और उन्हें उदाहरण सहित समझने की कोशिश करेंगे।

View State

सामान्य तौर पर हम कह सकते हैं कि **View State** उपयोग ASP.NET में उपयोगकर्ता का डेटा संग्रहीत करने के लिए किया जाता है, कभी-कभी ASP.NET एप्लिकेशन्स (ASP.NET applications) में उपयोगकर्ता [post-back](#) के बाद अस्थायी रूप से अपने डेटा को बनाए रखना या संग्रहीत करना चाहता है, इस कार्य के लिए VIEW STATE का सबसे अधिक उपयोग किया जाता है।

यह property डिफॉल्ट रूप से enable होती है, लेकिन यूजर अपनी कार्यक्षमता (functionality) के आधार पर इसमें परिवर्तन कर सकता है, हमें बस इतना करना है कि ViewState value को enable करने के लिए TRUE और इसके disable करने के लिए FALSE ऑप्शन का उपयोग करना होगा।



ComputerHindiNotes.com

```
// Page Load Event

protected void Page_Load(object sender, EventArgs e)

{

    if (IsPostBack)

    {

        if (ViewState["count"] != null)

        {

            int ViewstateVal = Convert.ToInt32(ViewState["count"]) +

1;

            View.Text = ViewstateVal.ToString();

            ViewState["count"]=ViewstateVal.ToString();

        }

        else

        {

            ViewState["count"] = "1";

        }

    }

}

// Click Event

protected void Submit(object sender, EventArgs e)

{
```

```
View.Text=ViewState["count"].ToString();
```

```
}
```

View State की कुछ विशेषताएं निम्नानुसार हैं:-

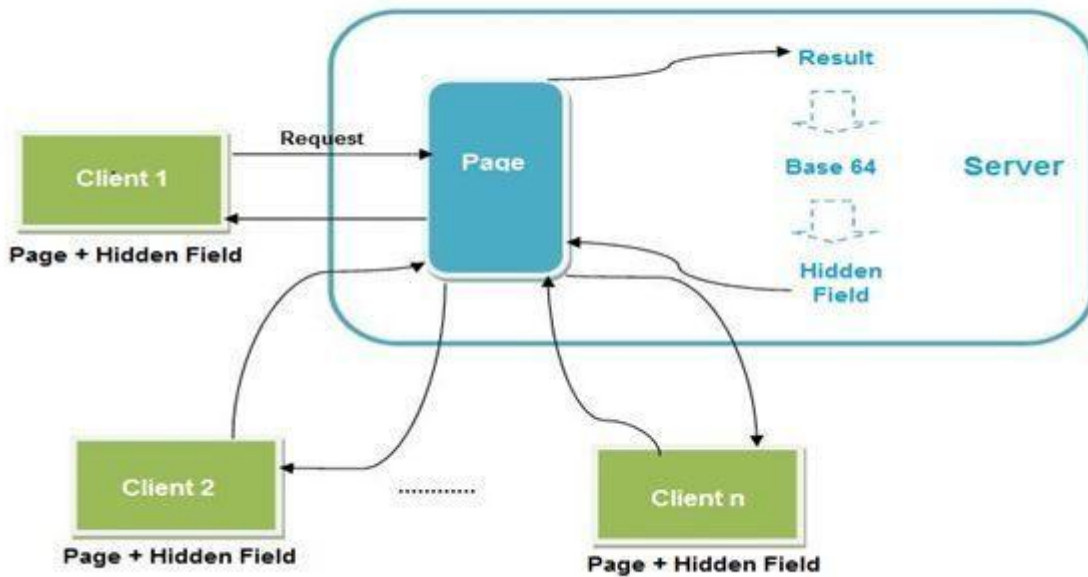
- यह page-level State Management है।
- view state का उपयोग अस्थायी रूप से डेटा रखने के लिए किया जाता है।
- इसमें किसी भी प्रकार का डेटा स्टोर किया जा सकता है।
- view state, Property पर निर्भर है।

Hidden Field

क्लाइंट साइड पर कम मात्रा में डेटा संग्रहीत करने के लिए Hidden Field का उपयोग किया जाता है। सरल शब्दों में यह केवल कुछ objects का एक कंटेनर है, लेकिन इसका जो भी परिणाम (result) होता है उसे वेब ब्राउज़र पर प्रस्तुत नहीं किया जा सकता। इसका परिणाम (result) ब्राउज़र में अदृश्य होता है।

यह सिंगल वेरिएबल (single variable) के लिए एक वैल्यू स्टोर करता है और यह एक अच्छा तरीका है जब एक वेरिएबल की वैल्यू बार-बार बदली जा सकती है और हमें अपनी एप्लिकेशन या वेब प्रोग्राम में बार बार इसको track रखने की आवश्यकता नहीं होती।

उदाहरण के लिए मान लीजिये आपने कोई वेब पेज बनाया है जिसमें एक फॉर्म है, इस फॉर्म के साथ साथ आप एक ऐसी जानकारी अगले पेज में भेजना चाहते हैं जो आप यूजर को तो नहीं दिखाना चाहते पर यह जानकारी फॉर्म के साथ ही भेजना जरूरी है तो इस समय हम Hidden Field का प्रयोग कर सकते हैं।



```
int newVal = Convert.ToInt32(HiddenField1.Value) + 1;
```

```
HiddenField1.Value = newVal.ToString();
```

```
Label2.Text = HiddenField1.Value;
```

ध्यान देने योग्य बातें

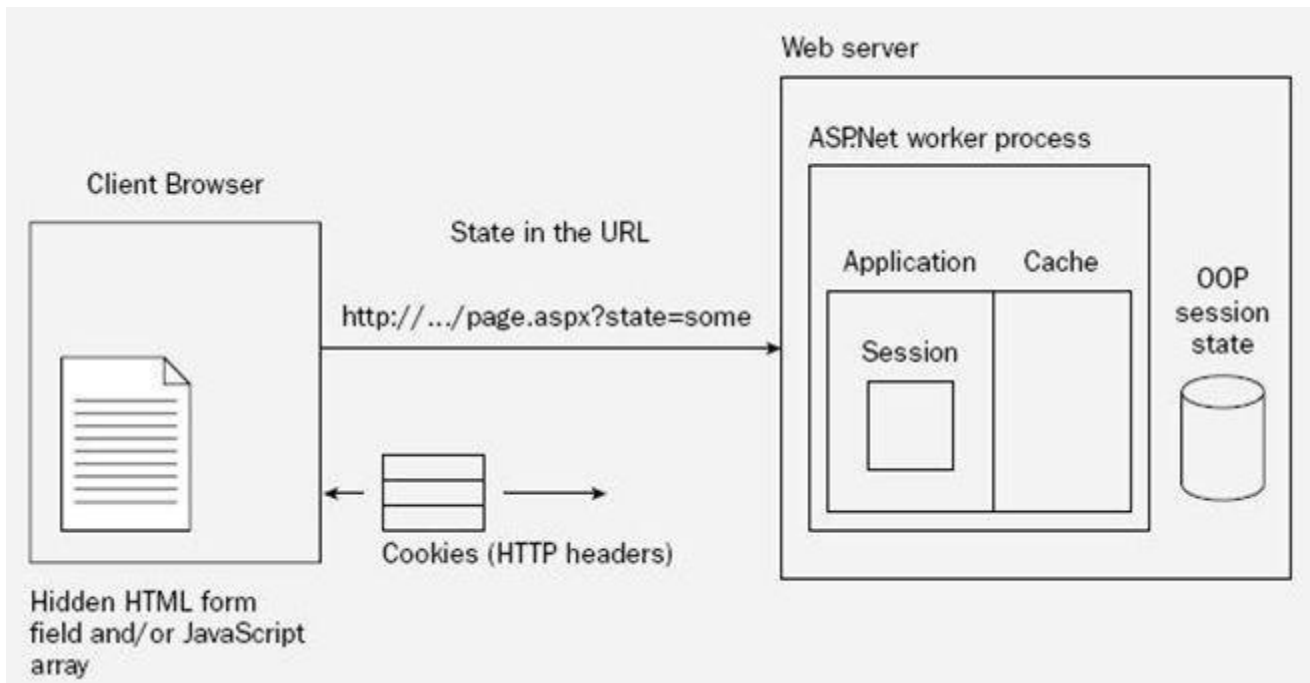
Hidden fields की कुछ विशेषताएं निम्नलिखित हैं:

- Hidden field में बहुत थोड़ी मात्रा में मेमोरी होती है
- इसकी कार्यक्षमता(functionality) को सीधे ही एक्सेस किया जा सकता है।

कुकीज़ (Cookies)

कुकीज़ का एक सेट एक छोटी टेक्स्ट फ़ाइल होती है जो क्लाइंट के ब्राउज़र का उपयोग करके यूजर की हार्ड ड्राइव में सेव (save) होती है। कुकीज़ का उपयोग केवल यूजर की पहचान मिलान (identity matching) के लिए किया जाता है क्योंकि यह केवल सत्र आईडी (sessions id), लगातार (frequent) नेविगेशन और पोस्ट-बैक अनुरोध (post-back request objects) ऑब्जेक्ट जैसी जानकारी को संग्रहीत करता है।

जब भी हम किसी विशेष कार्य के लिए इंटरनेट से जुड़ते हैं, तो यूजर की पहचान करने के लिए कुकी फ़ाइल को हमारे ब्राउज़र के माध्यम से हमारी हार्ड ड्राइव से एक्सेस किया जाता है।



```

int postbacks = 0;

if (Request.Cookies["number"] != null)
{
    postbacks = Convert.ToInt32(Request.Cookies["number"].Value) + 1;
}

// Generating Response

else
{
    postbacks = 1;
}

Response.Cookies["number"].Value = postbacks.ToString();

Result.Text = Response.Cookies["number"].Value;

```

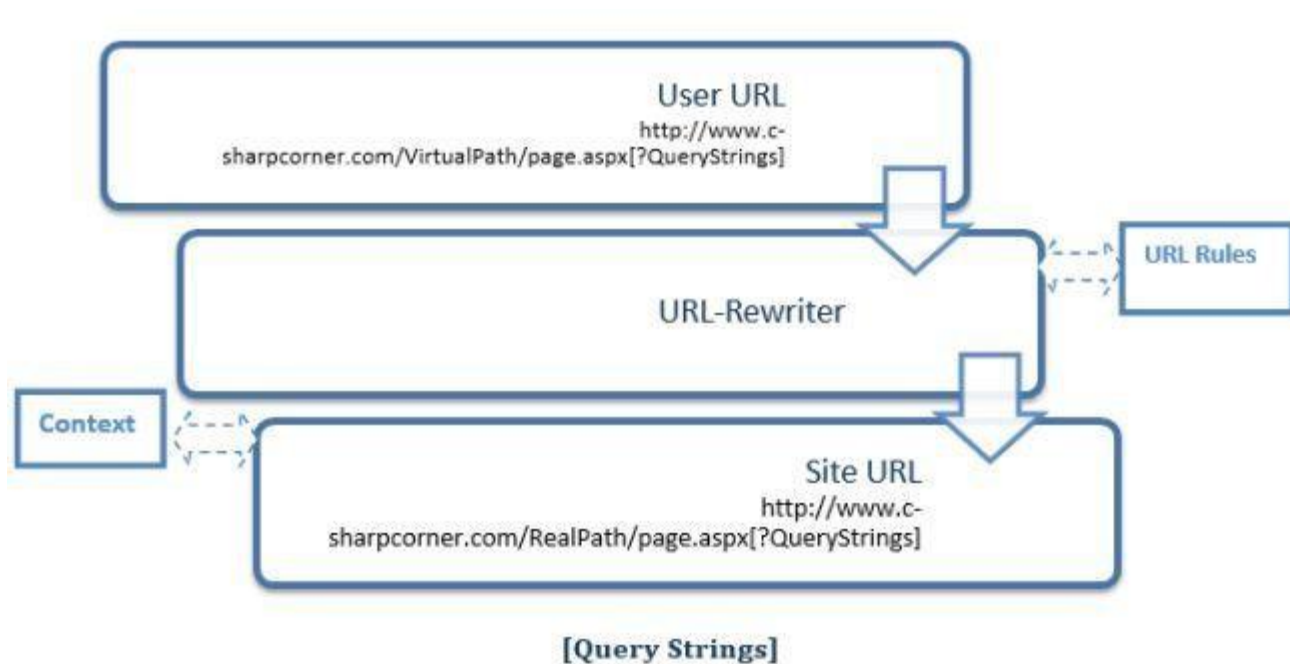

ध्यान रखने योग्य बातें

कुकीज़ (cookies) की कुछ विशेषताएं निम्नलिखित हैं:

- यह जानकारी को अस्थायी रूप से स्टोर करती है
- यह एक साधारण छोटे आकार की टेक्स्ट फ़ाइल है
- आवश्यकताओं के आधार पर इसे बदला जा सकता है
- इसे यूजर द्वारा पसंद किया जाता है।
- कुकीज़ बनाने के लिए केवल कुछ बाइट्स या KB स्थान की आवश्यकता होती है।

Query String

Query string का उपयोग किसी विशेष उद्देश्य को पूरा करने के लिए किया जाता है। आमतौर पर इनका उपयोग किसी पेज पर कुछ मान रखने और इन मानों को दूसरे पेजों पर ले जाने के लिए किया जाता है। इसमें संग्रहीत जानकारी को एक पेज से दूसरे पेज पर या उसी पेज पर भी आसानी से नेविगेट किया जा सकता है।



```
if (Request.QueryString["number"] != null)
{
View.Text = Request.QueryString["number"];
}
```

```
// Setting query string

int postbacks = 0;

if (Request.QueryString["number"] != null)

{

postbacks = Convert.ToInt32(Request.QueryString["number"]) + 1;

}

else

{

postbacks = 1;

}

Response.Redirect("default.aspx?number=" + postbacks);
```

ध्यान रखने योग्य बातें:

इसकी कुछ अन्य विशेषताएं निम्नलिखित हैं,

- इसका आमतौर पर वैल्यू को hold करने के लिए प्रयोग किया जाता है
- यह अस्थायी रूप से कार्य करती है
- जानकारी को एक से दूसरे पेज पर स्विच करती है
- इससे प्रदर्शन (performance) में वृद्धि होती है
- URL रूटिंग (URL routing) के लिए वास्तविक (real) और आभासी (virtual) path values का उपयोग करता है।