

State Yönetimi

State Yönetimi

- Web Page sınıflarının nesneleri, sayfa sunucu tarafına her defasında gönderildiğinde oluşturulur. Böyle bir durum sayfada kullanıcının girmiş olduğu ve sayfa ile ilişkili olan tüm verilerin her seferinde kaybolmasına neden olur ([https://msdn.microsoft.com/en-us/library/75x4ha6s\(v=vs.140\).aspx](https://msdn.microsoft.com/en-us/library/75x4ha6s(v=vs.140).aspx)).
- Bir web sayfası ile sunucu arasındaki etkileşim ;

Sunucu (Server)

İstemci (Client)

page1_aspx instance #1

_name.Value	<input type="text"/>
_personality.Value	<input type="text"/>

renders

HTTP/1.1 200 OK

... <html> ...

<input id="_name" type="text" />

response returned

page1_aspx instance #2

_name.Value	<input type="text" value="Sam"/>
_personality.Value	<input type="text" value="introverted"/>

renders

HTTP/1.1 200 OK

... <html> ...

<input id="_name" type="text"
value="Sam" />

response returned

submit request GET /test/page1.aspx HTTP/1.1

Enter name:
Personality:

client interaction

Enter name:
Personality:

submit pressed

submit request POST /test/page1.aspx HTTP/1.1
...
_name=Sam&_personality=introverted

Enter name:
Personality:

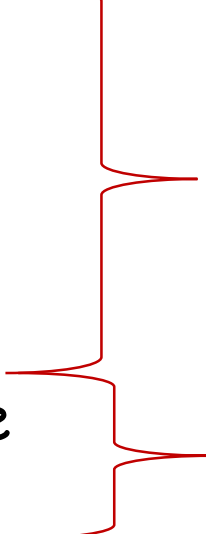
Hi Sam, you selected introverted

Etkiler

Sunucu kontrolleri aynı sayfaya olan POST istekleri sırasında durumlarını (states) muhafaza edebilirler.

- Böylelikle POST-back istekleri sırasında kontrolcülere yeniden başlangıç değerleri verilmek zorunda kalınmaz
- Kendisinden farklı bir sayfaya POST istediği (cross-page POST) yapıldığında durumu korumak için farklı teknikler kullanılması gerekir.

Sayfaya ait durumların (states) korunmasına ilişkin aşağıdaki yaklaşımlar geliştirilmiştir:

- View State
 - Control State
 - Hidden Fields
 - Cookies
 - Query Strings
 - Application State
 - Session State
- 
- İstemci Temelli
(Client-Based)
- Sunucu Temelli
(Server-Based)

İstemci Temelli State Yönetimi

I) View State

- ViewState özelliği dictionary nesnesini barındırır. Böylelikle aynı sayfaya birden fazla kez yapılan isteklerde değerleri saklayabilir.
- Bu, sayfa ve kontrol özelliklerinin korunmasında kullanılan varsayılan yöntemdir.

- İstemcide, sayfanın o anki durumu ve kontroller (page and control states) string şeklinde hashlenir ve sayfada gizli bir alan olarak (hidden field) saklanır.
- Sayfa sunucu tarafına "post-back" yapıldığında "page initialization" safasında ViewState stringi ile gelen data parse edilir ve sayfa içerisinde yer alan kontrollere elde edilen data aktarılır.
- Bknz:

[https://msdn.microsoft.com/en-us/library/75x4ha6s\(v=vs.140\).aspx](https://msdn.microsoft.com/en-us/library/75x4ha6s(v=vs.140).aspx)


```
<%@ Page Language="C#" %>
<html>
<script runat="server">
protected void Page_Load(object src, EventArgs e)
{
    if (IsPostBack)
    {
        int op1 = int.Parse(opInput.Value);
        int op2 = int.Parse(sumSpan.InnerText);
        sumSpan.InnerText = (op1+op2).ToString();
    }
}
</script>
<body>
    <form runat="server">
        <h2>ASP.NET accumulator page</h2>
        <input size="2" type="text" id="opInput" runat="server"/>
        Sum:<span id="sumSpan" runat="server">0</span>
        <p><input type="submit" value="Add" /></p>
    </form>
</body></html>
```

```
<html>

<body>
  <form name="_ctl0" method="post"
        action="accumulator.aspx" id="_ctl0">
    <input type="hidden" name="__VIEWSTATE"
          value="dDwtMTE3NzEWNDc2Njs7PvcRi11nMNe70yha9afq+YEvj46N" />

    <h2>ASP.NET accumulator page</h2>
    <input name="opInput" id="opInput" type="text" size="2" />
    Sum:<span id="sumSpan"></span>
    <p>
      <input type="submit" value="Add" />
    </p>
  </form>
</body>
</html>
```

- ViewState Base64 şeklinde kodlanarak sunucuya iletilir. Dolayısıyla herhangi bir kod çözücü ile rahatlıkla içeriğe ulaşılması mümkündür.

/wEPDwUKMjA3NjE4MDczNg9kFglCAw9kFglCBQ8PFgleBFRleHQFDUhpIHROZXJlIHRIc3RkZGR4XiT1zITp+tW79igP3d3KzGHtCw==



�
2076180736ddTextHi there
testdddx^$�̄��2�(����a�

ViewState & Güvenlik

- ViewState ile gönderilen veriler herhangi bir Base64 Decoder ile rahatlıkla ulaşılabilirdi için önemli veriler, ViewState üzerinden gönderilmemelidir. <https://www.youtube.com/watch?v=oQhytMju6kg>
- Eğer, mutlaka bu tarz verilerin ViewState üzerinden aktarılması zorunluluk ise aşağıda belirtilen niteliklerin önceden tanımlanması gerekir.
 - ViewStateEncryptionMode
 - EnableViewStateMac
- <http://radicaldevelopment.net/security-and-the-asp-net-view-state/>

Eğer sayfada karmaşık kontroller kullanılmışsa (DataGridView, Calender vb.) ViewState oldukça fazla yer kaplamaya başlar. Bu durumda istenirse `EnableViewState="false"` niteliği kullanılarak ViewState devre dışı bırakılabilir.

```
<%@ Page Language="C#" EnableViewState="False" %>  
<html>  
<!-- ... -->
```

```
<%@ Page Language="C#" %>  
<html>  
<body>  
  <form runat="server" ID="Form1">  
    <input type="text" id="opInput" runat="server" EnableViewState="false" />  
    <span id="span1" runat="server" EnableViewState="false" />  
    <p><input type="button" value="Enter" runat="server" /></p>  
  </form>  
</body></html>
```

- Sunucu, kontrollerdeki değişikliğin tespiti için ViewState'leri kullanmaktadır. Örn:
 - Kontrolün önceki değeri ViewState'de tutulur.
 - Ard arda gerçekleşen POST isteklerinde kontrolün o anki değeri ViewState'de cache'lenen değer ile karşılaştırılır ve buna göre herhangi bir olay gerçekleşir veya gerçekleşmez.
- ViewState, geçerli herhangi bir neden yoksa ViewState disable edilmemelidir.

EventTarget ve EventArgument Alanları

- Sunucu tarafı olayları (events) POST istekleri sırasında ele alınır.
- Olayın hangi kontrolcü tarafından gerçekleştirildiğinin bilinmesine gereksinim vardır. Örn:
 - Sayfadaki hangi drop-down list değişmiştir?
 - Sayfadaki LinkButton'a basılmış mıdır? vb.
- ASP.NET bu tarz bilgileri de sunucu tarafına sağlar.
 - Bu işlem için hidden field olan `_EVENTTARGET` ve `_EVENTARGUMENT` alanlarını kullanır.
 - `_EVENTTARGET` olaya neden olan control id'yi saklarken
 - `_EVENTARGUMENT` olayla ilgili parametreleri saklar.

File Edit Format

```
1
2
3 <!DOCTYPE html>
4
5 <html xmlns="http://www.w3.org/1999/xhtml">
6 <head><title>
7
8 </title></head>
9 <body>
10     <form method="post" action="WebForm1.aspx" id="form1">
11 <div class="aspNetHidden">
12 <input type="hidden" name="__EVENTTARGET" id="__EVENTTARGET" value="" />
13 <input type="hidden" name="__EVENTARGUMENT" id="__EVENTARGUMENT" value="" />
14 <input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"
value="PtN0CUxnVoMEIJEV5Ng2tcz16pRi13XCkK3/fUjEwcno8EG6k/UIeQkYwazFzeRUSrZQNGqRVhJ/35rU2t7guR0gzfPeTuVw
4eWC1pDugQqUII0C0lf6+
7qHQkfqaaijpTU3f3tDyPh9upE/UJyZzHC0YWEX+WEauFgnzgxZXdra80zjLSb4PLuk+/8kmcs3FvmVJEpgSA1Jxn2fJsErjFbaQSLh
myfBSk7QQ/rUoKkvQLKBG4jUGfvJGYKS0cDVG2L+263nEDQDd+v/X/0yR5jzce/HEci2Bz5B8KSxeIoQA8W2tCvZpR9mr2fArI3pIf1
6zHd90SSHzcF8R10liZ7thjtZlZJuZK1k1tNMS+qsR+VLPB5H9kyqZX4094zLkN04vnN1GfjjzJy/Y59DvSEBQxNJ4ETq01ABmFeJIA
MEsNgGY7Uh/2rbbn8zoPiZwjnSSBr6rXiykl9fMEzfRiM91ovjDH/2kKDIPcTc+TbDu4tAw0eYuvB86Hp6jlirMD6GXEiHbOfy+yfuc
+0FafRnl 351aMKR0mRC1NKMRful k2rNrSIlucvFPn5mWc+hFR7vPD1k7RNhRh30wW70v9F5TV32V84uRi0kfiI2Hk1vnr27CovChG+DF5
```


Özel Kullanım

Eğer istenirse ViewState istemcinin durumunu saklamak için programcı tarafından da kullanılabilir.

- ViewState'ler aynı sayfaya olan POST işlemlerinde durumlarını korurlar.
- ViewState'de saklanacak tipler Serializable olmalıdır
- Doğrudan Page.ViewState property'si kullanılarak erişim sağlanabilir.

```
public class PurchasePage : Page
{
    private void Page_Load(object sender, EventArgs e)
    {
        ArrayList cart = (ArrayList)ViewState["Cart"];
        if (cart == null)
        {
            cart = new ArrayList();
            ViewState["Cart"] = cart;
        }
        // Use items stored in ArrayList
    }
    //...
```

II) Control State

- Control state, ASP.NET 2.0 versiyonu ile birlikte kullanıma girmiştir.
- ViewState ile benzerlik gösterse de ViewState'den farklı olarak devre dışı bırakılamaz (disable edilemez).

- Control state, kontrolün zorunlu datalarını (Sayfa kontrolcüsünün sayfa numarası vb.) saklamak için tasarlanmıştır. Bu datalara post-back işlemi sırasında gereksinim duyulmaktadır.
- ASP.NET, kontrol state ile ilgili olan datayı ViewState'in de tutulduğu saklı alanda tutar.
- Veriye bağlı olan kontroller ControlState'i kullanmaktadırlar.
Örn: GridView, DetailsView, ListView

III) Saklı Alanlar (Hidden Fields)

- ASP.NET, HiddenField kontrolü ile kullanıcı sayfaya özel saklamak istediği bilgileri bu kontrolde depolayabilir.
- Bu alan, tarayıcıda görüntülenmez fakat herhangi bir standart herhangi bir kontrol gibi özellikleri (property) değiştirilebilir.
- [https://msdn.microsoft.com/en-us/library/75x4ha6s\(v=vs.140\).aspx](https://msdn.microsoft.com/en-us/library/75x4ha6s(v=vs.140).aspx)

Güvenlik!!!

- Eğer ki, kötü niyetli bir kullanıcı sayfada HiddenField'in olduğunu ve böylelikle bir data aktarımı olduğunu görürse buna müdahale etmek isteyebilir.
- Bu nedenle web uygulamasının çalışması için önem arz eden, önemli dataların HiddenField'de saklanılmaması gerekir.
- [https://msdn.microsoft.com/en-us/library/75x4ha6s\(v=vs.140\).aspx](https://msdn.microsoft.com/en-us/library/75x4ha6s(v=vs.140).aspx)

IV) Çerezler (Cookies)

- Çerezler, istemcinin bir oturumu sırasında dosyada ya da bellekte tutulan küçük boyutlu bilgilerdir. Sunucunun siteye özgü gönderdiği bilgilerden oluşur.
- Çerezler geçici veya kalıcı olabilir. (temporary or persistent)
(bkz: *http temelleri*)

- İstemci, request ile birlikte cookie içerisindeki bilgiyi de sunucuya gönderir.
- Sunucu gelen bilgiyi okur ve içeriğini elde eder.
- En tipik kullanımı sitenin kullanıcıyı tanımasını sağlayan bir token'ın (genellikle şifrelenmiştir) cookie içinde saklanmasıdır. Böylelikle sunucu kullanıcının "authenticated" veya "anonymous" olduğuna karar verir.


```
<div>
    Çerez bilgisini giriniz:
    <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    <br />
    <asp:Button ID="SubmitButton" runat="server" OnClick="SubmitButton_Click" Text="Submit" />
</div>
```

```
protected void SubmitButton_Click(object sender, EventArgs e)
{
    var cookie = new HttpCookie("mycookie", TextBox1.Text);
    Response.Cookies.Add(cookie);
}
```

V) Sorgu Stringi (Query String)

- Sorgu stringi sayfanın URL'sinin sonuna eklenen bir bilgidir. (bkz: http temelleri)

```
http://www.contoso.com/listwidgets.aspx?category=basic&price=100
```

- Bu string, durum bilgisini (state information) basit fakat sınırlı bir şekilde saklayabilir. Böylelikle bir sayfadan diğerine bilgi aktarımı gerçekleştirilebilir.

Sunucu Temelli State Yönetimi

VI) Application State

- Application State, web uygulamasında tüm sayfalardan erişilebilen genel depolama mekanizmasıdır.
- HttpSessionState sınıfının bir instance'dır.
- Application State, dataları key/value şeklinde saklayan bir dictionary tarzında bir veri yapısıdır.

VII) Session State

- Session State'in çalışması ApplicationState'e oldukça benzerdir.
- Farkı, o an çalışmakta olan tarayıcı penceresi ile sınırlı olmasıdır.
- Session State, dataları key/value şeklinde saklayan bir dictionary tarzında bir veri yapısıdır.
- HttpSessionState sınıfının bir instance'dır.

Özet

- Session State
 - Veri sunucu tarafında tutulur ve aynı tarayıcı içerisinde erişim mümkündür.
- Application State
 - Aynı web sunucusu üzerindeki çoklu kullanıcılar içindir.
 - Yine sunucu tarafında tutulur.
- Cookies
 - İstemci tarafında tutulur, aynı tarayıcı içerisinde erişim mümkündür.
- ViewState
 - İstemci tarafında tutulur
 - Kullanıcının verileri ViewState'in içerisinde saklanır.
 - Saklanacak olan veriler serileştirilebilir olmalıdır.
 - Cast işlemi uygulanabilir olmalıdır.