بررسی کوکی‌ها (Cookies) و جلسات (Sessions) در ASP.NET

در این مقاله با کوکی‌ها و جلسات آشنا شده و نحوه عمل و بکارگیری آنها در ASP.NET را یاد خواهید گرفت.

در این مقاله ابتدا به بررسی کوکی‌ها (Cookies) پرداخته و سپس جلسات (Sessions) را بررسی خواهیم کرد. به دلیل نزدیکی بسیار زیاد این دو مفهوم تصمیم گرفتم همه آنها را در یک مقاله جمع‌آوری کنم. اگر مفاهیم فوق را به درستی درک کنید و طرز استفاده از آنها را یاد بگیرید، می توانید به قدرت برنامه‌های کاربردی وب خود بیفزایید. اما استفاده نابجا از این مفاهیم در وب سایتهای مختلف می تواند به شدت روی عملکرد سایت تاثیر گذاشته و از مخاطبان و کارایی آن بکاهد. در نتیجه با مطالعه این مقاله در مورد این دو مفهوم، باید بتوانید تصمیم بگیرید که کدامیک برای وب سایت شما مناسب‌تر می باشد. در هر حال مثال‌هایی را در این مقاله خواهید دید که به طور حتم به شما کمک زیادی خواهند کرد. لازم به ذکر است که بیشتر سعی در انتقال مفاهیمی کرده‌ام که پایه کار نمی باشند و سعی کردم به بیان نکاتی بپردازم که راحت پیدا نمی‌شوند.

کوکی‌ها (Cookies)

استفان والتر در کتاب ASP.NET Unleashed در ابتدای بخش کوکی‌ها اینگونه می‌گوید: "پروتکل HTTP هیچ امکانی را در اختیار وب سرور قرار نمی‌دهد تا بتواند به کمک آن تشخیص دهد درخواست جدید از همان مرورگری صادر شده که در خواست قبلی را فرستاده یا از مرورگر دیگری آمده است. از این جهت به HTTP صفت ناپایداری (Stateless) را میدهند. از نقطه نظر وب سرور هر درخواستی که برای دریافت یک صفحه صادر شده است از طرف کاربری جدید ارسال شده است." این به طور قطع آن چیزی نیست که ما می خواهیم! وقتی می‌خواهیم اطلاعات کاربر را در هر صفحه به او نشان بدهیم (از قبیل شناسه کاربری، کلمه عبور، سبد خرید و...) باید بتوانیم وضعیت آن را حفظ کنیم یکی از راههای بسیار خوب در این زمینه استفاده از کوکی‌ها می‌باشد.

اولین بار Netscape کوکی‌ها را در مرورگر خود به کار برد و به تدریج کنسرسیوم وب (W3C) نیز آن را پذیرفت و امروزه اکثر مرورگرها از کوکی‌ها پشتیبانی می کنند. بر اساس مستندات اولیه Netscape، یک کوکی

نمیتواند حجمی بیشتر از ٤ کیلوبایت داشته باشد و با بستن صفحه مرورگر کوکی‌ها نیز از بین می روند. البته نگران نباشید اینها کوکی‌هایی هستند که پارامتر Expires آنها تنظیم نشده است. اما اگر این پارامتر را تنظیم کنید، کوکی‌ها باقی مانده و دائمی می شوند. اما تا کی؟ تا آن تاریخی که در خاصیت Expires تنظیم کرده‌اید. مرورگرهایی که می توانند با کوکی‌ها کار کنند دارای چند فایل ویژه می‌باشند که در ویندوز به آنها فایل‌های کوکی و در مکینتاش فایل‌های جادویی می گویند. کوکی‌ها از طریق هدرهای HTTP بین مرورگر و سرور جابجا می شوند. سرور با استفاده از هدر Set Cookie یک کوکی جدید ایجاد کرده و در درخواست‌های بعدی این کوکی به سرور فرستاده می شود.

در مقاله‌ای از سایت ASPFREE در مورد خواندن و نوشتن کوکی‌ها اینگونه نوشته شده است: "برای نوشتن کوکی یک شیئ جدید HttpCookie بسازید و مقدار یک رشته را به آن اختصاص دهید (به خاصیت Value آن) و سپس متد ()Add را در Response.Cookies فرا بخوانید. شما همچنین می توانید مقدار Expires را به یک مقدار تاریخ تغییر دهید تا زمان انقضاء برای کوکی‌تان تایین کرده باشید."

باید توجه داشته باشید که کوکی‌ها فقط مقادیر رشته‌ای را ذخیره می کنند و برای نوشتن مقادیر دیگر در کوکی‌ها باید هر آنها را به یک رشته تبدیل کنید. این کد از سایت CodeToad برای یادگیری نحوه استفاده کوکی‌ها بسیار مناسب می باشد:

```
Using System.Web;


// نوشتن
Response.Cookies["BackgroundColor"].Value = "Red";


// خواندن
Response.Write(Request.Cookies["BackgroundColor"].Value);
```

به دلایل امنیتی شما می‌توانید فقط کوکی‌هایی را بخوانید که از یک دامنه آمده باشند. همچنین ممکن

است شما نیاز به کوکی‌هایی داشته باشید که چند آیتم را در خود نگهداری کنند، یک مثال برای این کار در زیر می بینید:

```
HttpCookieCollection cookies = Request.Cookies;


for (int n = 0; n < cookies.Count; n++) {
    HttpCookie cookie = cookies[n];
    Response.Write("<hr/>Name: <b>" + cookie.Name + "</b><br />");
    Response.Write("Expiry: " + cookie.Expires + "<br />");
    Response.Write("Address1: " + cookie.Address1+ "<br />");
    Response.Write("Address2: " + cookie.Address2+ "<br />");
    Response.Write("City: " + cookie.City+ "<br />");
    Response.Write("Zip: " + cookie.Zip+ "<br />");
}
```

یک مثال درباره کوکی‌های تو در تو به زبان VB.NET:

```
If Request.Form("savecookie") = "Yes" and ValidLogin = "Yes" Then
   Response.Cookies("member")("username") = Request.Form("username")
   Response.Cookies("member")("password") = Request.Form("password")
   Response.Cookies("member").Expires = DATE + 365
End if
```

جدول زیر بعضی از خصوصیات پیشرفته کوکی‌ها را نمایش می‌دهد:

خاصیت	توضیحات
Domain	دامنه‌ای که محدوده کوکی را تعیین می کند.

Path	مسیر منتسب به کوکی.
Secure	مقدار بولینی که تعیین می کند آیا کوکی باید فقط روی یک اتصال رمز شده ارسال گردد یا نه؟
HasKeys	مقدار بولینی که تعیین می کند که آیا کوکی مربوط به یک کوکی دیکشنری است یا نه؟

بطور خلاصه، کوکی‌ها چه خوبیهایی دارند؟ اشغال فضا روی کلاینت که باعث کاهش ترافیک و اشغال فضا روی سرور می شود و اطمینان کار. کوکی‌ها چه بدیهایی دارند؟ بعضی از مرورگرها از کوکی‌ها پشتیبانی نمی کند، بعضی از کاربران کوکی‌ها را پاک می کنند یا نمی پذیرند و این که فقط داده نوع رشته‌ای را ذخیره می کنند.

جلسات (Sessions)

فریم ورک دات نت برای رد گیری حرکت کاربر ما را تنها نگذاشته و یک امکان خوب به نام Session State را در اختیار ما قرار داده است. به طور پیش فرض وقتی کاربر اولین بار صفحه‌ای را از یک وب سایت ساخته شده با ASP.NET درخواست می کند یک کوکی جلسه به نام ASP.NET_SessionID ساخته شده و به مرورگر او ارسال میشود. با این کار ASP.NET قادر به پیگیری کاربر شده و میتواند در درخواست‌های بعدی او را شناسایی کند.

بر این اساس در ASP.NET یک شیء به نام Session قرار داده شده است که میتوانید از آن برای نگهداری اطلاعات مربوط به هر کاربر استفاده کنید. برای مثال دستور زیر یک آیتم با نام MyItem ایجاد کرده و Hello را به آن نسبت میدهد:

```
Session("MyItem")="Hello!"
```

هنگام کار با Sessionها باید به نکات زیر توجه کنید:

۱. هر Session اگر کاریر مرورگر را ببندد یا ۲۰ دقیقه از سرور درخواست نکند از بین می رود.

۲. Session هر کاربر جدا از Session بقیه کاربران است.

۳. در Session بر خلاف کوکی‌ها می توان شیئ هم ذخیره کرد.

جدول زیر بعضی از خصوصیات و متدهای شیئ Session را نمایش میدهد:

خاصیت/ متد	**توضیحات**
Remove	پاک کردنSession
RemoveAll	پاک کردن تمامSession ها
SessionID	IDمنحصر به فرد جلسه فعلی را برمیگرداند.
Abandon	Sessionفعلی را خاتمه میدهد. اگر کاربر پس از دستور فوق درخواست یک صفحه جدید کند به عنوان کاربر جدید در نظر گرفته می شود.
TimeOut	تغییر مهلت پیش فرض ختم جلسه. این خصوصیت هر عددی که باشد بعد از همان قدر دقیقه اگر کاربر درخواستی به سرور نفرستد Session ختم می شود.

نکته: از طریق فایل web.config نیز می‌توان مهلت ختم جلسه را تغییر داد:

```
<configuration>
  <system.web>
    <sessionstate timeout="60" />
  </system.web>
</configuration>
```

Eventها یا وقایع جلسه‌ها دو مورد هستند: Session_Start و Session_End. که Session_Start وقتی رخ می دهد که جلسه آغاز و Session_End وقتی رخ می دهد که جلسه خاتمه پیدا کند. این Eventها را باید در فایل Global.asax تعریف کرد.

در زیر یک مثال عملی از این رویدادها را خواهید دید:

```
<html>
  <head>
    <title>SessionCount.aspx</title>
    <Script Runat="Server">
      Sub Page_Load()
        lblSessionCount.Text = Application("SessionCount")
      End Sub
    </Script>
  </head>

  <body>

  Current Sessions:
  <asp:Label ID="lblSessionCount" Runat="Server" />

  </body>
</html>
```

Default.aspx

```
<Script Runat="Server">
  Sub Session_Start()
    If Application("SessionCount") Is Nothing Then
      Application("SessionCount") = 0
    End If
```

```
    Application("SessionCount") += 1
  End Sub

  Sub Session_End()
    Application("SessionCount") -= 1
  End Sub
</Script>
```

Global.asax

بطور کلی برای نگهداری مقادیر Sessionها در ASP.NET سه روش وجود دارد: درون پروسه (In Process)، ذخیره در سرویس ویندوز و ذخیره در SQL Server.

Sessionها به طور پیش فرض در داخل پروسه مدیریت می شود و تمام آیتم‌هایی که در Sessionها می‌سازیم در همان پروسه وب سرور ذخیره می شوند. مهمترین مشکل این روش این است که اگر به هر دلیل سرور از کار بیفتد و یا Application Webما دستکاری شود، تمام داده‌ها از بین میرود و از طرف دیگر بسط پذیری را در سایت محدود می کند و نمی توان آن را به اشتراک گذاشت.

اما با استفاده از تکنیک ذخیره در پایگاه داده SQL Server می‌توان حتی در صورت از کار افتادن سرور نیز اطلاعات را حفظ کرد. تعریف اشیای ضروری در SQL Server به منظور مدیریت داده‌های جلسه با اجرای بچ فایل InstallSqlState.sql صورت می گیرد. بعد از این کار باید فایل web.config را نیز به شکل زیر تغییر داد:

```
<configuration>
  <system.web>
    <sessionstate
      mode="SqlServer"
      sqlConnectionString="Server=127.0.0.1;UID=sa;Pwd=YourPassword" />
  </system.web>
```

```
</configuration>
```