عنوان: **Cookie - قسمت اول**
نویسنده: یوسف نژاد
تاریخ: ۱۳۹۲/۰۲/۰۵ ۱۲:۳۰
آدرس: www.dotnettips.info
برچسب‌ها: Cookie, Http Cookie, Magic Cookie

مقدمه

مدت زیادی است که کوکی‌ها در عرصه وب نقش مهمی ایفا میکنند، اما متاسفانه مفهوم روشن و واضحی از آن و نحوه کار آن در اختیار اکثر کاربران و توسعه دهندگان وب نیست. شاید اولین مشکل ناشی از سوءتفاهم‌های بسیاری باشد که درباره کوکی وجود دارد. مثلا برخی آن را ابزاری صرفا برای جاسوسی از کاربران اینترنتی میدانند. برخی دیگر، از آنها و نحوه کارکردشان کلا صرفنظر میکنند. مشکل دیگری که در رابطه با کوکی‌ها میتوان برشمرد، عدم وجود رابط کاربری مناسب برای بررسی و مدیریت کوکی هاست. اما با وجود این مشکلات و برخی دیگر امروزه کوکی‌ها جزئی بسیار مهم در وب هستند که درصورت حذفشان، بسیاری از وب سایتها و برنامه‌های مبتنی بر وب از کار خواهند افتاد.

یک کوکی (cookie به معنی شیرینی یا کلوچه! که با عناوین دیگری چون Http Cookie و Web Coockie و Browser Cookie نیز شناخته میشود)، به داده‌های ارسالی از یک وب سرور (که معمولا بصورت داده‌های متنی کدگذاری شده هستند) اطلاق میشود که در مخزنی مخصوص در مرورگر کاربر به هنگام بازدید از یک سایت ذخیره میشود. وقتیکه کاربر سایت مذبور را در آینده دوباره مرور کند، این داده‌های ذخیره شده توسط مروگر به وب سرور ارسال میشود تا مثلا فعالیتهای قبلی کاربر مورد بررسی قرار گیرد. کوکی‌ها برای فراهم کردن مکانیزمی قابل اعتماد جهت ذخیره فعالیتهای قبلی یا آخرین وضعیت کاربر در یک وب‌سایت طراحی شده‌اند. با اینکه کوکی‌ها دسترسی بسیار محدودی در سمت کلاینت دارند (تقریبا هیچ دسترسی‌ای به هیچیک از منابع سیستم کاربر ندارند) اما با پیگیری هوشمند و هدفمند برخی از آنها میتوان به داده‌هایی از تاریخچه فعالیتهای کاربر در یک مرورگر و سایت خاص دست یافت که به نوعی **نقض حریم شخصی** کاربران به حساب می‌آید.

نکته: درواقع میتوان گفت که از کوکی به نوعی برای فراهم کردن "حافظه" موقت برای مرورگرها در ارتباط با وب سرورها استفاده میشود.

پروتوکل HTTP که برای تبادل داده‌ها میان مرورگر و وب سرور در بارگذاری صفحات وب استفاده میشود، پروتوکلی بدون حالت یا وضعیت (state-less) است. بدین معنی که به محض ارسال داده‌های یک صفحه وب به سمت مرورگری که آنرا درخواست کرده، وب سرور هیچ چیزی از این تبادل داده را ذخیره و نگهداری نمیکند. بنابراین در درخواست‌های دوباره و سه باره و ... بعدی، وب سرور با آنها همچون اولین درخواست برخورد کرده و رفتاری کاملا یکسان در برخورد با این درخواستها نشان خواهد و دقیقا همان داده‌ها را به سمت مرورگر ارسال خواهد کرد.

این رفتار در موارد زیادی میتواند دردسرساز باشد. مثلا وب سرور نمیتواند بفهمد که یک کاربر لاگ‌آن (LogOn یا همان SignIn) کرده و یا اینکه یکسری تنظیمات شخصی اعمال کرده است، چون جایی برای ذخیره و نگهداری این حالات یا وضعیتها در پروتوکل HTTP وجود ندارد. خوشبختانه وجود کوکی‌ها یکی از بهترین راه حل‌ها برای رفع مشکلات اشاره شده است.

بنابراین همانطور که اشاره شده یکی از مهمترین انواع کاربردهای کوکی‌ها در زمینه اعتبار سنجی کاربران است. با استفاده از این نوع کوکی وب سایتها میتوانند از وضعیت ورود یا خروج کاربران و نیز انواع دسترسی‌ها و تنظیمات آنها باخبر شوند. البته با توجه به حساسیت این موضوع، درباره نحوه ذخیره داده‌ها در این نوع کوکی‌ها باید دقت خاصی اعمال شود. اگر در این زمینه سهل انگاری هایی انجام شود، ممکن است خوراک جذابی برای هکرها فراهم شود! تبلیغات درون سایتها نیز از قسمتهایی است که استفاده بسیاری از کوکی میکند که بعضا موجب بروز خطراتی برای کاربران میشود.

تاریخچه

واژه Cookie از عبارت Magic Cookie برگرفته شده است. به طور خلاصه Magic Cookie به مجموعه‌ای از داده‌های «بدون نیاز به تغییر» میگویند که بین برنامه‌هایی که در ارتباط با یکدیگرند، ردوبدل میشود. داده‌های موجود در Magic Cookieها معمولا برای سمت دریافت کننده مفهوم خاصی ندارد و به نوعی برای ذخیره وضعیت «سمت دریافت کننده» در «برنامه ارسال کننده» و استفاده از آن در ارتباطهای بعدی کاربرد دارد. به بیان دیگر حالت یا وضعیت یا تنظیمات «برنامه مقصد در برنامه مبدأ» با استفاده

از کوکی در «خود برنامه مقصد» نگهداری میشود!
در سال 1994 آقای Lou Montulli هنگامیکه در شرکت Netscape Communications در توسعه یک برنامه تجاری تحت وب مشارکت داشت، ایده استفاده از این تکنولوژی را در ارتباطات وب ارائه داد که بعدها عنوان HTTP Coockie را بخود گرفت. برای اولین بار از این کوکی‌ها در نسخه 0.9 بتای نت اسکیپ که در همان سال ارائه شد، پشتیبانی و استفاده شد. مروگر IE هم در سال 1995 و در نسخه 2.0 آن، پشتیبانی از کوکی را آغاز کرد. آقای مانتولی پتنت (Patent) تکنولوژی کوکی را در سال 1995 ارائه داد اما ثبت نهایی آن به دلیل مشکلات و مباحث حریم شخصی کاربران تا سال 1998 طول کشید.

کوکی واقعا چیست؟
یک کوکی در واقع یک فایل متنی کوچک است که در قسمتی مشخص از کامپیوتر کلاینت که توسط مرورگر تنظیم شده است، ذخیره میشود. این فایل متنی کوچک حاوی اطلاعات زیر است:
- یک جفت داده نام-مقدار (name-value pair) که داده اصلی کوکی را نگهداری میکند.
- خاصیتی برای مشخص کردن زمان انقضای کوکی (پس از این زمان این فایل متنی کوچک از درون مرورگر حذف خواهد شد)
- خاصیتهایی برای مشخص کردن محدوده‌ها و مسیرهای قابل دسترسی کوکی
- خاصیتهایی برای تعیین نحوه تبادل داده‌های کوکی و نوع دسترسی به این داده‌ها (مثلا الزام به استفاده از پروتوکلهای امن)

انواع کوکی
بطور کلی دو نوع اصلی کوکی وجود دارد:
1. **Session cookie**

از این نوع کوکی برای نگهداری موقت داده‌ها نظیر داده‌های مربوط به وضعیت یک کاربر، تنها در زمان مرور وب سایت استفاده میشود. معمولا با بستن مرورگر (یا اتمام سشن) این کوکی‌ها ازبین میروند.
2. **Persistent cookie**

برخلاف کوکی‌های سشنی این نوع کوکی‌ها در سیستم کلاینت به صورت دائمی ذخیره میشوند. معمولا دارای یک تاریخ انقضا هستند که پس از آن از بین میروند. در طول زمان حیات این کوکی ها، مرورگرها داده‌های ذخیره شده در آنها را با توجه به تنظیمات درونشان در هر درخواست به سمت وب سرور سایت مربوطه ارسال میکنند.

با توجه به کاربردهای فراوان کوکی، دسته بندیها و انواع دیگری از کوکی را هم میتوان نام برد. مانند انواع زیر:
Secure cookie

معمولا به کوکی‌هایی که خاصیت امن (Secure Attribute) در آنها فعال است این عنوان اطلاق میشود. این نوع از کوکی‌ها تنها قابل استفاده در ارتباطهای امن (با استفاده از پروتوکل HTTPS) هستند. این خاصیت اطمینان میدهد که داده‌های موجود هنگام تبادل بین سرور و کلاینت همواره کدگذاری میشود.
HttpOnly cookie

در این کوکی‌ها خاصیت HttpOnly فعال است، که موجب میشود که از آنها تنها در ارتباطات از نوع HTTP و HTTPS بتوان استفاده کرد. در سایر روشهای دسترسی به کوکی (مثلا از طریق برنامه نویسی سمت کلاینت) نمیتوان به محتوای این نوع از کوکی‌ها دسترسی پیدا کرد.
Third-party cookie

این نوع از کوکی‌ها در مقابل کوکی‌های First party (یا شخص اول) وجود دارند. کوکی‌های شخص اول توسط وب سایت جاری تولید شده اند، یعنی نشانی دامِین این کوکیها مربوط به سایت جاری است. منظور از سایت یا دامین جاری، سایتی است که آدرس آن در نوار آدرس مرورگر نشان داده میشود. کوکی‌های Third party (یا شخص سوم) به آن دسته از کوکی‌ها میگویند که توسط دامین یا وب سایت دیگری غیر از وب سایت جاری تولید و مدیریت میشوند. مثلا کوکی‌های مربوط به سایتهای تبلیغاتی. البته در مرورگرهای مدرن این نوع از کوکی‌ها به دلیل مشکلات امنیتی و نقض حریم شخصی کاربران عموما بلاک میشوند.
Supercookie

یک سوپرکوکی به آن دسته از کوکی‌ها گفته میشود که خاصیت دامین آنها به یک پسوند خیلی کلی مثل com. تنظیم شده باشد. به دلیل مسائل امنیتی بیشتر مرورگرهای مدرن تمامی انواع این سوپرکوکی‌ها را بلاک میکنند. امروزه لیستی از این پسوندهای کلی با عنوان Public Suffix List موجود است که در مرورگرهای مدرن برای کنترل کوکی‌ها استفاده میشود.

موارد استفاده از کوکی

- مدیریت جلسات (Session Management):

از کوکی میتوان برای نگهداری داده‌های مربوط به یک کاربر در بازدید از صفحات سایت و تعامل با آنها (که ممکن است در زمانهای مختلف رخ دهد) استفاده کرد. یکی از موارد بسیار پرکاربرد در این زمینه، کوکی‌های تعیین اعتبار یک کاربر است که پس از ورود به سایت در هر درخواست توسط مرورگر به سمت سرور ارسال میشود.

مثال دیگری در مورد این کاربرد نگهداری از داده‌های سبد خرید یک کاربر است. این داده‌ها را میتوان قبل از تسویه نهایی درون یک کوکی ذخیره کرد. معمولا در تمام این موارد از یک کلید منحصر به فرد که در سمت سرور تولید شده و درون کوکی به همراه سایر اطلاعات ذخیره میشود، برای تعیین هویت کاربر استفاده میشود.

- شخصی سازی (Personalization):

یکی دیگر از موارد پرکاربرد کوکیها ذخیره تنظیمات یا داده‌های مرتبط با شخصی سازی تعامل کاربر با سایت است. مثلا میتوان تنظیمات مربوط به استایل یک سایت یا زبان انتخابی برای یک کاربر مشخص را درون کوکی در سمت کلاینت ذخیره کرد. سایتهای بزرگ معمولا از این روش برای ذخیره تنظیمات استفاده میکنند، مثل گوگل و ویکیپدیا. همچنین میتوان از کوکی برای ذخیره شناسه آخرینکاربری که در سایت لاگ‌آن کرده استفاده کرد تا در مراجعه بعدی به عنوان اولین انتخاب در صفحه ورود به سایت به کاربر نمایش داد (هرچند این کار معایب خودش را دارد).

- پیگیری یا ردیابی (Tracking):

از کوکی‌ها میتوان برای پیگیری بازدیدهای یک کاربر در یک کلاینت از یک سایت یا مسیری به خصوص از یک سایت بهره برد. بدین صورت که برای هر کاربر یک کد شناسایی منحصر به فرد تولید شده و درون کوکی مخصوص این کار ذخیره میشود. سپس برای هردرخواست میتوان مثلا نشانی صفحه موردنظر و زمان و تاریخ آن را درون یک منبع ذخیره کرد تا برای استفاده‌های آتی به کار روند.

البته کاربردها و استفاده‌های دیگری نیز برای کوکی میتوان برشمرد که بدلیل طولانی شدن بحث از آنها صرفنظر میشود.

ایجاد کوکی

همانطور که در بالا نیز اشاره شد، کوکی‌ها داده هایی هستند که توسط وب سرور برای ذخیره در کلاینت (مرورگر) تولید میشوند. مرورگرها نیز باید این داده‌ها را بدون هیچ تغییری در هر درخواست عینا به سمت سرور برگردانند. درواقع با استفاده از کوکی‌ها میتوان به ارتباط بدون حالت HTTP به نوعی خاصیتی از جنس state اضافه کرد. به غیر از خود وب سرور، برای تنظیم و یا تولید کوکی میتوان از زبان‌های برنامه نویسی سمت کلاینت (مثل جاوا اسکریپت) البته درصورت پشتیبانی توسط مرورگر نیز استفاده کرد.

در جدیدترین استانداردهای موجود (RFC 6265) درباره کوکی آورده شده که مرورگرها باید بتوانند حداقلهای زیر را پشتیبانی کنند:

- توانایی ذخیره حداقل 3000 کوکی
- توانایی ذخیره کوکیها با حجم حداقل 4 کیلوبایت
- توانایی ذخیره و نگهداری حداقل 50 کوکی به ازای هر سایت یا دامین

نکته: توانایی مرورگرهای مدرن در مدیریت کوکی‌ها ممکن است فراتر از استانداردهای اشاره شده در بالا باشد.

انتقال داده‌های صفحات وب سایتها از طریق پروتوکل HTTP انجام میشود. مرورگر برای بارگذاری صفحه موردنظر کاربر از یک وب سایت، معمولا یک متن کوتاه به وب سرور مربوطه ارسال میکند که به آن HTTP Request میگویند. مثلا برای دریافت صفحه http://www.dotnettips.info/index.html درخواستی به شکل زیر به سمت وب سرور ارسال میشود:

```
GET /index.html HTTP/1.1
Host: www.dotnettips.info
```

مثلا نمونه یک درخواست کامل خام (Raw) از صفحه اول سایت جاری در نرم افزار Fiddler بصورت زیر است:

```
GET http://www.dotnettips.info/ HTTP/1.1
```

```
Host: www.dotnettips.info
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/24.0.1312.56
Safari/537.17
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
Cookie: BlogPost-1175=NLOpR%2fgHcUGqPL8dZYv3BDDqgd4xOtiiNxHIp1rD%2bAQ%3d; BlogComment-
5002=Wl51iaIsiBnQN1UDD4p%2fHFvuoxC3b8ckbw78mAWXZOSWMpxPlLo65%2bA40%2flFVR54;
ReaderEmail=DP%2bx4TEtMT2LyhNQ5QqsArka%2fWALP5LYX8Y
```

وب سرور با ارسال محتویات صفحه موردنظر به این درخواست پاسخ میدهد که به آن HTTP Response میگویند.
در پاسخ ارسالی، وب سرور میتواند با استفاده از یک header مخصوص با نام **Set-Cookie** یک کوکی را ایجاد کند. در زیر یک نمونه از این پاسخها را مشاهده میکنید:

```
HTTP/1.1 200 OK
Content-type: text/html
Set-Cookie: cookieName=cookieValue
Set-Cookie: cookieName 2= cookieV alue2; Expires=Thr, 10-Jun-2021 10:18:14 GMT
...
```

نمونه پاسخ ارسالی خام (Raw) در نرم افزار Fiddler مربوط به درخواست صفحه اول سایت جاری بصورت زیر است:

```
HTTP/1.1 200 OK
Date: Wed, 30 Jan 2013 20:25:15 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-Compressed-By: HttpCompress
Set-Cookie: .ASPXROLES=NzZ9qIRpCWHofryYglbsQFv_SSGPn7ivo0zKFoS94gcODVdIKQAe_IBwuc-TQ-
03jGeIkZabTuxA0A3k2-
nChy7iAWw9rPMYXSkqzMkizRFkDC0k3gQTkdLqLmmeIfnL9UjfMNWO8iVkYQrSv24ecbpFDSQCH827V2kEj8k2oCm_5sKRSmFpifh4N
7kinEi0vomG1vW4Rbg9JWMhCgcvndvsFsXxpj-NiEikC1RqHpiLArIyalEMEN-cIuVtRe7uoo938u9l-
7OXb8yzXucVl4bdqPy2DXM3ddWzb3OH1jSFM6gxwJ8qRZDlSGmEEbhji7rA-
efI4aYGTKx6heWfUsY6E2k73jJLbuZ3RB4oNwRYmz8FRB0-vm1pO7rhF1JIoi1YB17ez-Ox5chNEFkPVREanHVU9DxboJ5dKgN-
2B5udUFPunnshbN8EBhixbFQOpqRiiOK4uWWaWy3rVEJYpCCDBRctKCfEyYD1URFYeajB0AXmiMUTcGeuUtwb-
XFjbQZnbylmMF3EJgG16bcc1IEkTAUv1JfKjaql0XGWJI1; path=/; HttpOnly
Cache-Control: private
Content-Type: text/html; charset=utf-8
Content-Length: 106727

<!DOCTYPE html>
<html>
...
```

وب سرور دستور Set-Cookie را تنها برای ثبت کوکی در مرورگر در پاسخ ارسالی قرار میدهد. برای آشنایی بیشتر با این هدر و ساختار آن به RFC 6265 مراجعه کنید. این دستور برای مرورگر مشخص میکند که (درصورت پشتیبانی از کوکی و فعال بودن آن در مرورگر) در درخواستهای بعدی باید این کوکی‌ها را در متن درخواست ارسالی به سمت سرور ضمیمه کند. البته اینکار با توجه به تنظیمات خاصیتهای کوکی مربوطه انجام میشود که در ادامه بحث میشود. برای ارسال کوکی به سمت وب سرور، مرورگر از هدر Cookie در درخواست خود استفاده میکند.
مثلا با توجه به مثال قبل برای درخواست صفحه http://www.dotnettips.info/index2.html مرورگر میتواند متن زیر را به سمت وب سرور ارسال میکند:

```
GET /index2.html HTTP/1.1
Host: www.example.org
Cookie: cookieName=cookieValue; cookieName2=cookieValue2
Accept: */*
```

این درخواست که برای صفحه دیگری از همان سایت قبلی است، با درخواست اول متفاوت است. با استفاده از این درخواست وب سرور میفهمد که این درخواست با درخواست قبلی مرتبط است. وب سرور در پاسخ ارسالی میتواند کوکی‌های دیگری نیز ثبت کند.

وب سرور میتواند مقدار یک کوکی ثبت شده در مروگر را با استفاده از دستوری مشابه Set-Cookie: cookieName=cookieNewValue در پاسخ ارسالی به سمت کلاینت تغییر دهد. مرورگر با دیدن این خط در پاسخ دریافتی از وب سرور مقدار کوکی را به روز رسانی میکند. البته به شرطی که سایر خواص تنظیم شده برای کوکی **عینا** یکسان باشد.

نکته: identity یا هویت کوکی با استفاده از **تمام خواص آن به جز expires یا max-age** تعیین میشود.

مقدار یک کوکی میتواند شامل هر کاراکتر قابل چاپ اسکی (از کاراکتر ! تا کاراکتر ~ یا از کد یونیکد \u0021 تا \u007E) به غیر از کاراکترهای , و ; و فضای خالی باشد (استفاده از این سه کاراکتر و سایر کاراکترهای غیر اسکی تنها با انکدینگ میسر است). نام یک کوکی نیز از این قاعده پیروی میکند، البته شامل کاراکتر = نیز نمیتواند باشد چون این کاراکتر نقش جداکننده «مقدار» از «نام» کوکی را ایفا میکند. استانداردهای کوکی که در RFC 6265 آورده شده است، محدودیتهای بیشتری نیز دارد که ممکن است توسط مروگرهای امروزی رعایت نشود!

نکته: انکدینگ کوکیها کمی بحث دارد. ازآنجاکه کوکی‌ها به صورت هدرهای پروتوکل HTTP انتقال داده میشوند، بنابراین کاراکترهای موجود در آن باید تنها از نوع ASCII باشند. اما چون ارسال کننده و دریافت کننده نهایی کوکی یک وب سرور یکسان است بنابراین وب سرور برای ذخیره کاراکترهای غیر اسکی میتواند از انکدینگ خاص خود استفاده کند. مثلا عموم وب سرورها و نیز مرورگرها از URL Encoding برای انکدینگ کوکی‌ها استفاده میکنند (^). ظاهرا در تمام مرورگرهای مدرن برای ذخیره کوکی ها، حداقل نام و مقدار کوکی به صورت جداگانه (مثلا برای ذخیره کاراکترهای نامعتبر) انکد میشود به غیر از کاراکتر تساوی (=) بین نام و مقدار کوکی.

با استفاده از زبانهای برنامه نویسی سمت کلاینت نیز میتوان کوکی‌ها را مدیریت کرد. مثلا در جاوا اسکریپت از **document.cookie** برای اینکار استفاده میشود. نحوه کاربرد و استفاده از این پراپرتی کمی غیرعادی است. مثلا دستور document.cookie = 'dummy=a11a' یک کوکی با نام dummy و با مقدار a11a ایجاد میکند! در ادامه با این دستور و نحوه کارکردن با کوکی در جاوا اسکریپت بیشتر آشنا میشویم.

نکته: برای انکد رشته‌ها در جاوا اسکریپت از دستور escape استفاده میشود. عملیات عکس آن با دستور unescape انجام میشود.

نکته: با اینکه استاندارد تعریف کوکی مشخص کرده که برای تعریف کوکی وجود عبارتی به صورت name=value اجباری است، اما ظاهرا بیشتر مرورگرها صحت تعریف کوکی و اعتبار آنرا برای پیروی از این طرح بررسی نمیکنند و بنابراین میتوان صرفا با استفاده از یک رشته بدون علامت مساوی یک کوکی را ایجاد کرد.

خواص یک کوکی

به غیر از نام و مقدار، کوکی‌ها خواص دیگری همچون دامین (domain)، مسیر (path)، تاریخ انقضا (expiration date) یا حداکثر طول عمر (maximum age)، و Secure و HttpOnly دارند که میتوانند توسط وب سرور و یا با استفاده از زبانهای برنامه نویسی کلاینتی تنظیم شوند. مرورگرها این خاصیتها را به وب سرور ارسال نمیکنند. تنها مقادیری که به سمت وب سرور برگشت داده میشوند، نام و مقدار کوکی‌هاست. مرورگرها با استفاده از خواص کوکی زمان حذف کوکی و یا کنترل دسترسی به مقدار آن با توجه به آدرس جاری مرورگر و نیز اینکه آیا اصلا کوکی را به وب سرور ارسال کنند یا نه، را تعیین میکنند. خواص کوکی در ادامه شرح داده شده است:

1. تاریخ انقضا و حداکثر طول عمر (Expires و Max-Age)

با استفاده از یکی از این دو خاصیت، تاریخی که دیگر نیازی نیست تا کوکی به سمت سرور ارسال شود، تعیین میشود. بنابراین پس از این تاریخ، ممکن است کوکی از مخزن مرورگر پاک شود. برپایه اطلاعات موجود در RFC 6265 ، در خاصیت Expires، تاریخ انقضای کوکی باید به فرمت "Wdy, DD Mon YYYY HH:MM:SS GMT" باشد. مثل Mon, 17-Mar-2014 1:00:00 GMT. همچنین خاصیت Max-Age طول عمر کوکی را برحسب **ثانیه** از لحظه دریافت توسط مرورگر مشخص میکند. به نمونه‌های زیر توجه کنید:

```
... ;Set-Cookie: cookie1=abc; Expires=Mon, 17-Mar-2014 01:00:00 GMT
... ;Set-Cookie: cookie2=123
... ;Set-Cookie: cookie3=abc; Expires=Thu, 01-Jan-1970 00:00:01 GMT
... ;Set-Cookie: cookie3=abc; max-age=31536000
.....
```

در دستور اول، cookie1 برای حذف در تاریخ مشخص شده تنظیم شده است. در خط دوم که بدون این دو خاصیت است، یک

نوع کوکی سشنی تعریف شده است. این کوکی پس از بسته شدن مرورگر (اتمام سشن) از حافظه پاک خواهد شد.

نکته: اتمام یک سشن برای کوکی‌های سشنی دقیقا به معنی بستن مرورگر (یا تب مربوطه در مروگرهای مدرن) است.

دستور سوم که تاریخ انقضای کوکی را به تاریخی در گذشته تنظیم کرده است به مرورگر اعلام میکند که باید cookie3 را پاک کند. این روش استاندارد برای حذف یک کوکی است.

نکته: استفاده از روش تنظیم یک تاریخ انقضا در گذشته برای حذف یک کوکی تنها وقتی کار خواهد که سایر خواص تعیین شده در دستور Set-Cookie با مقادیر موجود در حافظه مرورگر دقیقا یکی باشد تا هویت کوکی موردنظر به صورت منحصربه فرد تعیین شود.

در خط چهارم به مرورگر اعلام میشود که cookie4 باید دقیقا یک سال پس از لحظه دریافت کوکی، حذف شود.

نکته: خاصیت max-age در مرورگر IE8 و نسخه‌های قبل از آن پشتیبانی نمیشود.

نکته: گزینه ای که معمولا در صفحات لاگ‌آن (LogOn) یا ساین‌این (SignIn) برای ذخیره داده‌های کاربر وجود دارد (مثل «مرا به خاطر بسپار»)، مرتبط با این خاصیت از کوکی هاست. در صورت عدم انتخاب این گزینه معمولا یک کوکی سشنی (بدون خاصیت expires) ایجاد میشود. اما با انتخاب این گزینه، یک کوکی ماندگار (Persistent) با خاصیت expires برابر با تاریخی در آینده ایجاد میشود تا درصورت بسته شدن مرورگر (اتمام سشن) داده‌های کاربر پاک نشود.

نکته: تاریخ انقضای کوکی با استفاده از تاریخ کلاینت تعیین میشود. متاسفانه هیچ راه مستقیمی برای همزمانی این تاریخ با تاریخ سرور وجود ندارد.

2. دامین و مسیر (Domain و Path)

خاصیتهای دامین و مسیر کوکی، محدوده قابل دسترس کوکی را مشخص میکنند. با استفاده از این دو خاصیت مرورگر متوجه میشود که آیا کوکی را در موقعیت و آدرس جاری باید به سمت وب سرور ارسال کند یا خیر. همچنین دسترسی به کوکی‌ها در سمت کلاینت با توجه به این دو خاصیت محدود میشود. مقدار پیش فرض این دو خاصیت برابر مسیر و دامین جاری مرورگر است که اگر مقداری برای این دو خاصیت تعیین نشود، به کوکی تعلق میگیرد.

نکته: منظور از وضعیت جاری، موقعیتی است که کوکی مذبور در آن ایجاد شده است. مثلا آدرس صفحه ای که هدر Set-Cookie را ارسال کرده و یا آدرس صفحه ای که در آن با استفاده از دستوری مشابه document.cookie = 'a=b'; کوکی مربوطه ایجاد شده است. به عنوان نمونه اگر یک کوکی در صفحه جاری همین سایت ایجاد شود و خاصیتهای دامین و مسیر آن مقداردهی نشود، مقدار دامین به www.dotnettips.info و مقدار مسیر آن به post/ تنظیم خواهد شد.

نکته: مرورگر بررسی دامین کوکی را ازطریق «مقایسه از انتها» انجام میدهد. یعنی اگر مثلا دامین یک کوکی برابر dotnettips.info باشد، این کوکی در ساب دامین‌های d1.dotettips.info و یا www.dotnettips.info و از این قبیل در دسترس خواهد بود. برای کسب اطلاعات بیشتر میتوان به RFC 6265 (قسمت Domain Matching) مراجعه کرد.

نکته: بررسی مسیر کوکی برخلاف دامین آن، ازطریق «مقایسه از ابتدا» انجام میشود. یعنی آدرس صفحه جاری پس از مقدار دامین سایت باید با مقدار مشخص شده در خاصیت مسیر شروع شود. مثلا مسیر یک کوکی برابر post/ و دامین آن نیز برابر www.dotnettips.info باشد، این کوکی در آدرسهایی چون www.dotnettips.info/post/1286 و یا www.dotnettips.info/post/1 و یا www.dotnettips.info/post/test/test2 و ... در دسترس خواهد بود. برای کسب اطلاعات بیشتر میتوان به RFC 6265 (قسمت Paths and Path-Match) مراجعه کرد.

برای روشنتر شدن مطلب به هدرهای Set-Cookie زیر توجه کنید:

```
... Set-Cookie: MyCookie1=hi; Domain=d1.d2.com; Path=/employee
... /=Set-Cookie: MyCookie2=bye; Domain=.d3.com; Path
... Set-Cookie: MyCookie3=nth
.....
```

اولین دستور به مرورگر میگوید تا یک کوکی با نام MyCookie1 و مقدار hi را با دامین d1.d2.com و مسیر employee/ ثبت کند.
بنابراین مرورگر از این کوکی تنها درصورتیکه آدرس درخواست موردنظر شامل d1.d2.com/employee باشد، استفاده میکند.
دستور دوم به مرورگر میگوید تا از کوکی MyCookie2 در مسیرهای شامل d3.com. استفاده کند.
در دستور سوم دامین و مسیر با توجه به آدرس صفحه جاری تنظیم میشود. در واقع تنها مسیرهایی که شامل آدرس صفحه جاری باشند به این کوکی دسترسی دارند.

نکته: مقدار domain تنها میتواند مربوط به دامین اصلی جاری و یا زیرمجموعه‌های آن باشد. یعنی نمیتوان یک کوکی با دامین www.d1.com در صفحه‌ای با آدرس www.d2.com ایجاد کرد.

نکته: همچنین کوکی هایی که مثلا دارای دامین www.dotnettips.info هستند از آدرسی نظیر my.dotnettips.info در دسترس نیستند. کوکی‌ها تنها در دامین و ساب دامین‌های مربوط به خود قابل خواندن هستند.

نکته: اگر مقدار خاصیت domain کوکی به چیزی شبیه dotnettip.info تنظیم شود آنگاه این کوکی در آدرسهایی چون www.dotnettips.info و یا d1.dotnettips.info نیز در دسترس است.

نکته: اگر برای خاصیت path مقدار / تنظیم شود، بدین معنی است که کوکی در تمام محدوده دامین کوکی در دسترس است.

3. Secure و HttpOnly

این دوخاصیت برخلاف خواص قبلی مقداری را تنظیم نمیکنند! بلکه همانند یک flag عمل کرده که هر کدام رفتار خاصی را برای مرورگر الزام میکنند.
خاصیت Secure مرورگر را مجبور به استفاده از ارتباطات امن و کدگذاری شده (Https) برای تبادل داده‌های کوکی میکند.
درضمن طبیعی است که وب سرور دستور ثبت چنین کوکی‌هایی را خود از طریق یک ارتباط امن به مرورگر ارسال کند تا مبادا طی یک فرایند خرابکارانه داده‌های مهم درون کوکی در بین راه دزدیده نشود.

نکته: یک کوکی Secure تنها درصورتی به سمت سرور ارسال میشود که درخواست مذکور با استفاده از SSL و ازطریق پروتوکل HTTPS ایجاد شده باشد.

خاصیت HttpOnly به مرورگر اعلام میکند که استفاده از این کوکی تنها در ارتباطات از نوع پروتوکل HTTP مجاز است. بنابراین سایر روشهای دسترسی موجود (مثل document.cookie در جاوا اسکریپت) برای این نوع کوکی‌ها کار نخواهد کرد. درواقع نحوه برخورد با این نوع کوکی‌ها در سمت سرور با سایر انواع کوکی تفاوتی ندارد و تنها در سمت کلاینت و در مرورگر است که رفتاری متفاوت متوجه این کوکی‌ها میشود و اجازه دسترسی به برنامه‌های سمت کلاینت داده نمیشود.

نکته: این خاصیت ابتدا توسط مایکروسافت در نسخه IE 6 SP1 معرفی شد و بعدها بتدریج توسط سایر مرورگرها نیز پشتیبانی شد. این ویژگی همانطور که از آن برمی‌آید برای مقابله با حملات XSS پیاده سازی شده است. البته علاوه برای جلوگیری از دسترسی به این کوکی‌ها از طریق document.cookie، در مرورگرهای مدرن از دسترسی به هدر این کوکی‌ها ازطریق متدهای شی XMLHttpRequest نیز جلوگیری میشود.

نکته: امکان تنظیم این خاصیت از طریق document.cookie در جاوا اسکریپت وجود ندارد!

مدیریت کوکی‌ها در مرورگر

همانطور که قبلا اشاره شد هویت یک کوکی با استفاده تمامی خواص آن به جز expires یا max-age مشخص میشود. یعنی ترکیب name-domain-path-secure/httponly هویت یک کوکی را منحصربه‌فرد میکند. بنابراین تا زمانیکه حتی یکی از این خواص دو کوکی با هم فرق داشته باشد این دو کوکی از هم متمایز خواهند بود. دو کوکی زیر را درنظر بگیرید:

```
Set-Cookie: cookie1=value1
;Set-Cookie: cookie1=value2; domain=dotnettips.info
```

با اینکه به نظر میرسد که این دو کوکی یکسان هستند و اجرای دستور دوم موجب بازنویسی کوکی اول میشود، اما بررسی یک درخواست ارسالی از این صفحه نشان میدهد که دو کوکی مجزا با نام مشابه به سمت سرور ارسال میشود:

Cookie: cookie1=value1; cookie1=value2

حال اگر کوکی سومی به صورت زیر تعریف شود:

Set-Cookie: cookie1=value3; domain=dotnettips.info; path=/post

وضعیت از این نیز پیچیده‌تر میشود:

Cookie: cookie1=value1; cookie1=value2; cookie1=value3

بنابراین اجرای دستور زیر در همان صفحه:

Set-Cookie: cookie1= **value4**

تنها مقدار کوکی اول را تغییر خواهد داد. یعنی در درخواست ارسالی به سمت سرور خواهیم داشت:

Cookie: cookie1= **value4** ; cookie1=value2; cookie1=value3

بنابراین دقت مضاعف به این نکته که «هویت یک کوکی با استفاده از تمامی خواص آن به جز expires یا max-age تعیین میشود» مهم است. برای قسمت «به جز expire یا max-age» هم به مثال زیر توجه کنید:

Set-Cookie: cookie2=value1; max-age=1000

بنابراین خواهیم داشت:

Cookie: cookie1=value1; cookie1=value2; cookie1=value4; cookie2=value1

یک کوکی با طول عمر 1000 ثانیه تولید میکند. بنابراین با دستور زیر میتوان مقدار همین کوکی را تغییر داد:

Set-Cookie: cookie2= **value2**

پس داریم:

Cookie: cookie1=value4; cookie1=value2; cookie1=value3; cookie2= **value2**

هرچند در دستور آخر به نظر میرسد که کوکی آخر به نوع سشنی تغییر یافته است (چون خاصیت expires یا max-age ندارد) اما درواقع این چنین نیست. تنها اتفاقی که رخ داده است این است که مقدار کوکی مذبور تغییر یافته است، درحالیکه تغییری در خاصیت expires یا max-age آن رخ نداده است.

نکته: با تغییر خواص یک کوکی، میتوان آنرا از نوع سشنی به نوع ماندگار (Persistent) تغییر داد، اما عکس این عملیات ممکن **نیست** .

SubCookie

بدلیل محدودیت موجود در تعداد کوکی‌ها به ازای هر دامین، روشی برای نگهداری تعداد بیشتری تنظیمات درون همین تعداد محدود کوکی‌ها توسط توسعه گران ابداع شده است. در این روش از طرح ساده ای که نمونه ای از آن در زیر نشان داده شده است برای نگهداری داده‌های چندین کوکی درون یک کوکی استفاده میشود:

/=Set-Cookie: cookieName=cookie1=value1&cookie2=value2&cookie3=value3&cookie4=value4; path

در نمونه بالا با اینکه عملا تنها یک کوکی تعریف شده است اما درواقع داده‌های 4 کوکی مختلف درون یک کوکی آورده شده است. تنها عیب این روش این است که زحمت بیشتری برای استخراج داده‌های کوکی‌ها باید کشید. البته امروزه برخی از فریمورکها امکاناتی جهت کار با این کوکی‌ها فراهم کرده اند.

Cookie2

در ابتدای هزاه سوم! مدتی بحثی مطرح شد برای بهبود کارایی و امنیت کوکی‌ها و پیشنهادهایی مبنی بر پیاده سازی نوع جدیدی از کوکی‌ها با عنوان Cookie2 نیز ارائه شد. حتی در نسخه جدیدی از استانداردهای HTTP State Management Mechanism که در RFC 2965 آورده شده است کلا به این نوع جدید از کوکی‌ها پرداخته شده است. هرچند برخی از مرورگرها پشتیبانی از این نوع

جدید را آغاز کردند (مثل Opera) اما بعدها به دلیل عدم استقبال از آن، این نوع از کوکی‌ها منسوخ شد و حالت آن در نسخه‌های جدید استانداردها به Obsolete تغییر یافت (RFC 6265)! درهرصورت برای آشنایی با این نوع کوکی‌ها میتوان به مراجع زیر رجوع کرد: (Cookie2 (The Grinder Documentation

RFC2965

?What is the current state of the Cookie2 specification

منابع: HTTP cookie

All About Cookies

Cookies

HTTP cookies explained

The Unofficial Cookie FAQ

نظرات خوانندگان

نویسنده: saleh
تاریخ: ۱۳۹۲/۰۲/۰۶ ۰:۲۷

مطلب بسیار خوب و کاملی بود ممنون

نویسنده: میثم هوشمند
تاریخ: ۱۳۹۲/۰۲/۰۶ ۲۳:۵۶

رسما مطلب دیگری برای گفتن باقی نگذاشتید! خیلی ممنونم بابت توضیحات کامل!

نویسنده: یوسف نژاد
تاریخ: ۱۳۹۲/۰۲/۰۹ ۲۳:۴۶

البته هنوز مطالب زیادی درباره کوکی‌ها مونده که در قسمتهای بعدی به اونا پرداخته میشه.

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳۹۲/۰۲/۱۵ ۱۹:۵۱

داشتم مقالتونو میخوندم که به مشکل پروژه‌ی خودم برخوردم و مقاله‌ی شما باعث شد مشکل منم حل بشه و کلا بفهمم که کوکی دقیقا چیه :دی
خیلی ممنون :)

نویسنده: احمد
تاریخ: ۱۳۹۲/۰۵/۰۲ ۱۶:۴۸

سلام
چند تا سوال:
-1چرا expiration مربوط به کوکی رو مرورگر در مراجعات بعدی به سرور نمیفرسته؟اگه لازمش داشته باشیم چه جور باید بدستش بیاریم؟
-2اگه زمان expire مربوط به کوکی برسه کی مسئولیت نابودی cookie رو داره؟
-3مسیر دخیره سازی کوکی‌ها کجا هست؟چه زمانی که IsPersistent برابر true یا false باشه(هر چی میگردم پیداشون نمیکنم)؟
البته مورد 3 رو در استفاده formsauthentication بررسی کردم طبق همون نامی که توی تگ forms توی web.config تنظیم شده.