Share cookie between subdomain and domain

Asked 6 years, 7 months ago Active 3 months ago Viewed 277k times

401

157

I have two questions. I understand that if I specify the domain as `.mydomain.com` (with the leading dot) in the cookie that all subdomains can share a cookie.

Can `subdomain.mydomain.com` access a cookie created in `mydomain.com` (without the `www` subdomain)?

Can `mydomain.com` (without the `www` subdomain) access the cookie if created in `subdomain.mydomain.com`?

http cookies subdomain

edited Jun 13 '17 at 21:04

asked Aug 28 '13 at 15:46

adam0101

19.3k ● 17 ● 67 ● 132

3 Yes you can.. please see link below codeguru.com/csharp/csharp/cs_internet/article.php/c19417/… – Rahul Jain Sep 23 '13 at 22:08

Closely related: stackoverflow.com/questions/3089199/… – Ciro Santilli 冠状病毒审查六四事件法轮功 Nov 10 '14 at 8:17

can you please look at this question stackoverflow.com/questions/38351769/… – Jayavardhan Gange Jul 14 '16 at 4:22

1 @adam0101 What if domain and sub domain are hosted on different server ? – user3782114 Dec 9 '16 at 15:48

3 @user3782114, it doesn't matter if they are on different servers. In my case, they were not only on different servers, but each domain was load-balanced across multiple servers. One thing that did trip us up a bit was

is to have a separate machine key generated for each environment and save that in your Web.config (assuming you transform the config for each environment). – adam0101 Dec 12 '16 at 2:19

7 Answers

Active | Oldest | Votes

627

The 2 domains `mydomain.com` and `subdomain.mydomain.com` can only share cookies if the domain is explicitly named in the `Set-Cookie` header. Otherwise, the scope of the cookie is restricted to the request host. (This is referred to as a "host-only cookie". See What is a host only cookie?)

For instance, if you sent the following header from `subdomain.mydomain.com`, then the cookie won't be sent for requests to `mydomain.com`:

```
Set-Cookie: name=value
```

However if you use the following, it will be usable on both domains:

```
Set-Cookie: name=value; domain=mydomain.com
```

This cookie will be sent for *any* subdomain of mydomain.com, including nested subdomains like `subsub.subdomain.mydomain.com`.

In RFC 2109, a domain without a leading dot meant that it could not be used on subdomains, and only a leading dot (`.mydomain.com`) would allow it to be used across multiple subdomains (but not the top-level domain, so what you ask was not possible in the older spec).

However, all modern browsers respect the newer specification RFC 6265, and will ignore any leading dot, meaning you can use the cookie on subdomains as well as the top-level domain.

In summary, if you set a cookie like the second example above from `mydomain.com`, it would be accessible by `subdomain.mydomain.com`, and vice versa. This can also be used to allow `sub1.mydomain.com` and `sub2.mydomain.com` to share cookies.

See also:

- www vs no-www and cookies
- cookies test script to try it out

edited Aug 30 '19 at 7:04

answered Apr 15 '14 at 14:07

cmbuckley

29.7k ● 7 ● 60 ● 80

My question was regarding sharing cookies between `domain.com` and `subdomain.domain.com`, not between `www.domain.com` and `subdomain.domain.com`. – adam0101 Apr 16 '14 at 14:14

3 Thanks; I added a note about the significance of the dot. – cmbuckley Apr 16 '14 at 14:34

2 I don't understand why you wouldn't just put the leading "." on the domain for maximum compatibility with old and new – Alan Macdonald Dec 21 '15 at 8:56

4 @Frank, yes I know. My comment was to clarify that my question was regarding sharing cookies between a domain and a subdomain, NOT between two subdomains. – adam0101 Feb 12 '18 at 16:43

32

I'm not sure @cmbuckley answer is showing the full picture. What I read is:

> Unless the cookie's attributes indicate otherwise, the cookie is returned only to the origin server (and not, for example, to any subdomains), and it expires at the end of the current session (as defined by the user agent). User agents ignore unrecognized cookie.
>
> RFC 6265

Also

```
8.6.  Weak Integrity

   Cookies do not provide integrity guarantees for sibling domains (and
   their subdomains).  For example, consider foo.example.com and
   bar.example.com.  The foo.example.com server can set a cookie with a
   Domain attribute of "example.com" (possibly overwriting an existing
   "example.com" cookie set by bar.example.com), and the user agent will
   include that cookie in HTTP requests to bar.example.com.  In the
   worst case, bar.example.com will be unable to distinguish this cookie
   from a cookie it set itself.  The foo.example.com server might be
   able to leverage this ability to mount an attack against
   bar.example.com.
```

To me that means you can protect cookies from being read by subdomain/domain but cannot prevent writing cookies to the other domains. So somebody may rewrite your site cookies by controlling another subdomain visited by the same browser. Which might not be a big concern.

Awesome cookies test site provided by @cmbuckley /for those that missed it in his answer like me; worth scrolling up and upvoting/:

- http://scripts.cmbuckley.co.uk/cookies.php

edited Jul 26 '18 at 13:27

Velda
488 ● 1 ● 5 ● 20

answered Jun 6 '16 at 11:59

akostadinov
13.4k ● 5 ● 59 ● 70

4 That looks to agree with what I'm saying: unless you specify a `domain`, the cookie is only used for the request host. This means that `Set-Cookie: name=value` from `mydomain.com` won't be sent with requests to subdomains. Have a play with this test script too. – cmbuckley Jul 21 '16 at 10:00

Need to point out, that section 4.1.2 (first citation) is not normative... – Velda Jul 26 '18 at 12:07

thanks for the cmbuckley link. nice to test how it works quickly. – lawphotog Jan 21 '19 at 14:50

21

Here is an example using the DOM cookie API (https://developer.mozilla.org/en-US/docs/Web/API/Document/cookie), so we can see for ourselves the behavior.

If we execute the following JavaScript:

> document.cookie = "key=value"

It appears to be the same as executing:

> document.cookie = "key=value;domain=mydomain.com"

The cookie *key* becomes available (only) on the domain *mydomain.com*.

Now, if you execute the following JavaScript on mydomain.com:

> document.cookie = "key=value;domain=.mydomain.com"

The cookie *key* becomes available to *mydomain.com* as well as *subdomain.mydomain.com*.

Finally, if you were to try and execute the following on subdomain.mydomain.com:

> document.cookie = "key=value;domain=.mydomain.com"

Does the cookie *key* become available to *subdomain.mydomain.com*? I was a bit surprised that this is allowed; I had assumed it would be a security violation for a subdomain to be able to set a cookie on a parent domain.

edited Sep 26 '17 at 20:52

answered Sep 26 '17 at 19:45

llambda

2,785 ● 1 ● 17 ● 21

1 This makes me wonder if there are separate specs describing the behavior of `httponly` cookies versus the kind of cookies you are creating. – adam0101 Sep 26 '17 at 21:37

2 The docs you posted do not agree with the statements you make. The first 2 examples are **not** equivalent (a `domain` attribute causes the cookie to work on subdomains; no such attribute does not). Leading dots are ignored at best and actively blocked at worst. – cmbuckley Jan 11 '18 at 20:44

this is the best solution if you don't want to rely on host headers. I checked it and its working – Szymon Aug 6 '19 at 9:47

7

(sent in the response for requesting `subdomain.mydomain.com`)

```
Set-Cookie: name=value; Domain=mydomain.com // GOOD
```

But you **CAN'T** set a cookie from a domain on a subdomain.

(sent in the response for requesting `mydomain.com`)

```
Set-Cookie: name=value; Domain=subdomain.mydomain.com // Browser rejects cookie
```

WHY ?

According to the specifications RFC 6265 section 5.3.6 Storage Model

> If the canonicalized request-host does not **domain-match** the domain-attribute: Ignore the cookie entirely and abort these steps.

and RFC 6265 section 5.1.3 Domain Matching

> **Domain Matching**
>
> A string domain-matches a given domain string if at least one of the following conditions hold:
>
> 1. The domain string and the string are identical. (Note that both the domain string and the string will have been canonicalized to lower case at this point.)
> 2. All of the following conditions hold:
>   - The domain string is a suffix of the string.
>   - The last character of the string that is not included in the domain string is a %x2E (".") character.
>   - The string is a host name (i.e., not an IP address).

So "subdomain.mydomain.com" domain-matches "mydomain.com", but "mydomain.com" does NOT domain-match "subdomain.mydomain.com"

Check this answer also.

edited Sep 5 '19 at 10:17

answered Sep 5 '19 at 10:11

Accountant

3,856 ● 2 ● 18 ● 40

This was the most helpful answer for me. – Toby Oct 22 '19 at 4:50

In both cases yes it can, and this is the default behaviour for both IE and Edge.

script demonstrates that in (say) Chrome, the cookies are not shared between root and subdomains when no domain is specified. However the same test in IE shows that they are shared. This IE case is closer to the take-home description in CMBuckley's www-or-not-www link. I know this to be the case because we have a system that used different servicestack cookies on both the root and subdomain. It all worked fine until someone accessed it in IE and the two systems fought over whose session cookie would win until we blew up the cache.

answered Jun 6 '17 at 5:57

DannyW
81 ● 3

0

Be careful if you are working on localhost ! If you store your cookie in js like this:

```
document.cookie = "key=value;domain=localhost"
```

It might not be accessible to your subdomain, like `sub.localhost`. In order to solve this issue you need to use Virtual Host. For exemple you can configure your virtual host with `ServerName localhost.com` then you will be able to store your cookie on your domain and subdomain like this:

```
document.cookie = "key=value;domain=localhost.com"
```

edited Dec 16 '19 at 20:29

answered Dec 16 '19 at 20:20

Alexandre97122
67 ● 1 ● 6

-9

Simple solution

```
setcookie("NAME", "VALUE", time()+3600, '/', EXAMPLE.COM);
```

Setcookie's 5th parameter determines the (sub)domains that the cookie is available to. Setting it to (EXAMPLE.COM) makes it available to any subdomain (eg: SUBDOMAIN.EXAMPLE.COM)

Reference: http://php.net/manual/en/function.setcookie.php

edited Aug 8 '16 at 11:28

Brett Wolfington
5,973 ● 4 ● 28 ● 46

answered May 20 '16 at 13:09

Lawes
93 ● 3

15 This question is not PHP specific, I don't think it qualifies as valid. – sergelerator Aug 9 '16 at 21:53

1 Sergelerator, I did not pose a question. I was responding to the OP. – Lawes Sep 26 '17 at 21:01

4 @Lawes I believe sergelator means the OP's question is not PHP specific whereas your answer does seem to be a PHP-only solution, hence it wouldn't qualify to the OP's question. – Mirage Mar 15 '18 at 21:50