



# Cookies vs Sessions

R  
A  
H  
U  
L  
I  
S  
M

## Cookies vs Sessions

#webdev #javascript #programming #codenewbie



Rahul 20 فبراير 22 min read

[What The JavaScript \(27 Part Series\)](#)



31



5



71



2	Mastering the JavaScript Console
...	25 more parts...
26	<b>Cookies vs Sessions</b>
27	What is Debouncing in JavaScript?

Hey guys I also post my articles on [Hashnode](#) the amazing community and best blog platform for Devs.

The concept of Cookies and Session is very fundamental and every developer should know this.

## What is a Cookie?

A cookie is a small file with a maximum size of 4KB that the web server stores on the client computer.

Once a cookie has been set, all page requests that follow return cookie name and value. A cookie can only be read from the domain that it has been issued from.

A **cookie created by the user can only be visible to them**. Other users cannot see its value. Most web browsers have options for disabling cookies, third party cookies or both.



31



5



71



A session is a global variable stored on the server. Each session is assigned a unique id which is used to retrieve stored values.

Whenever a session is created, a cookie containing the unique session id is stored on the user's computer and returned with every request to the server.

If the client browser does not support cookies, the unique session id is displayed in the URL.

Sessions can store relatively large data compared to cookies.

The session values are automatically deleted when the browser is **closed**. If you want to store the values permanently, then you should store them in the database.

---

### **When to use cookies?**

Cookies allow us to track the state of the application using small files stored on the user's computer. The path where the cookies are stored **depends** on the browser.

### **When to use Sessions?**

To store important information such as the user id more securely on the server where malicious users cannot tamper with them.

Sessions are used to pass values from one page to another. It is also used when you want an alternative to cookies on the browser that does not support cookies.

## Cookie

- Cookies are client-side that contain user information
- Cookie ends depending on the lifetime you set for it
- The official maximum cookie size is 4kb
- A cookie is not dependent on session

## Session

- Sessions are server-side files that contain user information
- A session ends when a user closes his browser
- Within-session you can store as much data as you like. The only limits you can reach is the maximum memory a script can consume at one time, which is 128MB by default.
- A session is dependent on Cookie

---

😎Thanks For Reading | Happy Coding😓

Get weekly newsletter of amazing articles I posted this week and some offers or announcement. Subscribe from [Here](#)



Buy me a pizza



31



5



71



- 1 Window Object in JavaScript
- 2 Mastering the JavaScript Console
- ... 25 more parts...
- 26 **Cookies vs Sessions**
- 27 What is Debouncing in JavaScript?

## Discussion (3)

Subscribe

DEV

Add to the discussion



Piotr "MoroS" Mrozek • Feb 22 • Edited



"The session values are automatically deleted when the browser is closed." - no, that is not true, it's an oversimplification. Classic session management (not related to tokens, like JWT, and stateless services - which is another, more modern approach) is always based on cookies and server session working together.

What you described in that sentence is a so-called *session cookie*, which is still a cookie. Do not confuse this



31



5



71



browser (not only a tab) is closed.

You still can set your cookies to be valid for some time (24h for example), which can cause them to survive browser shutdown, but those are clearly not "session cookies" anymore, as they live longer than the current browser session. They can still contain the server session id though, giving you access to your session after you reopen your browser and revisit the site (as long as the server session can live long enough - see below).

As for the server session lifetime (in regard to statefull services), it's solely server-based, most of the time simply time-based (for example, 15 minutes from the last time a request with this session id was made). A server session can live long after the browser was closed, but, in case of "session cookies", is no longer accessible, as the cookie containing the session id was already deleted on browser close.

♡ 10    💬



Michael Grigoryan • Feb 22 • Edited



Pretty much agree on this and will also want to add the fact that many cookies that are not HttpOnly are getting hijacked using XSS attacks, so while it may be true that the backend can only access the cookies that's it bidden to, in the front end there's a whole different story than that...

♡ 4    💬



Josef Jelinek • Feb 24



In addition to other comments, session id (how you find out which client usea which session on the server) is nowadays transferred in a "http-only" and "secure" kind of cookies, URL param and non http-only cookies is

♡ 31

👏 5

🔖 71



cleanup the session even when the browser window is still open.

♡ 1    💬

[Code of Conduct](#) • [Report abuse](#)



**Rahul**

Frontend Web Developer and Blogger

Follow

**WORK**

Web Developer and Content Writer at [RAHULISM](#)

**LOCATION**

India

**EDUCATION**

Army Public School

**JOINED**

٣ نوفمبر ٢٠٢٠

**More from [Rahul](#)**

How to apply UX to your website?

[#webdev](#) [#design](#) [#ux](#) [#codenewbie](#)

♡ 31

👤 5

🔖 71



#webdev #codenewbie #css #programming

What is Debouncing in JavaScript?

#webdev #javascript #programming #codenewbie



31



5



71

