Authentication: Cookies vs Tokens

Linas Spukas Feb 16 · 3 min read

#authentication #cookies #tokens #beginners

Authentication is an exchange process, where a user is giving credentials (username, email, password etc.) for a piece of identifying information. This is identifying information should be included in HTTP requests to check if the user has permission to see content or call some actions. In this post, I will briefly go through the differences of adding identifying data into cookies versus tokens.

Cookies

A web cookie is a small portion of data, that is included in a browser's HTTP requests by default. A browser can store it and send back to the server to tell that the request is coming from the same browser, with the same identifying information to keep a user authenticated.

HTTP is a stateless protocol and cookies helps to remember important information. A while ago, when there was no storage management APIs, cookies were used to keep all client-side state. As cookies are sent with every browser request, it can have performance implications, especially when using weak mobile networks.

Cookies are included as a property in a header of any request. A server can choose to place a piece of unique information that identifies the user to that particular server. So for example, a user logs in a website, his credentials are sent to the server, where unique information, let's say ID12345 is added to the cookie and sent back to the browser. The next time a user goes the same domain (i.e. www.example.com), the browser sends a request to the server with a cookie, including the unique identifier ID12345. That will tell the server, that request came from the trusted browser, and it does not need to provide additional credentials.

Cookies are unique for each domain. That means google.com and facebook.com will each have different cookies. So if you go to facebook.com, your browser will send a request with a cookie, associated only with the Facebook domain, and it will not have access to Google's cookie. It is the default behaviour so that the hijacking of cookies would be more difficult.

Tokens

Opposed to cookies, are the idea of tokens. They were introduced as a replacement to cookies when they started to fall short as being useful.

One of the differences from cookies is a manual implementation. To use tokens, which is a set of characters and numbers, we have to specifically add them to the header every time we are making requests. So setting tokens you will have to do it yourself every time, while cookies are added automatically for you.

The advantage of tokens is that we can send them to any domain we want. If for example you are on example.com domain and want to send an authenticated request to a totally different, not a related domain, you have to use a token. Before sending a request, add a unique identifying token to the header and you will be authenticated on that domain.

This comes very handily if we build distributed systems, where servers are hosted on different domains but we want the same user to be authenticated across all of them.

Summing Up

Storing authenticated information is very dependent on your needs. While cookies are doing a great job for session management (logins, shopping carts), personalization (preferences, themes) or even tracking, and are added to request automatically, it all works for a singular domain level.

On the other hand, tokens come very useful when in need to authenticate users with cross-domain requests, but needs to be implemented manually for each of it.

Linas Spukas + FOLLOW

Full-stack web developer with a specialisation in React and NodeJS.

@spukas spukas

Add to the discussion

PREVIEW SUBMIT

▼

Babacar Cissé Dia Feb 18

great job delivering quality content very concisely!

♡ 2 REPLY

▼

AymenBz Mar 4

very clear thank you

♡ 1 REPLY

code of conduct - report abuse

Classic DEV Post from Nov 7 '19

What Are the Most Important CS Principles to Learn as a New Dev from a Non-Traditional Background?

Victoria Fluharty

❤ 138 💬 40

Another Post You Might Like

Developers mentoring other developers: practices I've seen work well

Gergely Orosz

❤ 71 💬 2

Another Post You Might Like

New Dev? Job-Hunting? Avoid These GitHub Mistakes!

Arit Amana

Enjoy content similar to this post on Arit, Developer! Subscribe to Arit's Newsletter to receive qu...

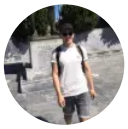
❤ 290 💬 22

Middle-Earth Ways To Learn Programming

Fyodor - Apr 4

Blogging Quick Tip - Hacking Time to Write

Rachel - Apr 5



Setting up a MySQL container in Docker

Lars Wächter - Apr 5

Twiliohackathon Project Part 1 - The Setup

Michael "lampe" Lazarski - Apr 4

Home About Privacy Policy Terms of Use Contact Code of Conduct