Browser Storage

Hunter Jorgensen

Persistence

In the last session, we looked at longer term storage on the backend. Specifically, we used MongoDB to persist user and other data for long periods of time.

However, most* modern browsers provide additional storage capacities that can persist depending on your needs.

In all of these, we can actually see these values in our dev console in the browser.

Local Storage

Local Storage was introduced in HTML5 and developers are able to access local storage through the globally available localStorage object. Local Storage works as a key/value pair: you can store a key, that is associated with another string.

Local Storage can use 5mb of data per domain and the data will persist even if you close the browser.

Demo

Session Storage

Session storage is similar to local storage, but behaves a little differently. In fact, we can simply swap out the keyword `localStorage` for `sessionStorage` in the previous example and the page will work the same. Let’s do that now!

One difference is the localStorage will persist even if you close the tab or browser, but sessionStorage will *not*.

Session storage can also store 5mb of data.

Cookies

Cookies are another type of browser side storage. Where local and session storage will persist only on the browser, cookies are included in every request that the browser makes to the server. Because of this, the maximum cookie size is only 5KB.

Cookies do not have a specific global object, so we need libraries to access them.

Cookies are typically used to maintain a ‘state’ of a user across the front and backend: for instance, keeping a user logged in. We’ll look more at how cookies look in the next section.

Indexed DBs

Lastly, modern browsers have a SQL-like database known as the indexedDB. The allow things like relationships, query, indexes, etc. and up to 50MBs of data.

Indexed DBs are typically used when you have complex data in the front end and don’t want to have the backend do numerous queries.

Demo

Indexed DB’s are pretty uncommon in web development, but useful when making small apps (like a calendar).

Concerns

There are many things to keep in mind when deciding to use browser storage. The two most important is this:

1. This data is tied specifically to the browser and is at the browser's mercy. If a user regularly clears their browser storage, or has software that prevents browser storage (as is default in some browsers) or the browser itself doesn't provide an API for this storage, it may be hard to use that resource.
2. The data that is stored in the browser is NOT secure in anyway. If any personal information (passwords, ID's, etc.) are available that is accessible to any malicious agent that has access to the browser.