

Advance Java Script

Cookies And Storages

Advance Concepts
Session Storage & LocalStorage
Basics of Cookies
Browser Debugging
Inspect element Window
Detail knowledge of different tabs in inspect element window
Caching

Web storage such as were introduced with [HTML 5](#). This made storing and retrieving data in browsers much easier, and one of the major improvements made with these in client-side storage was the storage size, which is much better than cookies.

Web storage could be accessed using Java-script, and none of this data could be read by the server unless manually passed along with the request.

There are two objects for data storage on the client provided by HTML web storage:

- **Local storage object** - Stores data with no expiration date
- **Session storage object** - Stores data for one session (data is lost when the browser tab is closed)

Local Storage

It is a web storage method that helps us store data on the client's computer in the form of key/value pairs in a web browser. The data is stored in local storage for a lifetime unless the user manually deletes it from the browser. It does not expire even when the user closes the

window or tab. Instead, the data remains in the browser until and unless the browser's memory is cleared.

It's data in the browser can only be accessed via JavaScript and HTML5. However, the user also could clear the browser data/cache to erase all local storage data. It has four methods that we can use to set, retrieve, remove and clear:

- We can use the `setItem()` method to set the data in local storage. This method takes two parameters, i.e., key and value. With this method, we can store value with a key.
`localStorage.setItem(key, value);`
- To retrieve the data stored in it, we can use the `getItem()` method. This method takes only one parameter, i.e., the key whose value we need to access.
`localStorage.getItem(key);`
- We can remove the data with the help of the `removeItem()` method, which is stored in memory about the key. `localStorage.removeItem(key);`
- The `clear()` method is used to clear all the data stored in it.

The local store has pros and cons to using local storage based on our use case.

Pros

- The data stored in it has no expiration date
- The storage limit is about 10 MB
- Its data is never transferred to the server

Cons

- Its data is plain text; hence it is not secure by design
- The data type is limited to string; hence it needs to be serialized
- Data can only be read on the client-side, not on the server-side

Session Storage

It is very similar to the local storage. Still, the main difference lies in the lifespan as it persists in the browser until its current tab is on. Once you close the tab or terminate it, the data on session storage also gets lost. We can also set and retrieve its data using `setItem()` and `getItem()` methods, respectively, similar to the local storage methods. For example:

```
session.setItem(key, value);  
  
session Storage .getItem(key);
```

CSI stands for Container Storage Interface. It is an initiative to combine the storage interface of Container Orchestrator Systems such as Mesos, Kubernetes, Docker Swarm, etc. Click to explore about, [Container Storage Interface for Kubernetes](#)

What exactly is a cookie?

The only option that was available before HTML 5 was introduced was cookies. So, storing data with it is a legacy approach to storing data on the client machine. It help us store the client-side data to enable a personalized experience for the website's users. These are sent with requests to the server and are sent to the client on response; hence its data is exchanged with the server on every request. The servers could use the cookie data to send personalized content to users.

Like web storage, it can also be created, updated, or read through JavaScript: `document.cookie`. There is an HTTP Only cookie flag available to us which can be used to restrict the cookie access in JavaScript to mitigate a few security issues such as cross-site scripting.

Cookies are categorized into two types: session cookies and persistent cookies.

Session

It do not specify the attributes such as Expires or Max-Age and hence are removed when the browser is closed.

Persistent

Persistent cookies specify the Expires or Max-Age attributes. These do not expire on closing the browser but will expire at a specific date (Expires) or length of time (Max-Age).

Which should we use: Comparison and use cases

There are many use cases of browser storage methods. The most common use cases of browser storage are:

- Personalizing site preferences
- Persisting site activities
- Storing the login state
- Saving data locally so that the website will be quicker to download or use without a network connection
- Improving website performance
- Reducing back-end server requests

The browser storage methods could be differentiated based on three main parameters - storage limit, accessibility, and expiration.

Storage Limit

Each browser storage method has a specific maximum data size. Both storage provide a large memory capacity. To be more specific, local Storage stores up to 10 megabytes and session storage stores up to 5 megabytes. On the other hand, these provide a very restrictive and small storage capacity of 4 kilobytes. So we cannot store large amounts of information in cookies.

Accessibility

From the accessibility perspective, it could be accessed in any window or tab open on the browser for a website. But if we talk about it, since session storage is tied to the particular session and each tab has its session, data is only available in the current tab in which we've set the session storage data. Lastly, cookies are somewhat similar to local storage as they are accessible from any window or tab. It could also be accessed on the server. Whenever we request the back-end server,

all the cookies are also sent along. So they are also used for tasks related to authentication.

Expiration

Its data never expires until you manually remove it, so in that sense, it could be very useful. Its data expires as soon as we close the tab because data is only available to a particular session and is equivalent to a tab. These are unique as we can manually set the expiration date for them.

	Cookies	Local storage	Session storage
Capacity	4KB	10MB	5MB
Browsers	HTML 4 / HTML 5	HTML 5	HTML 5
Accessible From	Any window	Any window	Same tab
Expiration	Manually set	Never	On tab close
Browser support	Very high	Very high	Very high
Supported data types	String only	String only	String only
Auto-expire option	Yes	No	Yes

Storage Location	Browser and server	Browser only	Browser only
Sent with requests	Yes	No	No
Editable and Blockable by users	Yes	Yes	Yes