Mobile Application Development

LECTURE 15

Need of a user-based data in browser

- Personalizing site preferences (e.g. showing a user's choice of custom widgets, color scheme, or font size). E.g Gmail preferences.
- Persisting previous site activity (e.g. storing the contents of a shopping cart from a previous session, remembering if a user was previously logged in).
- Saving JWT Token, Saving avatar, user name, user complete profile etc.

Web Storage APIs / Ionic Storage

- Before HTML5, application data had to be stored in cookies, included in every server request.
- Web Browsers provide different types of Storages inside the browser. With web storage, web applications can store data locally within the user's browser.
- Web storage is more secure, and large amounts of data can be stored locally, without affecting website performance.

Web Storage APIs

- The Web Storage API provides a very simple syntax for storing and retrieving smaller, data items consisting of a name and a corresponding value. This is useful when you just need to store some simple data, like the user's name, whether they are logged in, what color to use for the background of the screen, etc.

1. SessionStorage

2. LocalStorage

Web Storage APIs

- **SessionStorage**

sessionStorage maintains a separate storage area for each given origin that's available for the duration of the page session (as long as the browser is open, including page reloads and restores)

Stores data only for a session, meaning that the data is stored until the browser (or tab) is closed.

Data is never transferred to the server.

Storage limit is larger than a cookie (at most 5MB).

Web Storage APIs

- **localStorage**

localStorage does the same thing, but persists even when the browser is closed and reopened.

Stores data with no expiration date, and gets cleared only through JavaScript, or clearing the Browser cache / Locally Stored Data.

localStorage is not a database. It is (or should be perceived as) a persistent, string-based key-value store (kind of like NoSQL DBs). Kind of Similar to Redis.

Web Storage APIs

- **methods provided by localStorage**

localStorage.getItem(‘username’);

localStorage.setItem(‘username’, ‘Alamgir’);

localStorage.removeItem(‘username’);

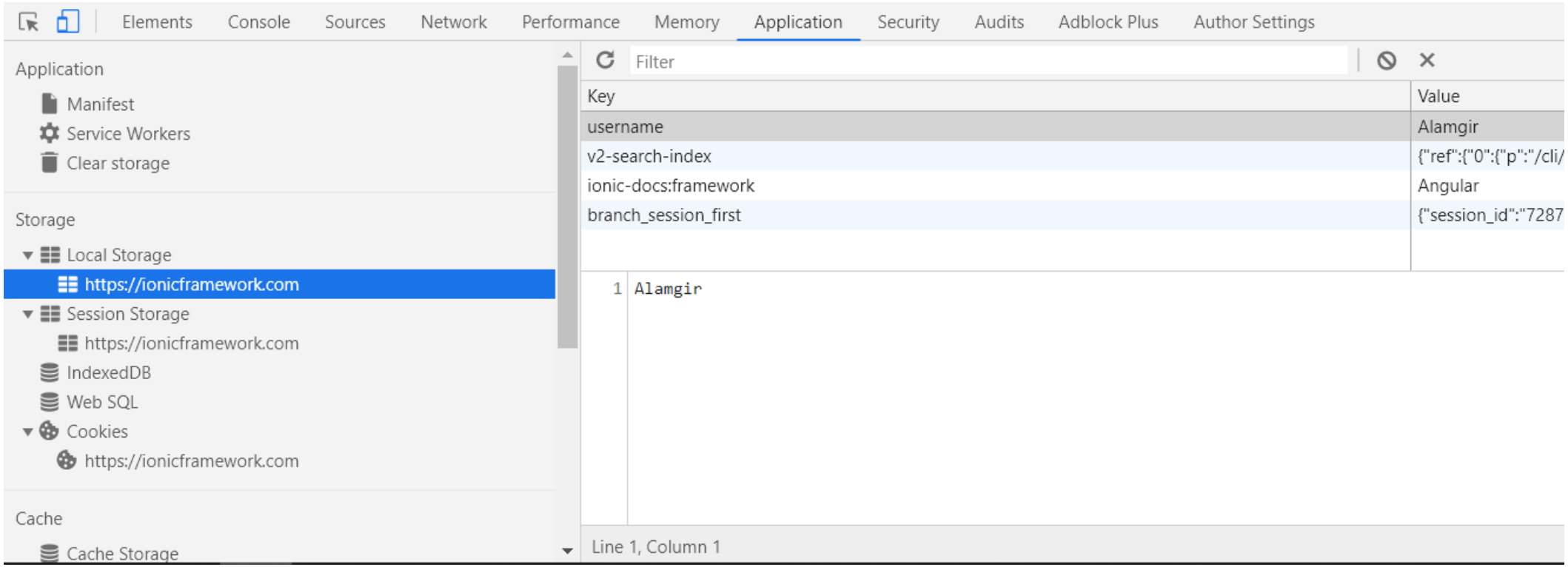
localStorage.clear();

Web Storage APIs

- You can also save entire objects/arrays in localStorage. But you need to convert them to string.

- localStorage.setItem(‘user’, JSON.stringify(user));

Web Storage APIs



What If I told you there is a way to get to where you want to be in two years rather than 5 years?

A way to achieve your 5 year goals in two years

Books

WebSQL

- SQL Based Database in the browser
- **Web SQL Database** is a web page API for storing data in databases that can be queried using a SQL. Upgraded through **IndexedDB.**
- https://www.tutorialspoint.com/html5/html5_web_sql.htm
- Web SQL database is a deprecated.

IndexedDB

- The IndexedDB API (sometimes abbreviated IDB) is a complete database system available in the browser in which you can store complex related data, the types of which aren't limited to simple values like strings or numbers. You can store videos, images, and pretty much anything else in an IndexedDB instance.

- Heavy lifting Database in the browser.

IndexedDB

- IndexedDB is more power than local storage and useful for applications that requires to store large amount of the data.
- However, this does come at a cost: IndexedDB is much more complex to use than the Web Storage API.

LocalForage (library)

- https://github.com/localForage/localForage

- A wrapper around browser databases (Wraps IndexedDB, WebSQL, or localStorage using a simple but powerful API).

PouchDB

- enables applications to store data locally while offline, then synchronize it with CouchDB and compatible servers when the application is back online,
- Real-time, synchronized, with offline support.
- Integrates superbly with CouchDB (NoSQL database).

Ionic Storage

- Ionic Storage provides an easy way to store key/value pairs and JSON objects.
- When running in a native app context, Storage will prioritize using **SQLite**, as it's one of the most stable and widely used file-based databases, and avoids some of the pitfalls of things like **localstorage** and **IndexedDB**, such as the OS deciding to clear out such data in low disk-space situations (Android OS).
- When running in the web or as a Progressive Web App, Storage will attempt to use **IndexedDB**, WebSQL, and localstorage, in that order.

Ionic Storage

Installation (https://ionicframework.com/docs/building/storage)

```
ionic cordova plugin add cordova-sqlite-storage
```

```
npm install --save @ionic/storage
```