



# CM3131 Mobile Application Design and Development

Lecture 08

Data Persistence

# Data Persistence

We often want to store data that has been generated within the app, e.g. preferences, favourite lists, scores.

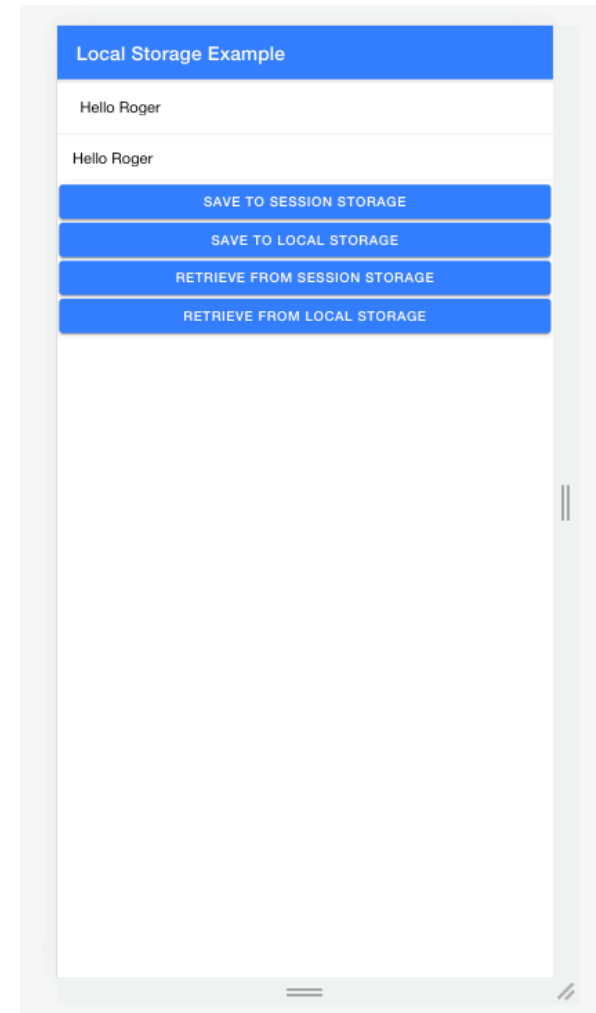
We will look at three ways of doing this. The choice of which one to use depends on what kind of data you want to store, where you want to store it and how long you want the data to persist.

Because we are creating PWAs, we can use session storage or local storage on the browser.

Alternatively, we can save to a database such as Firebase.

# Session Storage and Local Storage

In a Web App, session storage is available to store data which only persists for the duration of the application. It is also possible to use Local Storage for simple data that is required to be stored beyond the immediate session.



# Using IndexedDB

If you have more complicated storage needs, then you can use IndexedDB.

This is a client-side database that built into the browser, and is more powerful than **localStorage**.

- It stores almost any kind of data as key-value pairs, i.e. objects. It supports transactions for reliability. The values can be any JavaScript type including boolean, number, string, undefined, null, date, object, array, regex, blob, and files
- It supports key range queries, indexes.
- It can store much bigger volumes of data than **localStorage**.

Note that IndexedDB is not a relational database and it does not use SQL but it does support basic searching and sorting.

# Using IndexedDB

To start using IndexedDB, we first need to open (connect to) a database.

```
let openRequest = indexedDB.open(name, version, <callback>);
```

where **name** is the **name** of the database (string), **version** is a version number (positive integer, by default set at 1) and **callback** is an optional function parameter which can deal with updates.

The call returns an **openRequest** object with three possible outcomes:

**success:** database is ready, i.e. a database object is formed as **openRequest.result**, which we can then make use of.

**error:** opening failed.

**upgradeneeded:** database has not been set-up yet or it is ready but its version is outdated.

# Using IndexedDB

Data Persistence with IndexedDB

Application

- Manifest
- Service Workers
- Storage

Storage

- Local Storage
  - http://127.0.0.1:5500
- Session Storage
- IndexedDB
  - zoo - http://127.0.0.1:5500
    - zoo
    - animal**
- Web SQL
- Cookies
- Trust Tokens

Cache

- Cache Storage
- Back-forward Cache

Background Services

- Background Fetch
- Background Sync
- Notifications
- Payment Handler
- Periodic Background Sync
- Push Messaging

Frames

- top

Start from key

#	Key (Key path: "animal")	Primary key (Key path: "id")	Value
0	"Cat"	1	{id: 1, animal: 'Cat', so...
1	"Dog"	2	{id: 2, animal: 'Dog', so...
2	"Frog"	4	{id: 4, animal: 'Frog', sc...
3	"Mouse"	3	{id: 3, animal: 'Mouse', s...