

Front-end Development for Web & Mobile Workshop 17 Web Storage

Introduction

- When it comes to client-side development, you could be forgiven for thinking that there are a vast array of databases to choose from: Lawnchair, PouchDB, LocalForage, Dexie, Lovefield, LokiJS, AlaSQL, MakeDrive, ForerunnerDB, YDN-DB.... there are apparently lots of options.
- However all of these are based on just one of the following technologies:
 - LocalStorage
 - WebSQL
 - IndexedDB
- The important choice is not the database, but how you use it, since the way you use a database, and the ways it interacts with your JavaScript has the biggest impact on how your users perceive performance.

Introduction

- Take some time now to read about each of these:
 - LocalStorage: <https://developer.mozilla.org/en/docs/Web/API/Window/localStorage>
 - IndexedDB: https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API
 - WebSQL: https://en.wikipedia.org/wiki/Web_SQL_Database
- We are going to focus on the two more popular technologies; localStorage and IndexedDB. We are specifically going to examine why you would choose one over the other. Keep your console open for these activities and make liberal use of console.log()

localStorage

- A lightweight way to store (string) key value pairs
- Limited to 5Mb
- Good support
- Synchronous requests.... (remember that JS is single threaded)

Benefits

- Easy add fast to setup!!!

Implications

- Can not store arrays or objects without conversion.
- Synchronous operations block the DOM (which is also single threaded)

Exercise 1: Populate localStorage

Markup a simple HTML page that will use the following function on a button click.

- Include an animated gif in your page.
- Note that there is one provided with today's unit.

```
function storeLoadsOfStuff() {  
    for (var i = 0; i < 100000; i++) {  
        //dynamically construct variable name...  
        //... remember localStorage is key-value pairs  
        localStorage.setItem("A"+i, i);  
    };  
}
```

- Watch what happens to the gif when you first click the button to run the function.

Exercise 2: Clear localStorage

- For the next task you will need to be able to regularly clear localStorage.
- Write a function to do this and trigger it from a button click.
- **Hint...** search online for the **localStorage.clear()** method.

Exercise 3: Control the Number of Loop Iterations

- Rewrite the function from task number one, such that it accepts a parameter to control the number of loop iterations.
- You will need to supply the number using a HTML input element.
- Use **getElementById** to find the value of this element.
(http://www.w3schools.com/jsref/prop_text_value.asp)

Exercise 4: Browser Performance

- Experiment to see how fast your browser performs by varying the number of iterations.
- You should begin to notice that the gif slows or even stops animating at a certain threshold.

Optional Tasks: IndexedDB

- Rewrite your script to use IndexedDB instead of localStorage.
 - This will require a bit of research. Here are some hints.
 - https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API/Using_IndexedDB
- **Suggested approach:**
 - Open the DB
 - Create an object (or objects) to store your data
 - Populate the object(s)
 - Add the object(s) to the DB