# Offline web applications

We will look at Web SQL, IndexedDB, the FileSystem API, and the HTTP cache

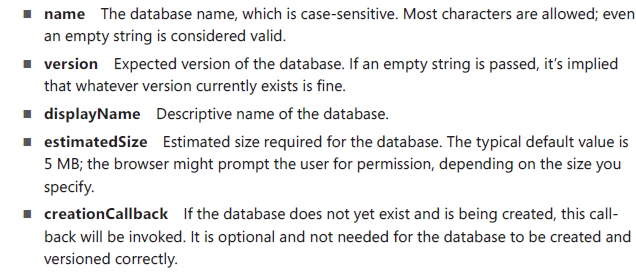
## Lesson 1: Working with Web SQL

**Serious Note**: W3C *no longer supports Web SQL*, but some browsers have continued the support of it. See http://caniuse.com/#feat=sql-storage for a full list of browsers (If you do plan on developing with Web SQL, consider the lack of browser support).

### Creating and opening the database

- Use the *openDatabase* method to start communication with a database (it returns a Database object). Note that if you attempt to open a database that doesn't exist, it will be automatically created for you.

The following are the *openDatabase* parameters:



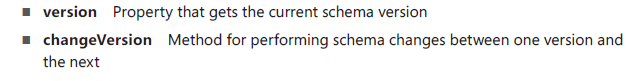
The following example creates a database named 'Library' with an estimated size of 5MB (it returns a Database object that supports transactional operations):

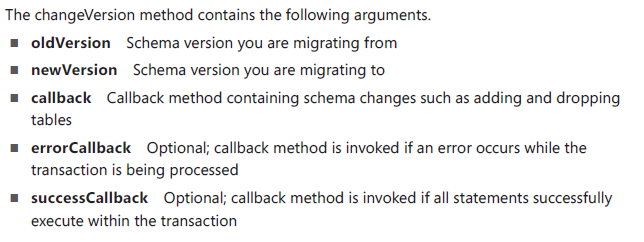
var db = openDatabase('Library', '1.0', 'My Library', 5 \* 1024 \* 1024);

**Closing a connection**: with Web SQL, closing a connection is automatically handled for you

### Performing schema updates

You may need to add new tables, drop existing ones, or even change particular columns. The *Database* object provides the following hooks.



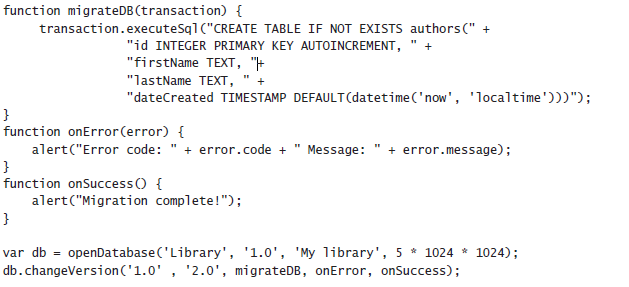


#### Adding a table

To add a table, need a callback method that accepts a *transaction* object, which executes the CREATE TABLE script.

**Note**: The *transaction* object allows multiple actions within it, and it automatically rolls back all changes if any fail.

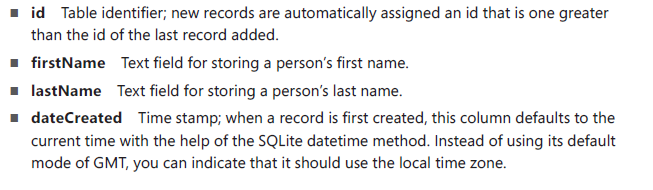
Javascript Example:



JSFiddle: https://jsfiddle.net/bs4r5ooa/ (**The code did not work on JsFiddle, or on my local computer - Forget this garbage**)

**To read the database version:** alert("Current schema: " + db.version);

**After the migration**, the current database has the following fields:



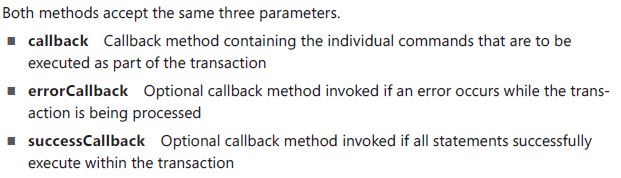
#### Using transactions

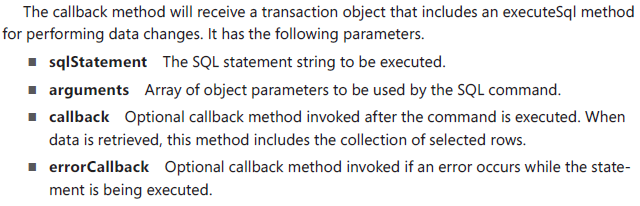
Use transactions to execute SQL statements.

The Database Object provides the ff two methods:



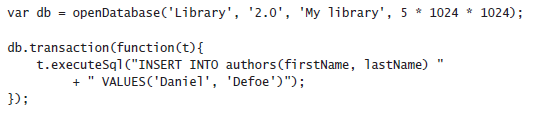




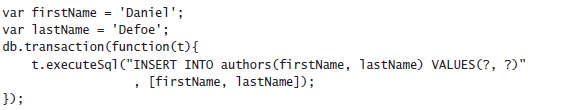


#### Inserting a new record

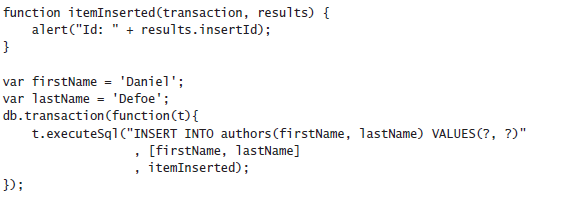
**Without using SQL parameters:**



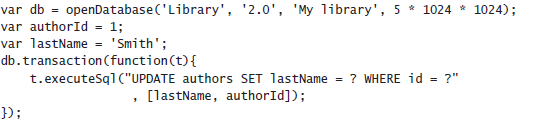
**Using SQL parameters:**

****

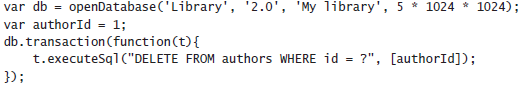
**Add a callback function to the executeSql method, which allows us capture the Id of the newly created row:**

****

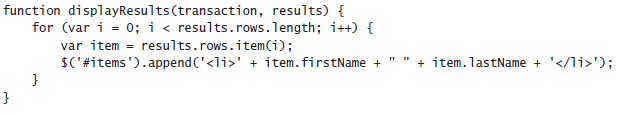
#### Updating an existing record

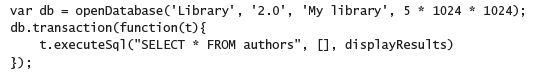


#### Deleting a record



#### Reading values from the database

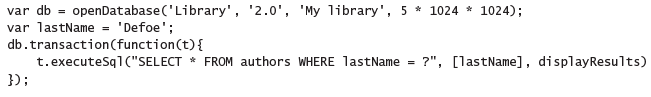




NOTE: Because you are only retrieving data, you just as easily could have used the *readTransaction* method instead of the transaction method



#### Filtering results (using WHERE)



**Using Wildcard symbol** (%)

