

# SQLite avec Cordova

Michaël MATHIEU

2 mars 2021

## Résumé

SQLite est une base de données relationnelles minimaliste qui peut facilement être utilisée sur un appareil mobile.

## 1 Bref historique

WebSQL est un projet d'une interface SQL pour navigateurs web. Cette interface n'ayant été implémentée que par SQLite, en novembre 2010, il a été décidé d'arrêter de maintenir cette interface à jour ; le but d'une interface étant d'être implémentée par plusieurs produits.

WebSQL reste notamment disponible sur Chrome et pour la WebView des applications Android. En revanche, depuis 2020, iOS a déprécié la WebView UIWebView au profit de WKWebView. Apple a décidé de ne pas conserver WebSQL pour WKWebView. De ce fait, WebSQL n'est plus une alternative possible ; il faut explicitement installer SQLite.

## 2 Installation

Un plugin permet d'utiliser SQLite avec Cordova ; la documentation complète se trouve sur <https://github.com/storesafe/cordova-sqlite-storage>

Pour installer ce plugin, il suffit d'exécuter : `cordova plugin add cordova-sqlite-storage`

## 3 Créer un schéma de base de données

Les types de données disponibles se trouvent ici : <https://www.sqlite.org/datatype3.html>

### 3.1 Connexion à la base de données

Si la base de données n'existe pas encore, la commande suivante la créera avant de s'y connecter.

```
<script type="text/javascript">
  var db = window.sqlitePlugin.openDatabase({
    name: 'mydb',
    location: 'default'
  },
  null,
  function (err) {
    alert("db error: " + JSON.stringify(err));
  });
</script>
```

Le paramètre `location`<sup>1</sup> a un impact pour iOS quant à la sauvegarde de la base de données.

Afin d'avoir le même fonctionnement sur toutes les plateformes, nous conseillons de laisser ce paramètre à `default` et de gérer vous-même la sauvegarde/restauration des données<sup>2</sup>.

---

1. <https://github.com/storesafe/cordova-sqlite-storage#opening-a-database>

2. Voir le document <https://michaelmathieu.net/resources/ownAPI.pdf>

## 3.2 Initialisation du modèle de données

Après s'être connecté à la base de données, il faut créer le modèle de données<sup>3</sup> si celui-ci n'existe pas encore (premier démarrage de l'application). Pour cela, il faut ajouter le code suivant après la connexion à la base de données.

```
db.transaction(function (tx) {
  tx.executeSql('CREATE TABLE IF NOT EXISTS LOGS (' +
    'id integer primary key autoincrement,' +
    'log text not null)');
});
```

### 3.2.1 Utilisation de la base de données

Voici un exemple d'une fonction JavaScript qui ajoute des données et une autre qui récupère les données et les affiche dans un div qui a comme nom `result`.

⚠ SQLite ne supporte pas les transactions concurrentes. Si nous ajoutons à cela le fait que la méthode `db.transaction(...)` est asynchrone<sup>4</sup>, cela signifie que si nous écrivons à la suite (ou qui s'exécute à la suite, par exemple dans une boucle `for`) plusieurs appels à cette méthode, seul le dernier appel sera exécuté correctement : les transactions précédentes étant mortes lorsque l'instruction SQL sera exécutée.

Notez l'utilisation du callback à la méthode `errorDB` en cas d'erreurs liées à la base de données pour que ces dernières ne soient pas muettes (par exemple lorsque le format d'une requête SQL n'est pas correct).

```
function insertData() {
  db.transaction(function (tx) {
    tx.executeSql('INSERT INTO LOGS (log) VALUES ("foobar")', null, null, errorDB);
    tx.executeSql('INSERT INTO LOGS (log) VALUES ("barfoo")', null, null, errorDB);
  });
}

function selectData() {
  db.readTransaction(function (tx) {
    tx.executeSql('SELECT * FROM LOGS', [], function (tx, results) {
      let len = results.rows.length;
      result.innerHTML += "<p><b>Found rows: " + len + "</b></p>";

      for (let i = 0; i < len; i++){
        result.innerHTML += "<br>" + results.rows.item(i).log;
      }
    }, errorDB);
  });
}

function errorDB(tx, error) {
  alert("error DB: " + JSON.stringify(error));
}
```

---

3. <http://www.sqlitetutorial.net/sqlite-create-table>

4. <https://www.w3.org/TR/webdatabase/#asynchronous-database-api>