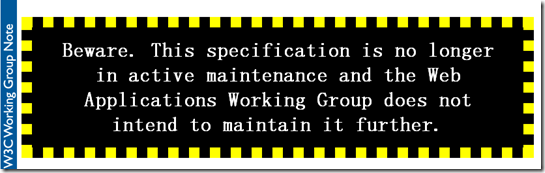
### **Web SQL Database**

在[HTML5 WebStorage](http://www.cnblogs.com/dolphinX/p/3348469.html)介绍了html5本地存储的Local Storage和Session Storage，这两个是以键值对存储的解决方案，存储少量数据结构很有用，但是对于大量结构化数据就无能为力了，灵活大不够强大。

### Web SQL Database

我们经常在数据库中处理大量结构化数据，html5引入Web SQL Database概念，它使用 SQL 来操纵客户端数据库的 API，这些 API 是异步的，规范中使用的方言是SQLlite，悲剧正是产生于此，[Web SQL Database规范页面](http://dev.w3.org/html5/webdatabase/)有着这样的声明

[](http://images.cnitblog.com/blog/349217/201311/03195047-034221fd6b77423aa92966d6ca38d5f5.png)

This document was on the W3C Recommendation track but specification work has stopped. The specification reached an impasse: all interested implementors have used the same SQL backend (Sqlite), but we need multiple independent implementations to proceed along a standardisation path.

 大概意思就是

这个文档曾经在W3C推荐规范上，但规范工作已经停止了。目前已经陷入了一个僵局：目前的所有实现都是基于同一个SQL后端（SQLite），但是我们需要更多的独立实现来完成标准化。

也就是说这是一个废弃的标准了，虽然部分浏览器已经实现，但。。。。。。。

### 三个核心方法

但是我们学一下也没什么坏处，而且能和现在W3C力推的IndexedDB做比较，看看为什么要废弃这种方案。Web SQL Database 规范中定义的三个核心方法：

1. openDatabase：这个方法使用现有数据库或新建数据库来创建数据库对象
2. transaction：这个方法允许我们根据情况控制事务提交或回滚
3. executeSql：这个方法用于执行SQL 查询

### openDatabase

我们可以使用这样简单的一条语句，创建或打开一个本地的数据库对象

var db = openDatabase('testDB', '1.0', 'Test DB', 2 \* 1024 \* 1024);

openDatabase接收五个参数：

1. 数据库名字
2. 数据库版本号
3. 显示名字
4. 数据库保存数据的大小（以字节为单位 )
5. 回调函数（非必须)

如果提供了回调函数，回调函数用以调用 changeVersion() 函数，不管给定什么样的版本号，回调函数将把数据库的版本号设置为空。如果没有提供回调函数，则以给定的版本号创建数据库。

### transaction

transaction方法用以处理事务，当一条语句执行失败的时候，整个事务回滚。方法有三个参数

1. 包含事务内容的一个方法
2. 执行成功回调函数（可选）
3. 执行失败回调函数（可选）

[IMG_257](http://www.cnblogs.com/dolphinX/p/javascript:void(0);)

db.transaction(function (context) {

context.executeSql('CREATE TABLE IF NOT EXISTS testTable (id unique, name)');

context.executeSql('INSERT INTO testTable (id, name) VALUES (0, "Byron")');

context.executeSql('INSERT INTO testTable (id, name) VALUES (1, "Casper")');

context.executeSql('INSERT INTO testTable (id, name) VALUES (2, "Frank")');

});

[IMG_258](http://www.cnblogs.com/dolphinX/p/javascript:void(0);)

这个例子中我们创建了一个table，并在表中插入三条数据，四条执行语句任何一条出现错误，整个事务都会回滚

### executeSql

executeSql方法用以执行SQL语句，返回结果，方法有四个参数

1. 查询字符串
2. 用以替换查询字符串中问号的参数
3. 执行成功回调函数（可选）
4. 执行失败回调函数（可选）

在上面的例子中我们使用了插入语句，看个查询的例子

[IMG_259](http://www.cnblogs.com/dolphinX/p/javascript:void(0);)

db.transaction(function (context) {

context.executeSql('SELECT \* FROM testTable', [], function (context, results) {

var len = results.rows.length, i;

console.log('Got '+len+' rows.');

for (i = 0; i < len; i++){

console.log('id: '+results.rows.item(i).id);

console.log('name: '+results.rows.item(i).name);

}

});

[IMG_260](http://www.cnblogs.com/dolphinX/p/javascript:void(0);)

### 完整示例

[IMG_261](http://www.cnblogs.com/dolphinX/p/javascript:void(0);)

<!DOCTYPE HTML><html><head>

<title>Web SQL Database</title></head><body>

<script type="text/javascript">

var db = openDatabase('testDB', '1.0', 'Test DB', 2 \* 1024 \* 1024);

var msg;

db.transaction(function (context) {

context.executeSql('CREATE TABLE IF NOT EXISTS testTable (id unique, name)');

context.executeSql('INSERT INTO testTable (id, name) VALUES (0, "Byron")');

context.executeSql('INSERT INTO testTable (id, name) VALUES (1, "Casper")');

context.executeSql('INSERT INTO testTable (id, name) VALUES (2, "Frank")');

});

db.transaction(function (context) {

context.executeSql('SELECT \* FROM testTable', [], function (context, results) {

var len = results.rows.length, i;

console.log('Got '+len+' rows.');

for (i = 0; i < len; i++){

console.log('id: '+results.rows.item(i).id);

console.log('name: '+results.rows.item(i).name);

}

});

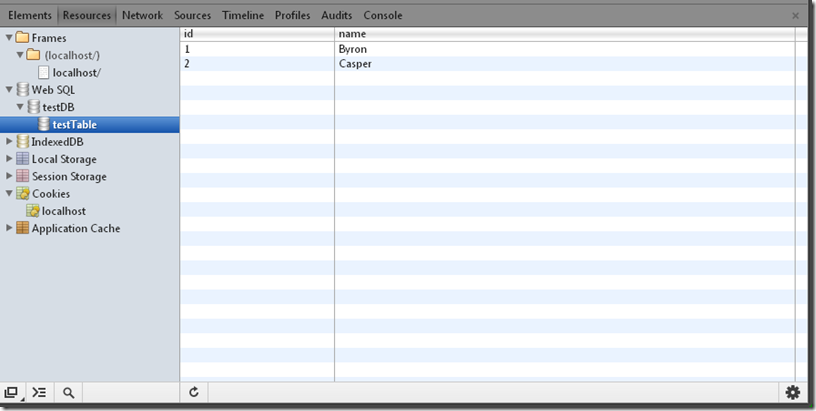
});

</script></body></html>

[IMG_262](http://www.cnblogs.com/dolphinX/p/javascript:void(0);)

### 最后

由于Web SQL Database规范已经被废弃，原因说的很清楚，当前的SQL规范采用SQLite的SQL方言，而作为一个标准，这是不可接受的，每个浏览器都有自己的实现这还搞毛的标准。这样浏览器兼容性就不重要了，估计慢慢会被遗忘。不过Chrome的控制台真心好用啊，神马cookie、Local Storage、Session Storage、Web SQL、IndexedDB、Application Cache等html5新增内容看的一清二楚，免去了很多调试代码工作。

[](http://images.cnitblog.com/blog/349217/201311/03201954-3d88b394c99d47deaae3126a67e730ac.png)