**HTML5 Web SQL 数据库**

Web SQL 数据库 API 并不是 HTML5 规范的一部分，但是它是一个独立的规范，引入了一组使用 SQL 操作客户端数据库的 APIs。

如果你是一个 Web 后端程序员，应该很容易理解 SQL 的操作。

你也可以参考我们的 [SQL 教程](http://www.runoob.com/sql/sql-tutorial.html)，了解更多数据库操作知识。

Web SQL 数据库可以在最新版的 Safari, Chrome 和 Opera 浏览器中工作。

**核心方法**

以下是规范中定义的三个核心方法：

1. **openDatabase**：这个方法使用现有的数据库或者新建的数据库创建一个数据库对象。
2. **transaction**：这个方法让我们能够控制一个事务，以及基于这种情况执行提交或者回滚。
3. **executeSql**：这个方法用于执行实际的 SQL 查询。

**打开数据库**

我们可以使用 openDatabase() 方法来打开已存在的数据库，如果数据库不存在，则会创建一个新的数据库，使用代码如下：

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

openDatabase() 方法对应的五个参数说明：

1. 数据库名称
2. 版本号
3. 描述文本
4. 数据库大小
5. 创建回调

第五个参数，创建回调会在创建数据库后被调用。

**执行查询操作**

执行操作使用 database.transaction() 函数：

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

db.transaction(function (tx) {

tx.executeSql('CREATE TABLE IF NOT EXISTS LOGS (id unique, log)');

});

上面的语句执行后会在 'mydb' 数据库中创建一个名为 LOGS 的表。

**插入数据**

在执行上面的创建表语句后，我们可以插入一些数据：

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

db.transaction(function (tx) {

tx.executeSql('CREATE TABLE IF NOT EXISTS LOGS (id unique, log)');

tx.executeSql('INSERT INTO LOGS (id, log) VALUES (1, "菜鸟教程")');

tx.executeSql('INSERT INTO LOGS (id, log) VALUES (2, "www.runoob.com")');

});

我们也可以使用动态值来插入数据：

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

db.transaction(function (tx) {

tx.executeSql('CREATE TABLE IF NOT EXISTS LOGS (id unique, log)');

tx.executeSql('INSERT INTO LOGS (id,log) VALUES (?, ?)', [e\_id, e\_log]); });

实例中的 e\_id 和 e\_log 是外部变量，executeSql 会映射数组参数中的每个条目给 "?"。

**读取数据**

以下实例演示了如何读取数据库中已经存在的数据：

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

db.transaction(function (tx) {

tx.executeSql('CREATE TABLE IF NOT EXISTS LOGS (id unique, log)');

tx.executeSql('INSERT INTO LOGS (id, log) VALUES (1, "菜鸟教程")');

tx.executeSql('INSERT INTO LOGS (id, log) VALUES (2, "www.runoob.com")'); });

db.transaction(function (tx) {

tx.executeSql('SELECT \* FROM LOGS', [], function (tx, results) {

var len = results.rows.length, i;

msg = "<p>查询记录条数: " + len + "</p>";

document.querySelector('#status').innerHTML += msg;

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

alert(results.rows.item(i).log );

} }, null);

});

**完整实例**

**实例**

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

var msg; db.transaction(function (tx) {

tx.executeSql('CREATE TABLE IF NOT EXISTS LOGS (id unique, log)');

tx.executeSql('INSERT INTO LOGS (id, log) VALUES (1, "菜鸟教程")');

tx.executeSql('INSERT INTO LOGS (id, log) VALUES (2, "www.runoob.com")');

msg = '<p>数据表已创建，且插入了两条数据。</p>';

document.querySelector('#status').innerHTML = msg; });

db.transaction(function (tx) {

tx.executeSql('SELECT \* FROM LOGS', [], function (tx, results) {

var len = results.rows.length, i;

msg = "<p>查询记录条数: " + len + "</p>";

document.querySelector('#status').innerHTML += msg;

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

msg = "<p><b>" + results.rows.item(i).log + "</b></p>";

document.querySelector('#status').innerHTML += msg;

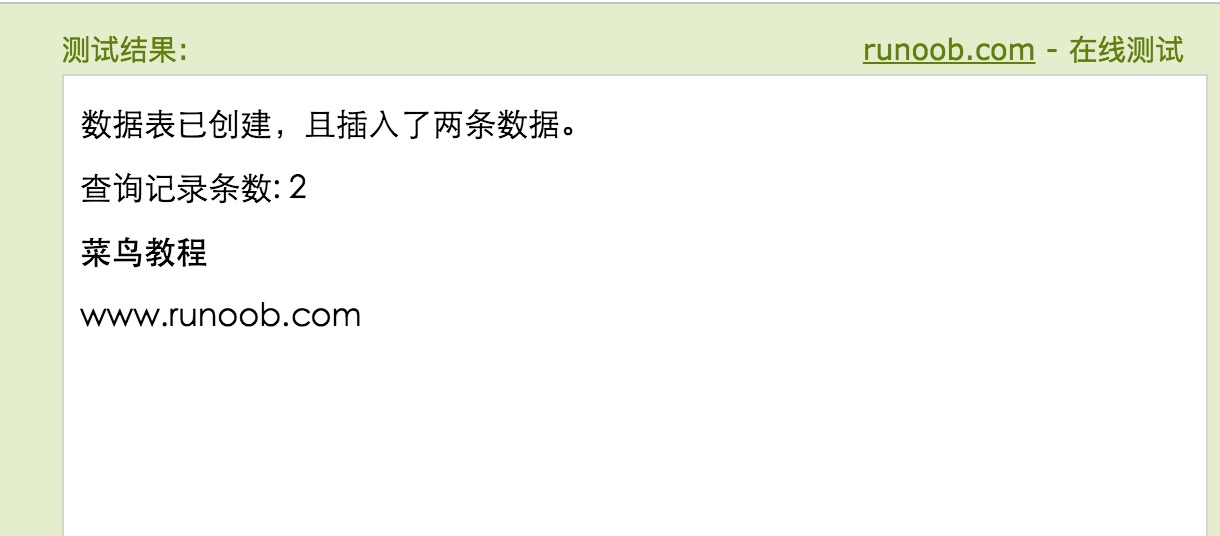
}

}, null);

});

[**尝试一下 »**](http://www.runoob.com/try/try.php?filename=tryhtml5_websql)

以上实例运行结果如下图所示：



**删除记录**

删除记录使用的格式如下：

db.transaction(function (tx) {

tx.executeSql('DELETE FROM LOGS WHERE id=1');

});

删除指定的数据id也可以是动态的：

db.transaction(function(tx) {

tx.executeSql('DELETE FROM LOGS WHERE id=?', [id]);

});

**更新记录**

更新记录使用的格式如下：

db.transaction(function (tx) {

tx.executeSql('UPDATE LOGS SET log=\'www.w3cschool.cc\' WHERE id=2');

});

更新指定的数据id也可以是动态的：

db.transaction(function(tx) {

tx.executeSql('UPDATE LOGS SET log=\'www.w3cschool.cc\' WHERE id=?', [id]);

});

**完整实例**

**实例**

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

var msg;

db.transaction(function (tx) {

tx.executeSql('CREATE TABLE IF NOT EXISTS LOGS (id unique, log)');

tx.executeSql('INSERT INTO LOGS (id, log) VALUES (1, "菜鸟教程")');

tx.executeSql('INSERT INTO LOGS (id, log) VALUES (2, "www.runoob.com")');

msg = '<p>数据表已创建，且插入了两条数据。</p>';

document.querySelector('#status').innerHTML = msg; });

db.transaction(function (tx) { tx.executeSql('DELETE FROM LOGS WHERE id=1');

msg = '<p>删除 id 为 1 的记录。</p>';

document.querySelector('#status').innerHTML = msg; });

db.transaction(function (tx) {

tx.executeSql('UPDATE LOGS SET log=\'www.w3cschool.cc\' WHERE id=2');

msg = '<p>更新 id 为 2 的记录。</p>';

document.querySelector('#status').innerHTML = msg; });

db.transaction(function (tx) {

tx.executeSql('SELECT \* FROM LOGS', [], function (tx, results){ var len = results.rows.length, i;

msg = "<p>查询记录条数: " + len + "</p>";

document.querySelector('#status').innerHTML += msg;

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

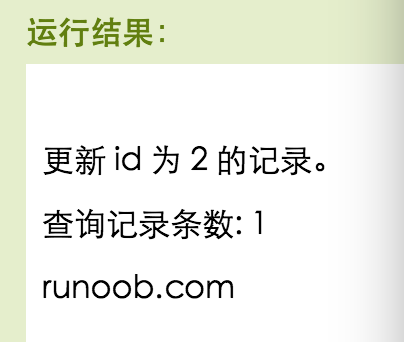
msg = "<p><b>" + results.rows.item(i).log + "</b></p>";

document.querySelector('#status').innerHTML += msg;

} }, null); });

[**尝试一下 »**](http://www.runoob.com/try/try.php?filename=tryhtml5_websql2)

以上实例运行结果如下图所示：



# HTML5前端数据库——Web SQL Database

[](https://www.jianshu.com/u/b6daf42c2cdd)

[datura\_lj](https://www.jianshu.com/u/b6daf42c2cdd) 关注

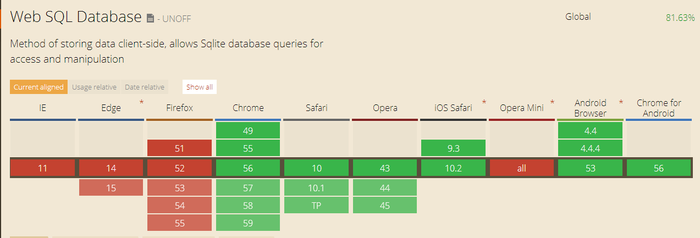
2017.03.10 14:51\* 字数 549 阅读 3023评论 6喜欢 7

**前言**：在HTML5 WebStorage介绍了html5本地存储的Local Storage和Session Storage，这两个是以键值对（字符串）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，规范中使用的是SQLlite（SQL后端）。  
**ps:附上**[**gihub**](https://link.jianshu.com/?t=https://github.com/Datura900607/websql/blob/master/webaql.html)**项目连接**

###### 它包含三个核心方法：

1、openDatabase：这个方法使用现有数据库或创建新数据库创建数据库对象。  
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：这个方法用于执行真实的SQL查询。

##### [兼容性](https://link.jianshu.com/?t=http://caniuse.com/#search=websql)：一些主流浏览器还是可以的。如下图：



#### 一、基础操作

###### 1. 新建（打开）数据库

**openDatabase**方法可以打开已经存在的数据库，不存在则创建：  
vardataBase = window.openDatabase(dbname, version, dbdesc, dbsize,function() {});  
openDatabasek中五个参数分别为：数据库名、版本号、描述、数据库大小、创建回调。创建回调没有也可以创建数据库。  
Ex：

var dataBase = window.openDatabase('websql', '1.0', 'Datura练习', 4\*1024\*1024,function() {});

###### 2. 建表

用到**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  
a). 包含事务内容的一个方法;  
b). 执行成功回调函数（可选）;  
c). 执行失败回调函数（可选）;  
Ex：

var tableName = 'websqlTable';*//创建表的名称*

*//这里定一个变量来存建表相关信息，并声明主键，需要存储的字段及格式（NAME、AGE、HEIGHT、WEIGHT）*

var creatTableSQL = 'CREATE TABLE IF NOT EXISTS '+ tableName + ' (rowid INTEGER PRIMARY KEY AUTOINCREMENT, NAME text,AGE text,HEIGHT text,WEIGTH text)';

dataBase.transaction(function (ctx,result) {

ctx.executeSql(creatTableSQL,[],function(ctx,result){

alert("表创建成功 " + tableName);*//建表成功*

},function(tx, error){

alert('创建表失败:' + tableName + error.message);*//建表失败*

});

});

###### 3. 向指定表中插入数据

同样用到**transaction**方法用以处理事务。

function websqlInsterDataToTable(tableName,NAME,AGE,HEIGHT,WEIGTH){

var insterTableSQL = 'INSERT INTO ' + tableName + ' (NAME,AGE,HEIGHT,WEIGTH) VALUES (?,?,?,?)';

dataBase.transaction(function (ctx) {

ctx.executeSql(insterTableSQL,[NAME,AGE,HEIGHT,WEIGTH],function (ctx,result){

console.log("插入" + tableName + NAME + "成功");

},

function (tx, error) {

alert('插入失败: ' + error.message);

});

});

}

websqlInsterDataToTable(websqlTable,"小红","18","175cm","40kg");

websqlInsterDataToTable(websqlTable,"小黄","17","180cm","45kg");

websqlInsterDataToTable(websqlTable,"小蓝","19","185cm","70kg");

websqlInsterDataToTable(websqlTable,"小绿","19","175cm","60kg");

websqlInsterDataToTable(websqlTable,"小青","21","162cm","52kg");

websqlInsterDataToTable(websqlTable,"小紫","25","195cm","80kg");

###### 4. 指定表中查找所有数据或查找某条数据

同样用到**transaction**方法用以处理事务。  
a). 查找所有数据

function websqlGetAllData(tableName){

var selectALLSQL = 'SELECT \* FROM ' + tableName;

dataBase.transaction(function (ctx) {

ctx.executeSql(selectALLSQL,[],function (ctx,result){

alert('查询成功: ' + tableName + result.rows.length);

var len = result.rows.length;

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

console.log("NAME = " + result.rows.item(i).NAME);

console.log("AGE = " + result.rows.item(i).AGE);

console.log("HEIGHT = " + result.rows.item(i).HEIGHT);

console.log("WEIGTH = " + result.rows.item(i).WEIGTH);

console.log("-------- 我是分割线 -------");

}

},

function (tx, error) {

alert('查询失败: ' + error.message);

});

});

}

websqlGetAllData(websqlTable);

b). 查找某条数据

function websqlGetAData(tableName,name){

var selectSQL = 'SELECT \* FROM ' + tableName + ' WHERE NAME = ?'

dataBase.transaction(function (ctx) {

ctx.executeSql(selectSQL,[name],function (ctx,result){

alert('查询成功: ' + tableName + result.rows.length);

var len = result.rows.length;

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

console.log("NAME = " + result.rows.item(i).NAME);

console.log("AGE = " + result.rows.item(i).AGE);

console.log("HEIGHT = " + result.rows.item(i).HEIGHT);

console.log("WEIGTH = " + result.rows.item(i).WEIGTH);

}

},

function (tx, error) {

alert('查询失败: ' + error.message);

});

});

}

websqlGetAData(websqlTable,"小紫");

###### 5. 删除数据

a). 删除表里的全部数据

function websqlDeleteAllDataFromTable(tableName){

var deleteTableSQL = 'DELETE FROM ' + tableName;

localStorage.removeItem(tableName);

dataBase.transaction(function (ctx,result) {

ctx.executeSql(deleteTableSQL,[],function(ctx,result){

alert("删除表成功 " + tableName);

},function(tx, error){

alert('删除表失败:' + tableName + error.message);

});

});

}

websqlDeleteAllDataFromTable(websqlTable);

b). 删除表里的一条数据

function websqlDeleteADataFromTable(tableName,name){

var deleteDataSQL = 'DELETE FROM ' + tableName + ' WHERE NAME = ?';

localStorage.removeItem(tableName);

dataBase.transaction(function (ctx,result) {

ctx.executeSql(deleteDataSQL,[name],function(ctx,result){

alert("删除成功 " + tableName + name);

},function(tx, error){

alert('删除失败:' + tableName + name + error.message);

});

});

}

websqlDeleteADataFromTable(websqlTable,"小蓝");

###### 6. 修改某条数据

function websqlUpdateAData(tableName,name,age){

var updateDataSQL = 'UPDATE ' + tableName + ' SET AGE = ? WHERE NAME = ?';

dataBase.transaction(function (ctx,result) {

ctx.executeSql(updateDataSQL,[age,name],function(ctx,result){

alert("更新成功 " + tableName + name);

},function(tx, error){

alert('更新失败:' + tableName + name + error.message);

});

});

}

websqlUpdateAData(websqlTable,"小红","1000")

##### 在chrome浏览器中的展示如下图：