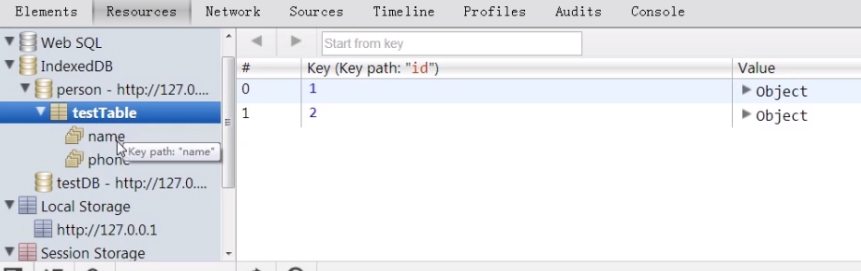
Web SQL数据库是一个客户端的关系型数据库，但是W3C已不再维护，且在移动端支持的不是很好。

IndexedDB其实是一种非关系型数据库，是一个对象数据库，作用域限制在同源中。

一、每个源中可以有任意数目的IndexedDB，而每个IndexedDB的名字必须唯一。

二、每个IndexedDB就是objectStore的集合，可以有一个或多个objectStore。

三、每个objectStore可以存储多个对象，但这些对象的key必须唯一。Key值可以是字符串，可以是日期对象，也可以是数字。你可以在存储这个对象时为这个对象创建一个key，但是其实在该对象被存储时，已经在对象内部生成了一个唯一的值，你可设置键路径。实际上，键路径是一个值，就是告诉数据库从一个对象中抽取出该对象的键。



主键（primary key）----对象（value）

IndexedDB的API都是异步的！

Var indexedDB=window.indexedDB;

Var openRequest=indexedDB.open(dbName);

打开一个数据库的时候，如果该数据库存在就打开，不存在就新建。

1.openRequest.onupgradeneeed 新的数据库创建或数据库版本号变更时触发，一般新建objectstore，版本号是会改变的

2.openRequest.onsuccess 成功打开数据库

3.openRequest.onerror 打开数据库失败

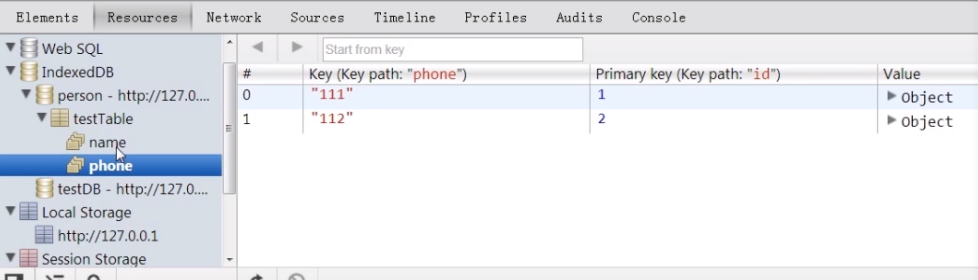
var objectstore=thisDB.createObjectStore(tableName,{keyPath:"id",autoIncrement});

//在创建表的时候，我们可以指定哪些字段是可以被索引的字段，Index取名的由来

objectstore.createIndex("name","name",{unique:false});//unique说明这个指定索引的字段是不是唯一的

objectstore.createIndex("phone","phone",{unique:false});

正是由于可以在对象存储区上创建索引，所以这是Indexed名字的由来。



创建索引就相当于创建了一个“次键”，但这些次键对应的对象极有可能不唯一unique,所以遍历的话有可能需要游标cursor。

var transaction=db.transaction([tableName],'readwrite');

var objectStore=transaction.objectstore(tablename);

对对象存储区的所有操作都要建立在“事务”上。创建事务，且找到相应的对象存储区。都是在事务上操作，保证了这些操作的一致性和可靠性。

objectStore.add() 向对象存储区添加数据

objectStore.get() 通过键获得数据

objectStore.put() 更新数据

objectStore.delete() 删除数据

objectStore.clear() 清空对象存储区

db.deleteObjectStore(StoreName) 调用数据库实例的deleteObjectStore可删除对象存储区

永远记住IndexedDB的所有API都是异步的。不仅是打开数据库的onsuccess、onerror、onupgradeneeded，包括你创建事务、用事务得到对象存储区，然后对对象存储区put、add等等。

如var request=store.get(value);

request.onsuccess=function(e){

var student=e.target.result;

student.age=35;

store.put(student);

};

创建事务对象、查询获得对象存储区都是同步的，但是put（），get（）或者通过游标openCursor（）查询对象存储区都是异步的，他们会立刻返回一个request对象，在成功或者失败的时候，在request对象上触发onsuccess和onerror事件，事件对象的event.target.result上返回你需要的值。

注意: openCursor也是异步，也是error、success事件。

遍历的话一般依靠游标，遍历的话直接objectStore.openCursor()。

//游标

objectStore.openCursor().onsuccess=function(event){

var cursor=event.target.result;

if(cursor){

console.log(cursor.key);

console.log(cursor.value);

lastCursor=cursor.key;

cursor.continue();//下一条游标

}else{

console.log('Done with cursor');

}

}

当想要查询表示键值范围的对象，可以创建一个IDBRange对象，传入给openCursor。

查询的话IDBRange的话:objectStore.index('name').openCursor(boundKeyRange)

//IDBKeyRange为全局对象，创建IDBRange对象

var boundKeyRange=IDBKeyRange.only(curName);

//表示在name一项中查找IDBRange objectStore.index('name').openCursor(boundKeyRange).omsuccess=function(event){

var cursor=event.target.result;

var rowData=cursor.value;

cursor.continue();

}

IndexedDB通过次键查找，则先查询对象存储区的命名索引，然后在该索引对象上get（）或者openCursor。

如果在对象存储区上直接get（），肯定只能得到一个对象，因为这时是通过主键查询。