HTML5存储

1 cookie的劣势

存储大小限制，仅4kb左右

单个域名下限制50个左右

污染请求头，浪费流量

2 webStorage

只能存储字符串，包括localStorage和sessionStorage，相同的使用方法，不同的存储时效。

localStorage：持久化存储，2~5mb

sessionStorage：会话存储，各浏览器限制不同。相同url不同标签页不能共享数据，关闭页面时失效。

设置存储内容 storage.setItem(key, value);

storage.key = value;

storage[key] = value;

获取存储内容 storage.getItem(key);

storage.key;

storage[key];

删除存储内容 storage.removeItem(key);

清空存储内容 storage.clear();

获取存储内容个数 storage.length;

存储容量超出限制异常 QuotaExceededError

3 数据库indexedDB

优势：存储类型丰富，条件搜索优势明显，存储容量更大，可以在worker中使用，适合存储复杂数据

劣势：学习曲线略陡峭，跨域存储限制，兼容性问题略严重（ios10，ie12，firefox单次存储不能超过50mb）

创建或打开数据库 var db = indexedDB.open(name, version);

关闭数据库 db.close();

创建或打开成功（回调） db.onsuccess = function() {}

创建或打开失败（回调） db.onerror = function() {}

数据库升级成功（回调） db.onupgradeneeded = function() {}

创建表 var table = db.result.createObjectStore(name, pk);

删除表 db.result.deleteObjectStore(name);

连接表 tst = db.result.transaction(name, type); type: readonly（默认）、readwirte

打开表 var table = tst.objectStore(name);

插入数据 table.add(data);

根据主键获取数据 var record = table.get(pkvalue).result;

获取所有数据 var records = table.getAll().result;

修改数据 table.put(data);

删除数据 table.delete(pkvalue);

清除数据 table.clear();

操作成功（回调） table.onsuccess = function() {}

操作失败（回调） table.onerror = function() {}

操作结果 table.getAll().result;

设置主键

设置自增主键：{ autoIncrement: true }

数据中字段作为主键：{ keyPath: 字段名 }

索引

可以使用存储记录中的值进行检索

索引自动更新

索引数据自动排序

创建索引 var index = table.createIndex(indexName, item, optionParameters);

索引名称 indexName

索引字段 keypath

索引配置参数 optionParameters: { unique: true（唯一约束） }

获得索引 var index = table.index(name);

通过索引获取数据 index.get(key).onsuccess = function (e) { e.target.result };

游标

可以查询指定数据集范围

拥有逆序遍历能力

创建游标 var cursor = table.openCursor(IDBKeyRange.rangecode(key));

更新数据 cursor.result.update(data);

删除数据 cursor.result.delete();

range code

all keys >= x && <= y bound(x, y)

all keys > x && < y bound(x, y, true, true)

all keys > x && <= y bound(x, y, true, false)

all keys >= x && < y bound(x, y, false, true)

all keys <= x upperBound(x)

all keys < x upperBound(x, true)

all keys >= y lowerBound(y)

all keys > y lowerBound(y, true)

the key = z only(z)

direction（游标取值方向）

顺序查询 next

顺序唯一查询 nextunique

逆序查询 prev

逆序唯一查询 prevunique

通过游标获取数据

一条数据

var cursor = table.openCursor(IDBKeyRange.only(key));

cursor.onsuccess = function() { console.log(cursor.result.value); };

多条数据

var cursor = table.openCursor(IDBKeyRange.upperBound(key));

if (cursor.result) {

console.log(cursor.result.value);

cursor.continue();

}

游标配合索引

索引按值搜索，游标范围遍历

索引排序，游标按序遍历

var index = table.index(name);

var cursor = index.openCursor(IDBKeyRange.upperBound(key));

if (cursor.result) {

console.log(cursor.result.value);

cursor.continue();

}