客户端存储的区别

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 特性 | cookies | localStorage | sessionStorage | indexedDB |
| 声明周期 | 关闭浏览器  可设置时间  一般服务器端设置 | 永久  除非手动删 | 关闭当前页面/浏览器 | 永久 |
| 数据大小 | 4K | 2.5M-5M | 2.5M-5M | 无上限 |
| 与服务器通信 | 会http通信  携带在http头中 | 不参与 | 不参与 | 不参与 |
| 同源策略 | 同源 | 同源 | 同源 | 同源 |

indexedDB案例：

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Document</title>

</head>

<body>

<input type="button" value="添加" id="btn1">

<input type="button" value="删除" id="btn2">

<input type="button" value="修改" id="btn3">

<input type="button" value="查询1条" id="btn4">

<input type="button" value="查询全部" id="btn5">

<h3>一条数据</h3>

<div id="box"></div>

<h3>所有数据</h3>

<div id="box2"></div>

</body>

</html>

<script>

console.log('开始操作DB');

var db;//数据库 实例

//1）打开数据库

var request = indexedDB.open('stuDB');

// - 数据库创建成功函数

request.onsuccess = function(e){

db = e.target.result;

console.log('打开数据库success');

}

// - 数据库创建失败函数

request.onerror = function(e){

console.log('打开数据库error');

}

//2）在数据库里 - 新建表

request.onupgradeneeded = function(e){

db = e.target.result;//数据库实例

//建表 - person 主键：id

if(!db.objectStoreNames.contains('person')){//不存在

console.log('不存在person表');

var objectStore = db.createObjectStore('person',{

keyPath:'id'

});

//或者设置主键自增

/\*

var objectStore = db.createObjectStore('person',{

autoIncrement:true

});

\*/

//添加索引

objectStore.createIndex('name','name',{unique:false});

objectStore.createIndex('age','age');

}

}

function getObjectStore(){

return db.transaction(['person'],'readwrite').objectStore('person');

}

//3）数据操作 ：

// 3-1）增加数据

function add(data){

// 拿到数据库 表格对象

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

console.log(objectStore);

// 往表格里 增加数据

var req = objectStore.add(data);

req.onsuccess = function(e){

console.log('数据写入成功');

}

req.onerror = function(e){

console.log('数据写入失败');

}

}

// 3-2) 删除数据

function remove(key){

var objectStore = getObjectStore();

//删除 - delete(主键)

var req = objectStore.delete(key);

req.onsuccess = function(e){

console.log('删除成功');

}

}

// 3-3) 修改数据

function update(data){

var request = getObjectStore();

var req = request.put(data);

req.onsuccess = function(){

console.log('数据更新成功');

}

req.onerror = function(){

console.log('数据更新失败');

}

}

// 3-4) 查询一条数据

function query(key,callback){

var request = getObjectStore();

var req = request.get(key);

req.onsuccess = function(e){

if(req.result){

callback(req.result)

}

}

req.onerror = function(){

console.log('事务失败');

}

}

// 3-5) 查询所有数据

function queryAll(callback){

var data = [];

var request = getObjectStore();

var req = request.openCursor();//打开游标

req.onsuccess = function(e){

var cursor = e.target.result;

if(cursor){

// console.log(cursor);

// console.log(cursor.value);

callback(cursor.value);

cursor.continue();//继续

}else{

console.log('没有更多数据了');

}

}

}

//操作

//增加

btn1.onclick = function(){

add({

id:3,

name:'pear',

age:15

});

}

//删除

btn2.onclick = function(){

remove(3);

}

//修改

btn3.onclick = function(){

update({

id:1,

name:'小乔',

age:18

})

}

//查询一条数据

btn4.onclick = function(){

query(1,function(data){

box.innerHTML = "<p>学生id："+data.id + '</p>';

box.innerHTML += "<p>姓名："+data.name + '</p>';

box.innerHTML += "<p>年龄："+data.age + '</p>';

});

}

btn5.onclick = function(){

queryAll(function(data){

//data是一条数据

console.log(data);

box2.innerHTML += "<p>";

box2.innerHTML += "学生id:"+data.id + " ";

box2.innerHTML += "姓名:"+data.name + " ";

box2.innerHTML += "年龄:"+data.age + " ";

box2.innerHTML += "</>";

});

}

</script>