# Vue-ls中memory存储的理解

经过尝试，发现Vue-ls使用memory存储的是临时数据，刷新浏览器之前存储的数据就会被清空。猜测是直接存储在变量当中的。当查看源码时，发现确实是存储在一个叫ls的变量当中。而localStorage和sessionStorage则是直接用原生对象进行封装。

/\* eslint class-methods-use-this: off \*/

var ls = {};

var MemoryStorageInterface =

/\*#\_\_PURE\_\_\*/

function () {

function MemoryStorageInterface() {

\_classCallCheck(this, MemoryStorageInterface);

Object.defineProperty(this, 'length', {

/\*\*

\* Define length property

\*

\* @return {number}

\*/

get: function get() {

return Object.keys(ls).length;

}

});

}

/\*\*

\* Get item

\*

\* @param {string} name

\* @returns {\*}

\*/

\_createClass(MemoryStorageInterface, [{

key: "getItem",

value: function getItem(name) {

return name in ls ? ls[name] : null;

}

/\*\*

\* Set item

\*

\* @param {string} name

\* @param {\*} value

\* @returns {boolean}

\*/

}, {

key: "setItem",

value: function setItem(name, value) {

ls[name] = value;

return true;

}

/\*\*

\* Remove item

\*

\* @param {string} name

\* @returns {boolean}

\*/

}, {

key: "removeItem",

value: function removeItem(name) {

var found = name in ls;

if (found) {

return delete ls[name];

}

return false;

}

/\*\*

\* Clear storage

\*

\* @returns {boolean}

\*/

}, {

key: "clear",

value: function clear() {

ls = {};

return true;

}

/\*\*

\* Get item by key

\*

\* @param {number} index

\* @returns {\*}

\*/

}, {

key: "key",

value: function key(index) {

var keys = Object.keys(ls);

return typeof keys[index] !== 'undefined' ? keys[index] : null;

}

}]);

return MemoryStorageInterface;

}();

var MemoryStorage = new MemoryStorageInterface();