

HTML新特性

1.cookie、webStorage、IndexedDB

特性	cookie	sessionStorage	localStorage	IndexedDB
生命周期	由服务器生成，可设置过期时间	页面关闭就清理	除非被手动清理，否则一直存在	除非被手动清理，否则一直存在
存储大小	4K	5M	5M	不限
与服务 器通信	每次都会携带在 Header 中	不参与	不参与	不参与

2.VueRouter实例

2.1 方法

- push()

push() 方法用于导航到不同的 URL ,并且会向 history 栈添加一个记录，其语法结构是：

```
router.push(location)
```

- replace()

replace() 方法用于导航到不同的 URL ,并且会替换掉当前 history 栈的记录，其语法结构是：

```
router.replace(location)
```

- go()

go() 方法在 history 记录中前进或后退多少步，其语法结构是：

```
route.go(n)
```

- back()

`back()` 方法在 `history` 记录中后退一步，其语法结构是：

```
router.back()
```

- **`forward()`**

`forward()` 方法在 `history` 记录中前进一步，其语法结构是：

```
router.forward()
```

2.2 路由对象(`route`)

路由对象表示当前激活的路由的状态信息，包含了当前 URL 解析得到的信息。

- **`fullPath` 属性**

`fullPath` 属性将返回解析后的 URL，包括查询参数和完整路径，其语法结构是：

```
route.fullPath
```

- **`path` 属性**

`path` 属性将返回当前路由的路径，总是解析为绝对路径，其语法结构是：

```
route.path
```

- **`query` 属性**

`query` 属性返回一个包含 URL 参数的键/对值，其语法结构是：

```
object route.query
```

- **`params` 属性**

`params` 属性将返回一个包含动态片断和全匹配片断的键/值对，其语法结构是：

```
object route.params
```

3.Web Worker

3.1 概述

JavaScript 采用单线程模型，也就代表所有任务只能在一个线程上完成，一次只能做一件事。如果前面的任务没有完成的话，后面的任务只能等待。

web worker 就是为 JavaScript 创造多线程环境，主线程创建 worker 线程后，将相关的、繁重的计算工作交由后者完成，主线程工作的同时，worker 线程也在工作，而且互不干扰。

3.2 web worker 需要注意事项

- 1、同源策略，必须保证主线程与 worker 线程同源
- 2、DOM 限制，worker 线程不允许访问 DOM 对象
- 3、通信联系，主线程与 worker 线程不能直接通信，**必须通过消息完成**
- 4、脚本限制，worker 线程可以使用 XMLHttpRequest 对象 发送 AJAX 请求
- 5、文件限制，worker 线程无法读取本地文件，**必须来自网络**

3.3 主线程

• 创建 worker 线程

```
variable = new Worker(脚本文件URL)
```

脚本文件必须遵守同源策略

• postMessage() 方法

postMessage() 方法向 worker 线程内部发送消息，其语法结构是：

```
worker.postMessage(message)
```

message 参数可以是任意类型的数据

• onmessage 属性

onmessage 属性当 worker 线程返回一条消息时被调用，其语法结构是：

```
worker.onmessage = function(e){  
    ...  
}
```

3.4 worker 线程

- message 事件

message 事件是 worker 线程内部的事件，监听主线程发送的消息，语法结构是：

```
this.addEventListener('message', function(e) {  
    ...  
});
```

- postMessage() 方法

postMessage() 方法用来向主线程发送消息，其语法结构是：

```
this.postMessage(message)
```