

Web Worker

简介：为 JavaScript 引入线程技术

JavaScript是一个单线程的语言，单线程最大的问题：同步代码会造成线程阻塞。原来怎么解决线程阻塞？异步方案：xmlhttprequest, jquery.ajax, promise(), setInterval(), setTimeout(), async/await, window.fetch(), axios,

如果再提供一个方式，用来解决性能问题更好了。webworker就是出现了。

1.什么是webworker?

webworker是H5新提供的处理比较耗时任务的解决方案：和异步的区别：异步是在单线程上执行的，而webworker在多线程上执行的。

Web Worker 的作用，就是为 JavaScript 创造【多线程】环境，允许【主线】程创建 Worker 线程，将一些任务分配给各个【分线程】运行。

2.Web Worker的特点?

- 同源限制：分配给 Worker 线程运行的脚本文件，必须与主线程的脚本文件同源，**只能运行在同源策略下**。
- 文件限制：Worker 线程无法读取本地文件，即不能打开本机的文件系统（file:///），它所加载的脚本，必须来自网络。
- DOM限制：Worker 线程所在的全局对象，与主线程不一样，无法读取主线程所在网页的 DOM 对象，也无法使用document、window、parent、history、screen这些对象。但是，Worker 线程可以navigator对象和location对象。web worker的出现是了解决耗时任务引起的性能问题，不建议操作DOM。
- 通信联系：Worker 线程和主线程不在同一个上下文环境（不在一条路），它们不能直接通信，必须通过消息完成。
- 脚本限制：Worker 线程不能执行alert()方法和confirm()方法，但可以使用 XMLHttpRequest 对象发出 AJAX 请求。

使用：

1. 创建新的 worker 对象

```
var worker = new Worker('task.js');
```

2. 创建 Worker 之后，通过调用 postMessage() 方法启动：

```
worker.postMessage('');
```

3. 通过监听message事件，监听的过程中等待分线程（task.js）的结果。ev事件对象可以接收子线程发送过来的结果。

```
worker.onmessage = function(ev) {  
    console.log(ev.data);  
}
```

4. task.js

```
// 分线程中this不再指向window
this.onmessage = function(ev) {
    this.postMessage(ev.data)
}
```

this可以写**self**代替，**self** 和 **this** 指的都是 **Worker** 的全局作用域；所以还可以省略不写

5. 加载外部脚本

通过 `importScripts()` 函数将外部脚本文件或库加载到 Worker 中

此示例将 `script1.js` 和 `script2.js` 加载到了 Worker 中：

- `task.js`：

```
importScripts('script1.js');
importScripts('script2.js');
```

- 也可以写成单个导入语句：

```
importScripts('script1.js', 'script2.js');
```

6. 处理错误 `task.js`：

```
onerror = function(){
    console.log("worker执行中所有的错误都能被onerror捕获")
}

onmessageerror = function(){
    console.log("worker执行中和消息传递相关的错误才能被onmessageerror捕获")
}
```

6. 停止worker

停止 Worker 的方法有两种：在主网页中调用 `worker.terminate()`，或在 Worker 本身内部调用 `self.close()`。

```
worker.terminate()//在主线程关闭
self.close();//在分线程关闭
```

文档：

<https://www.html5rocks.com/zh/tutorials/workers/basics/>

<http://www.ruanyifeng.com/blog/2018/07/web-worker.html>

http://www.w3school.com.cn/html5/html_5_webworkers.asp

<https://developer.mozilla.org/zh-CN/docs/Web/API/Worker>

https://developer.mozilla.org/zh-CN/docs/Web/API/Web_Workers_API