

浏览器 - 创建一个事件中心

作者: [Vanessa](#)

原文链接: <https://ld246.com/article/1572445359241>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

2019-10-30

描述

使用 `emit`, `on` 和 `off` 方法创建一个发布/订阅(`publish-subscribe`)的事件中心。

提示

- 使用 `Object.create(null)` 创建一个空的 `hub` 对象, 他不需要继承来自 `Object.prototype` 的属性
- `emit`, 首先基于 `event` 参数找到数组中对应的一组操作, 然后使用 `Array.prototype.forEach()` 运这组操作中的每一项, 其参数都为 `data`
- `on`, 如果事件不存在, 则为其创建一个数组, 然后使用 `Array.prototype.push()` 将操作添加到数组中
- `off`, 使用 `Array.prototype.findIndex()` 找到事件数组中对应操作的索引, 使用 `Array.prototype.splice()` 将其移除

代码

```
const createEventHub = () => ({
  hub: Object.create(null),
  emit(event, data) {
    (this.hub[event] || []).forEach(handler => handler(data));
  },
  on(event, handler) {
    if (!this.hub[event]) this.hub[event] = [];
    this.hub[event].push(handler);
  },
  off(event, handler) {
    const i = (this.hub[event] || []).findIndex(h => h === handler);
    if (i > -1) this.hub[event].splice(i, 1);
    if (this.hub[event].length === 0) delete this.hub[event];
  }
});
```

示例

事件中心的调用:

```
const handler = data => console.log(data);
const hub = createEventHub();
let increment = 0;

// 订阅: 监听不同类型的事件
hub.on('message', handler);
hub.on('message', () => console.log('Message event fired'));
hub.on('increment', () => increment++);

// 发布: 提交事件以调用所有订阅过的操作
hub.emit('message', 'hello world'); // logs 'hello world' and 'Message event fired'
hub.emit('message', { hello: 'world' }); // logs the object and 'Message event fired'
```

```
hub.emit('increment'); // `increment` variable is now 1
```

```
// 取消订阅: 从监听器中停止 'message' 事件中指定的操作  
hub.off('message', handler);
```

返回总目录

[每天 30 秒系列之 JavaScript 代码](#)