

file对象

<https://www.electronjs.org/docs/api/file-object>

API 参考

简介

进程对象

支持的命令行开关

环境变量

Chrome 扩展支持

重要的API变更

自定义 DOM 元素

`File` 对象

`<webview>` 标签

`window.open` 函数

`BrowserWindowProProxy` 对象

Main Process 模块

`app`

`autoUpdater`

`BrowserView`

`BrowserWindow`



Bluetooth

Certifica

Certifica

体

Cookie 文

CPUUsa

崩溃报告

CustomS

Desktop

Display

事件对象

Extensio

扩展信息

FileFilter

FilePath

GPU 特性

InputEve

IOCount

g Terminal Help index.html - electron-quick-start - Visual Studio Code

package.json JS main.js JS renderer.js index.html x # styles.css

```
index.html > html > body > div#drag_test.for_file_drag
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Hello World!</title>
6     <link rel="stylesheet" href="css/styles.css" />
7   </head>
8   <body>
9     <div class="for_file_drag" id="drag_test">
10       <h2>File对象</h2>
11       <span>往这里拖文件</span>
12     </div>
13     <div>
14       <h2>Process</h2>
15       <button onclick="getProcessInfo()">查看Process信息</button>
16     </div>
17
18     <!-- You can also require other files to run in this process -->
19     <script src="./renderer.js"></script>
20   </body>
21 </html>
22
```

RELAPPOINT
js 40:3
194:13

DEBUG Debug Main Process

package.json JS main.js JS renderer.js x index.html # styles.css

VARIABLES WATCH CALL STACK LOADED SCRIPTS

```
JS renderer.js > ...
1 // This file is required by the index.html file and will
2 // be executed in the renderer process for that window.
3 // All of the Node.js APIs are available in this process.
4 const electron = require("electron");
5
6 function getProcessInfo() {
7   console.log("getCPUUsage:", process.getCPUUsage());
8   console.log('env', process.env);
9   console.log('arch', process.arch);
10 }
11
12 const dragWrapper = document.getElementById("drag_test");
13 dragWrapper.addEventListener("drop", (e) => {
14   e.preventDefault();
15   const files = e.dataTransfer.files;
16   if(files && files.length > 0) {
17     const path = files[0].path;
18     console.log('path:', path);
19   }
20 })
21
22 dragWrapper.addEventListener("dragover", (e) => {
23   e.preventDefault();
24 })
```

阻止 dragover 的行为

