

# 查看环境信息

<https://www.electronjs.org/docs/api/process>

- electron-forge
- electron-builder
- electron-react-boilerplate
- 其它工具和模板

## 🔧 详细信息

### 安装 Electron

- 代理
- 自定义镜像和缓存
- 故障排查

### Electron发布 & 开发者反馈

- 版本规则
- 发布时间线

### 测试 Widevine CDM

## ⚙️ 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

## 🧪 API 结构

BluetoothDevice Object 蓝牙设备

Certificate Object 证书对象

CertificatePrincipal Object 证书体

Cookie 对象

CPUUsage 对象

崩溃报告对象

CustomScheme 对象

文档 | Electron

进程 | Electron

+

不安全 | [electronjs.org/docs/api/process](https://www.electronjs.org/docs/api/process)

In sandboxed renderers the `process` object contains only a subset of the APIs:

- `crash()`
- `hang()`
- `getCreationTime()`
- `getHeapStatistics()`
- `getProcessMemoryInfo()`
- `getSystemMemoryInfo()`
- `getSystemVersion()`
- `getCPUUsage()`
- `getIOCounters()`
- `argv`
- `execPath`
- `env`
- `pid`
- `arch`
- `platform`
- `沙盒化`
- `type`
- `version`
- `versions`
- `mas`
- `windowsStore`

```
</div>
<div>
  <h2>Process</h2>
  <button onclick="getProcessInfo()">查看Process信息</button>
</div>
```

JS main.js ▶ createWindow

```
1 // Modules to control application life and create native browser window
2 const {app, BrowserWindow} = require('electron')
3 const path = require('path')
4
5 // Keep a global reference of the window object, if you don't, the window will
6 // be closed automatically when the JavaScript object is garbage collected.
7 let mainWindow
8
9 function createWindow () {
10   // Create the browser window.
11   mainWindow = new BrowserWindow({
12     width: 800,
13     height: 600,
14     webPreferences: {
15       preload: path.join(__dirname, 'preload.js'),
16       nodeIntegration: true
17     }
18   })
19
20   mainWindow.webContents.on("did-finish-load", () => {
21     console.log("****did-finish-load");
22   })
23
24   mainWindow.webContents.on("dom-ready", () => {
25     console.log("****dom-ready");
26   })
27
28   // and load the index.html of the app.
29   mainWindow.loadFile('index.html');
30
31   // Open the DevTools.
32   // mainWindow.webContents.openDevTools()
33
34   // Emitted when the window is closed.
35   mainWindow.on('closed', function () {
36     // Dereference the window object, usually you would store windows
37     // in an array if your app supports multi windows, this is the time
38     // when you should delete the corresponding element.
39     mainWindow = null
```

```

// Keep a global reference of the window object, if you don't, the window will
// be closed automatically when the JavaScript object is garbage collected.
let mainWindow

function createWindow () {
  // Create the browser window.
  mainWindow = new BrowserWindow({
    width: 800,
    height: 600,
    webPreferences: {
      preload: path.join(__dirname, 'preload.js'),
      nodeIntegration: true // 这个属性非常重要，用户的require和process等变量的使用需要事先设定上这个属性
    }
  })

  mainWindow.webContents.on("did-finish-load", () => {
    console.log("****did-finish-load");
  })

  mainWindow.webContents.on("dom-ready", () => {

```

初始化window的时候要添加  
nodeIntegration:true 不然回报下面的错

Uncaught ReferenceError: require is not defined  
at renderer.js:4

C:\develop\quanzs\trinity-warnings.js:  
Electron Security Warning (Insecure Content-Security-Policy) Th

Terminal Help renderer.js - electron-quick-start - Visual Studio Code

```

package.json JS main.js JS renderer.js x index.html
JS renderer.js ▶ getProcessInfo
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
5
6 function getProcessInfo() {
7   console.log("getCPUUsage:", process.getCPUUsage());
8   console.log('env', process.env);
9   console.log('arch', process.arch);
10 }

```