

快捷键

<https://www.electronjs.org/docs/api/global-shortcut>

进程：主进程

`globalShortcut` 模块可以在操作系统中注册/注销全局快捷键, 以便可以为操作定制各种快捷键。

注意: 快捷方式是全局的; 即使应用程序没有键盘焦点, 它也仍然在持续监听键盘事件。 This module cannot be used before the `ready` event of the app module is emitted.

```
const { app, globalShortcut } = require('electron')

app.whenReady().then(() => {
  // Register a 'CommandOrControl+X' shortcut listener.
  const ret = globalShortcut.register('CommandOrControl+X', () => {
    console.log('CommandOrControl+X is pressed')
  })

  if (!ret) {
    console.log('registration failed')
  }

  // 检查快捷键是否注册成功
  console.log(globalShortcut.isRegistered('CommandOrControl+X'))
})

app.on('will-quit', () => {
  // 注销快捷键
  globalShortcut.unregister('CommandOrControl+X')

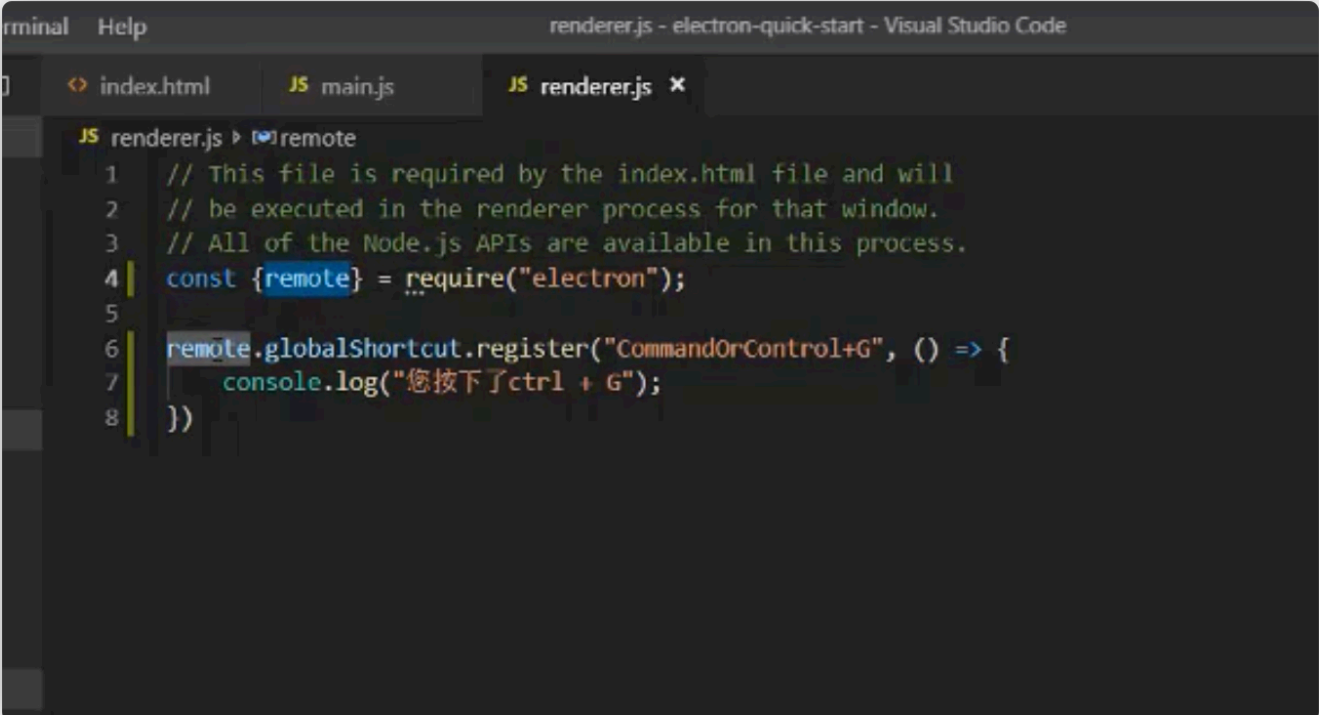
  // 注销所有快捷键
  globalShortcut.unregisterAll()
})
```

主进程注册快捷键

```
main.js - electron-quick-start - Visual Studio Code
index.html JS main.js JS renderer.js
JS main.js ▶ globalShortcut
1 // Modules to control application life and create native browser window
2 const {app, BrowserWindow, globalShortcut} = 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 // and load the index.html of the app.
21 mainWindow.loadFile('index.html')
22
23 // Open the DevTools.
```

```
main.js - electron-quick-start - Visual Studio Code
File Edit Selection View Go Debug Terminal Help
index.html JS main.js JS renderer.js
JS main.js ▶ app.on('window-all-closed') callback
34
35 // This method will be called when Electron has finished
36 // initialization and is ready to create browser windows
37 // Some APIs can only be used after this event occurs.
38 app.on('ready', () => {
39   createWindow();
40   globalShortcut.register("CommandOrControl+I", ()=>{
41     console.log("您按下了ctrl + I");
42   })
43   const isRegetered = globalShortcut.isRegistered("CommandOrControl+I");
44   console.log('**isRegetered', isRegetered);
45 })
46
47 // Quit when all windows are closed.
48 app.on('window-all-closed', function () {
49   globalShortcut.unregisterAll();
50   // On macOS it is common for applications and their menu bar
51   // to stay active until the user quits explicitly with Cmd + Q
52   if (process.platform !== 'darwin') app.quit()
53 })
54
55 app.on('activate', function () {
56   // On macOS it's common to re-create a window in the app when the
57   // dock icon is clicked and there are no other windows open.
58   if (mainWindow === null) createWindow()
59 })
60
61 // In this file you can include the rest of your app's specific main process
62 // code. You can also put them in separate files and require them here.
63
64 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
C:\development\training\video\electron\code\electron-quick-start\node_modules/.bin/electron.cmd --inspect
```

渲染进程快捷键注册



```
renderer.js - electron-quick-start - Visual Studio Code
index.html  JS main.js  JS renderer.js x
JS renderer.js > remote
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 {remote} = require("electron");
5
6 remote.globalShortcut.register("CommandOrControl+G", () => {
7   console.log("您按下了ctrl + G");
8 })
```