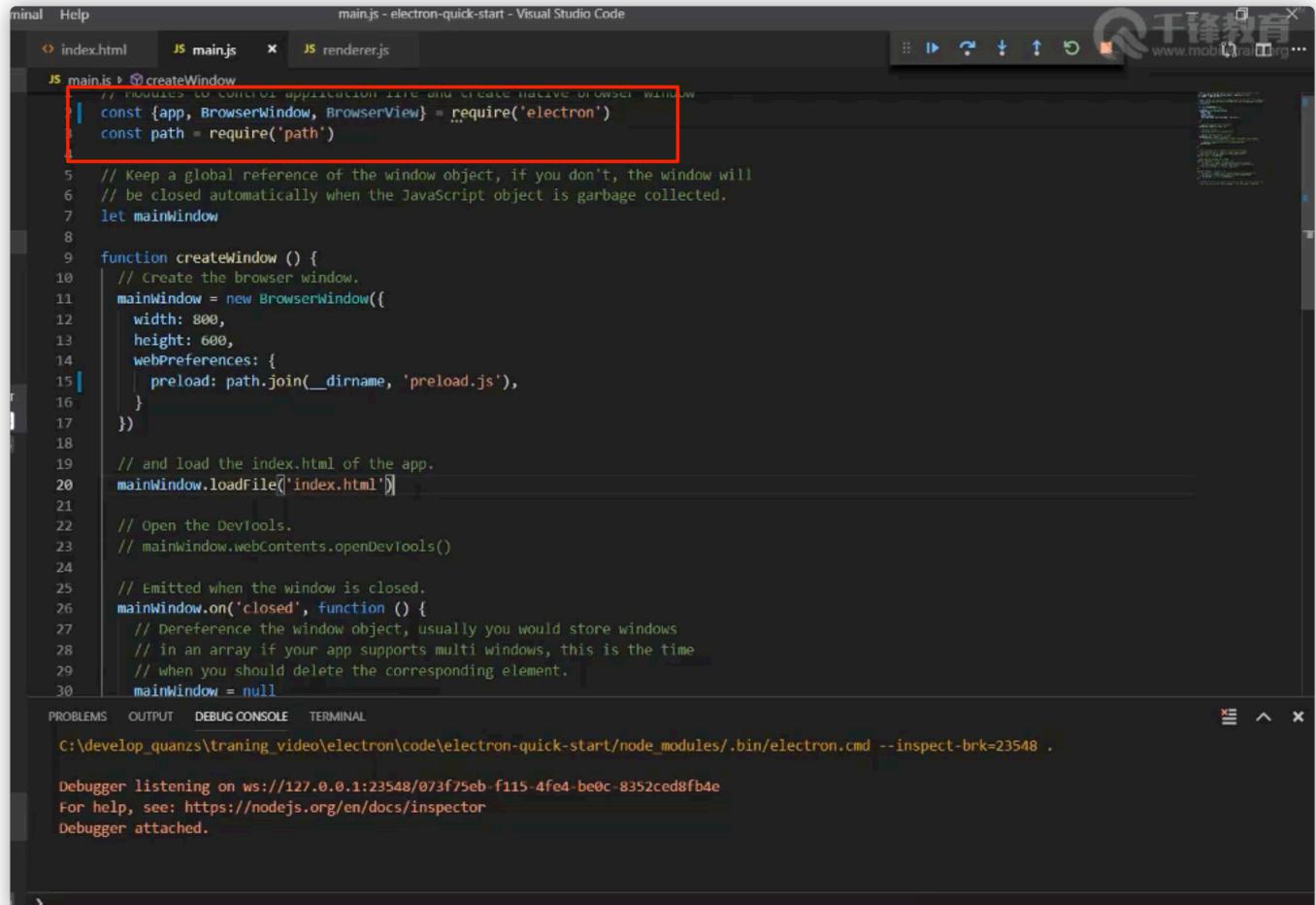


browseView

<https://www.electronjs.org/docs/api/browser-window>



```
mainl Help main.js - electron-quick-start - Visual Studio Code
index.html JS main.js x JS renderer.js
JS main.js > createWindow
  // Modules to control application life and create native browser window
  const {app, BrowserWindow, BrowserView} = require('electron')
  const path = require('path')

  // 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'),
      }
    })

    // and load the index.html of the app.
    mainWindow.loadFile('index.html')

    // Open the DevTools.
    // mainWindow.webContents.openDevTools()

    // Emitted when the window is closed.
    mainWindow.on('closed', function () {
      // Dereference the window object, usually you would store windows
      // in an array if your app supports multi windows, this is the time
      // when you should delete the corresponding element.
      mainWindow = null
    })
  }

  // Emitted when the window is closed.
  mainWindow.on('closed', function () {
    // Dereference the window object, usually you would store windows
    // in an array if your app supports multi windows, this is the time
    // when you should delete the corresponding element.
    mainWindow = null
  })
}

// Load the index.html of the application.
if (require.main === module) {
  createWindow()
}

// If you want to allow running it as a standalone application
// instead of as a node module, you can use the following code
// instead of 'if (require.main === module)':
// app.setAsPrimaryApplication()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
C:\develop_quanzs\training_video\electron\code\electron-quick-start\node_modules/.bin/electron.cmd --inspect-brk=23548 .

Debugger listening on ws://127.0.0.1:23548/073f75eb-f115-4fe4-be0c-8352ced8fb4e
For help, see: https://nodejs.org/en/docs/inspector
Debugger attached.
```

```
File Edit Selection View Go Debug Terminal Help
main.html JS main.js JS renderer.js
DEBUG ▶ Debug Main Process
VARIABLES
Local
this: App
Closure
Global
WATCH
CALL STACK PAUSED ON BREAKPOINT
(anonymous function) main.js 57:3
emit events.js 199:15
const view = new BrowserView();
view.setBounds({
  x: 10,
  y: 10,
  width: 300,
  height: 200
})
view.webContents.loadURL('https://www.baidu.com');

mainWindow.setBrowserView(view);
// Open the DevTools.
// mainWindow.webContents.open
// Emitted when the window is

```

实例方法

使用 `new BrowserView` 创建的对象具有以下实例方法:

`view.destroy()`

强制关闭视图, 不会为网页发出 `unload` 和 `beforeunload` 事件。完成视图后, 请调用此函数, 以便尽快释放内存和其他资源。

`view.isDestroyed()`

返回 `Boolean` - 判断窗口是否被销毁

`view.setAutoResize(options)` 实验功能

- `options Object`
 - `width Boolean` - 如果为 `true`, 视图宽度跟随窗口变化. 默认为 `false`.
 - `height Boolean` - 如果为 `true`, 视图高度跟随窗口变化. 默认为 `false`.
 - `horizontal Boolean` - If `true`, the view's x position and width will grow and shrink proportionally with the window. `false` by default.
 - `vertical Boolean` - If `true`, the view's y position and height will grow and shrink proportionally with the window. `false` by default.

`view.setBounds(bounds)` 实验功能

- `bounds Rectangle`

File View Go Debug Terminal Help

main.js - electron-quick-start - Visual Studio Code

Debug Main Process ▾ ⚙️ 🏁 index.html JS main.js ✘ JS renderer.js

JS main.js > createWindow > setTimeout() callback

```
15 |     preload: path.join(__dirname, 'preload.js'),
16 |
17 | }
18 |
19 | // and load the index.html of the app.
20 | mainWindow.loadFile('index.html')
21 |
22 | const view = new BrowserView();
23 | view.setBounds({
24 |   x: 10,
25 |   y: 10,
26 |   width: 300,
27 |   height: 200
28 | })
29 | view.webContents.loadURL('https://www.baidu.com');
30 |
31 | mainWindow.setBrowserView(view);
32 |
33 | setTimeout(()=>{
34 |   view.destroy();
35 | }, 5000);
36 |
37 | // Open the DevTools.
38 | // mainWindow.webContents.openDevTools()
```

