



SMU | LYLE
SCHOOL OF ENGINEERING

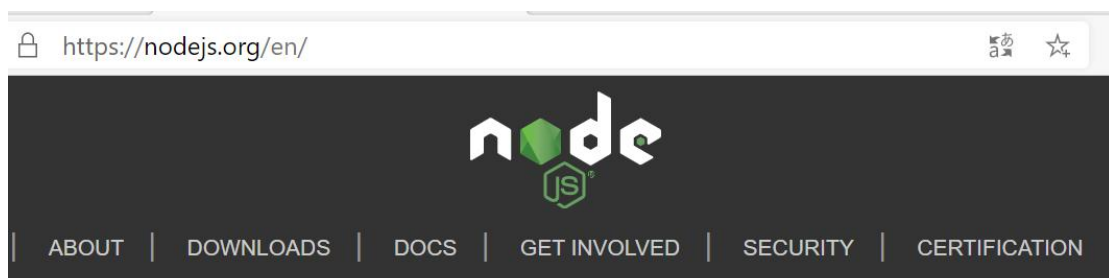
CS 7345 ADVANCED APPLICATIONS: LAB 5

Qing Gao 48358925

1. Document the complete process to install and build the code base with the native framework.

1) First, we will install the required software.

a. Download the node.js in <https://nodejs.org/en/>



Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Node.js training and exams discounted through Dec. 6th

Download for Windows (x64)

16.13.1 LTS

Recommended For Most Users

17.2.0 Current

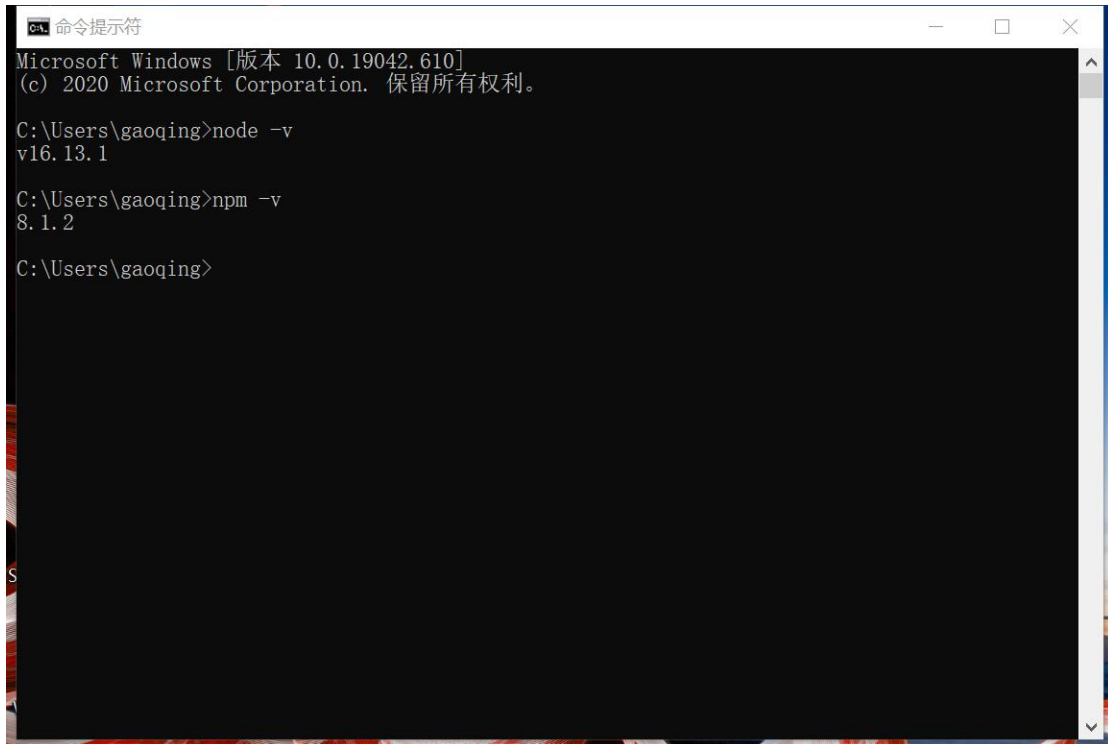
Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#)

[Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the [Long Term Support \(LTS\) schedule](#)

b. Use “node -v” and “npm -v” to check the version



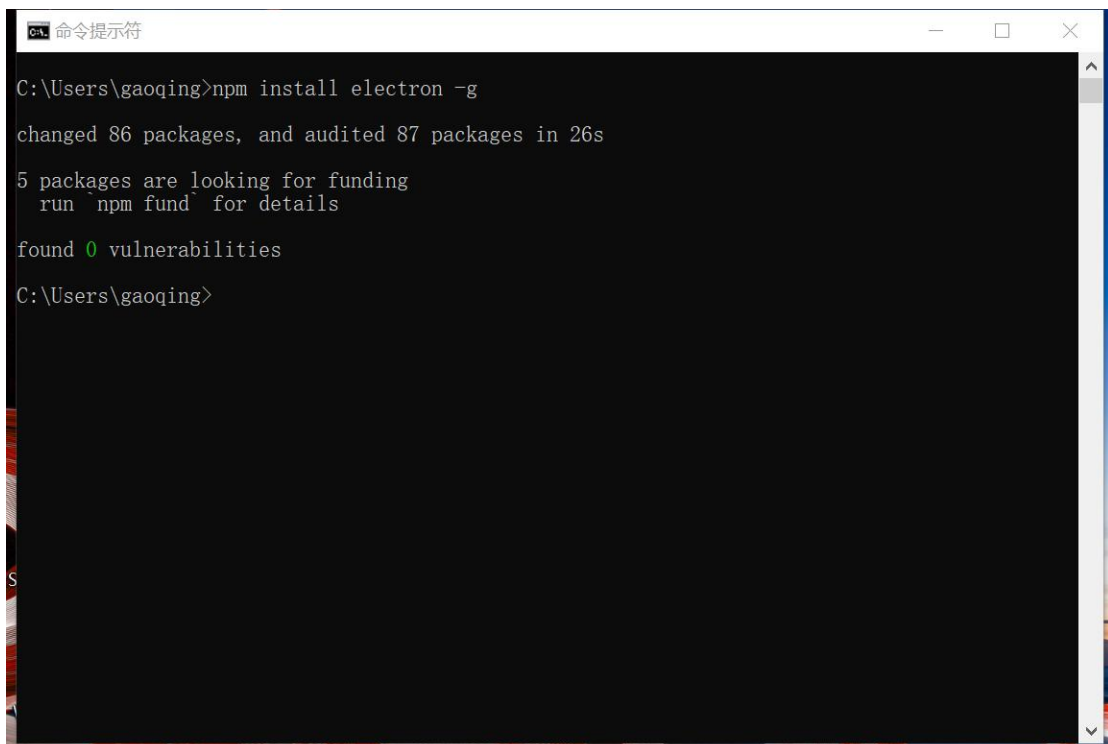
```
命令提示符
Microsoft Windows [版本 10.0.19042.610]
(c) 2020 Microsoft Corporation. 保留所有权利。

C:\Users\gaoqing>node -v
v16.13.1

C:\Users\gaoqing>npm -v
8.1.2

C:\Users\gaoqing>
```

c. Use “npm install electron -g” and “npm install electron-packager -g ” to install electron and packaging dependency.



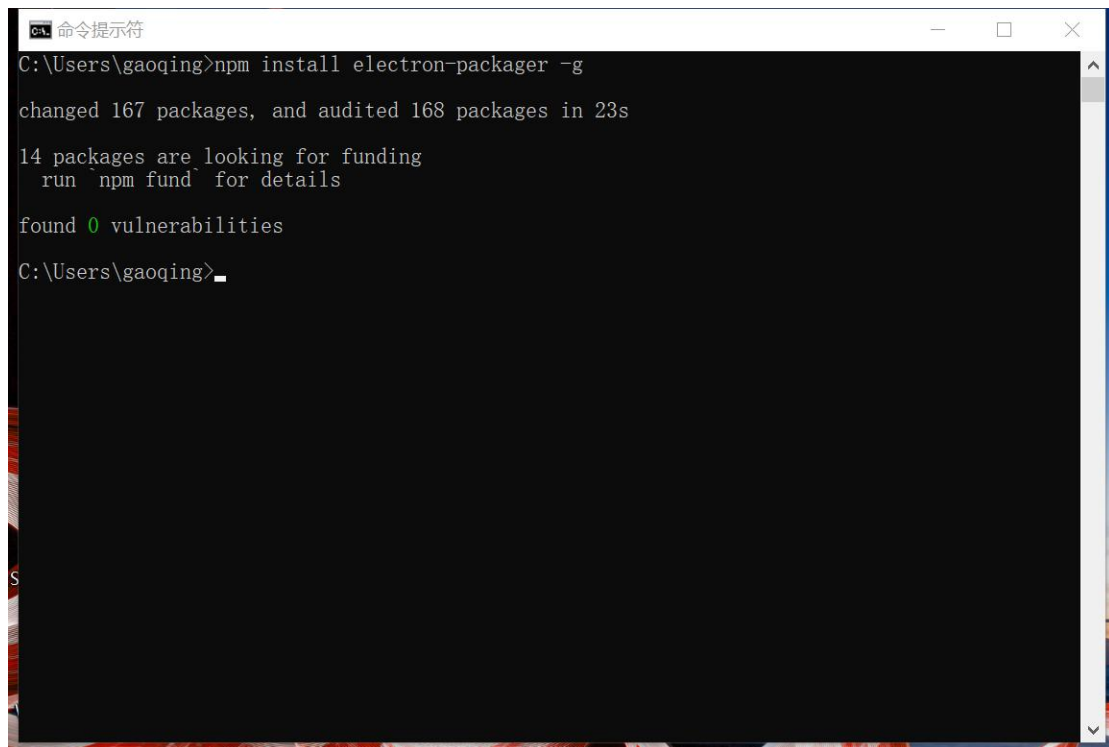
```
命令提示符

C:\Users\gaoqing>npm install electron -g
changed 86 packages, and audited 87 packages in 26s

5 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\gaoqing>
```

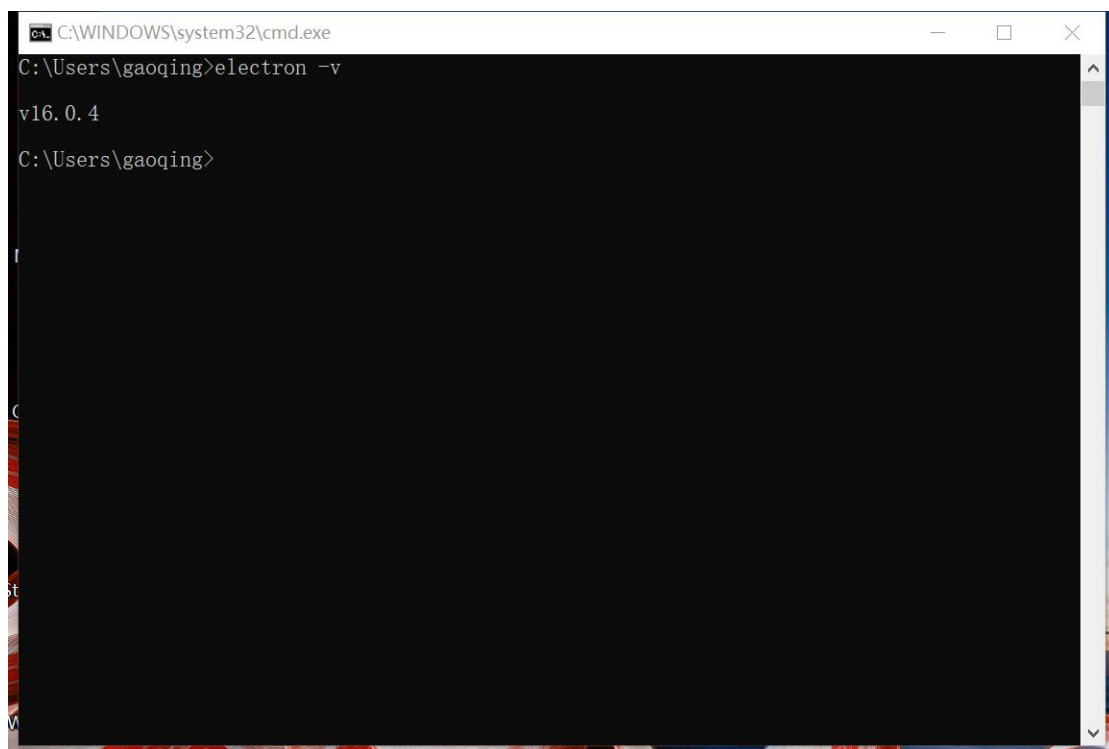


```
命令提示符
C:\Users\gaoqing>npm install electron-packager -g
changed 167 packages, and audited 168 packages in 23s

14 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
C:\Users\gaoqing>
```

d. Use “electron -v” to check the version of electron.



```
C:\WINDOWS\system32\cmd.exe
C:\Users\gaoqing>electron -v
v16.0.4
C:\Users\gaoqing>
```

Now we can use electron to package web files into local files!

2) Packing the file

e. Create the package.json file and add the following code.

```
{  
  "name": "app-name",  
  "version": "0.1.0",  
  "main": "main.js",  
  "dependencies": {  
    "nodejs-websocket": "^1.7.2",  
    "websocket": "^1.0.34"  
  }  
}
```

/"name" is our app's name

/"version" is our app's version

/"main.js" is to realize the function of electron

/"dependencies" could help us packing

f. "main.js" created the electron application to realize the functions of notifying and calling the page in Lab4. "index.html" is come from lab4.

```

const {app, BrowserWindow} = require('electron')
const path = require('path')
const url = require('url')

let win

function createWindow () {
  // Create the browser window.
  win = new BrowserWindow({width: 800, height: 600})

  // and load the index.html of the app.
  win.loadURL(url.format({
    pathname: path.join(__dirname, 'index.html'),
    protocol: 'file:',
    slashes: true
  }))

  // Open the DevTools.
  //win.webContents.openDevTools()
  // Emitted when the window is closed.
  win.on('closed', () => {
    win = null
  })
}

app.on('ready', createWindow)

// Quit when all windows are closed.
app.on('window-all-closed', () => {
  if (process.platform !== 'darwin') {
    app.quit()
  }
})

app.on('activate', () => {
  if (win === null) {
    createWindow()
  }
})

```

```

const createWindow_tips = () => {
  // create window
  const mainWindow = new BrowserWindow({
    width: 800,
    height: 600,
    webPreferences: {
      // start node
      nodeIntegration: true,
      // De context isolation
      contextIsolation: false,
      // start remote
      enableRemoteModule: true,
    }
  });
  // Load local file
  mainWindow.loadFile(path.join(__dirname, "index.html"));
  // Load remote address
  // mainWindow.loadURL('https://github.com');

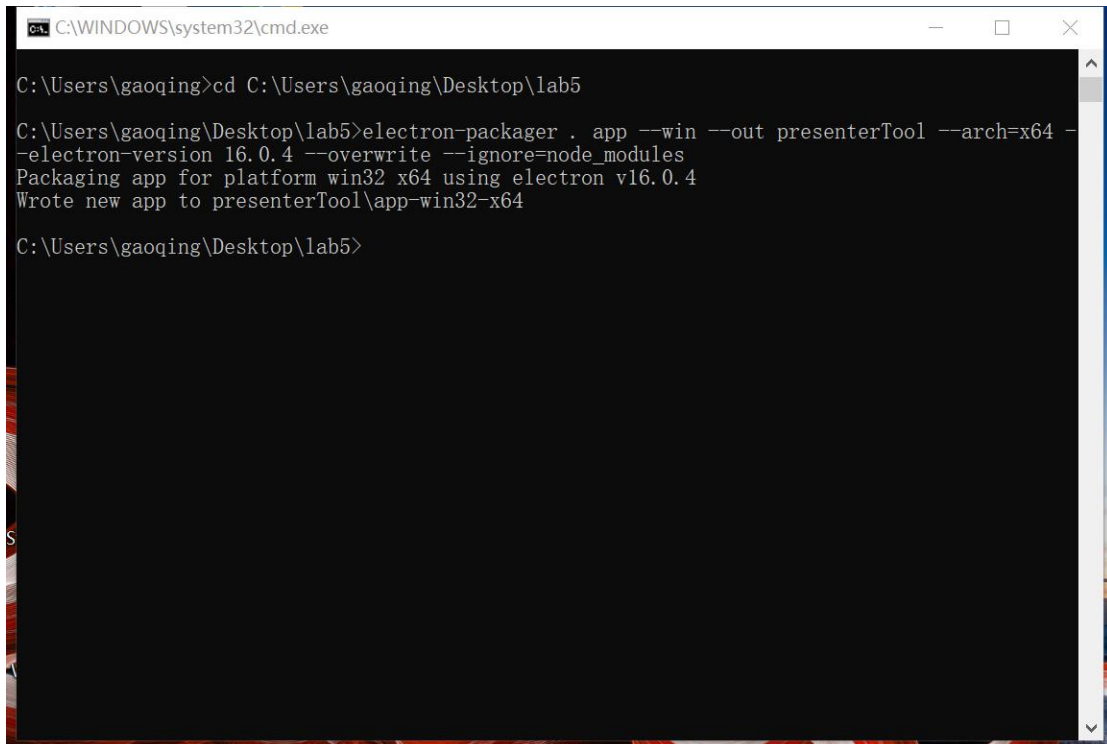
  // Turn on debugging mode
  mainWindow.webContents.openDevTools();
}

// Listen for application startup events
app.on("ready", createWindow_tips);
// Windows closing function compatible with MacOS system
app.on("window-all-closed", () => {
  // Non MacOS direct exit
  if (process.platform !== "darwin") {
    app.quit();
  }
});

// Restart the window when click the menu at the bottom of MacOS
app.on("activate", () => {
  if (BrowserWindow.getAllWindows().length == 0) {
    createWindow_tips();
  }
})

```

g. Package with “ electron-packager . app --win --out
presenterTool --arch=x64 --electron-version 16.4`0.4
--overwrite --ignore=node_modules” ,
electron-packager . [The file name of the executable] --win --out
[Folder name packaged into] --arch=x64--electron-version [version
number]--overwrite --ignore=node_modules

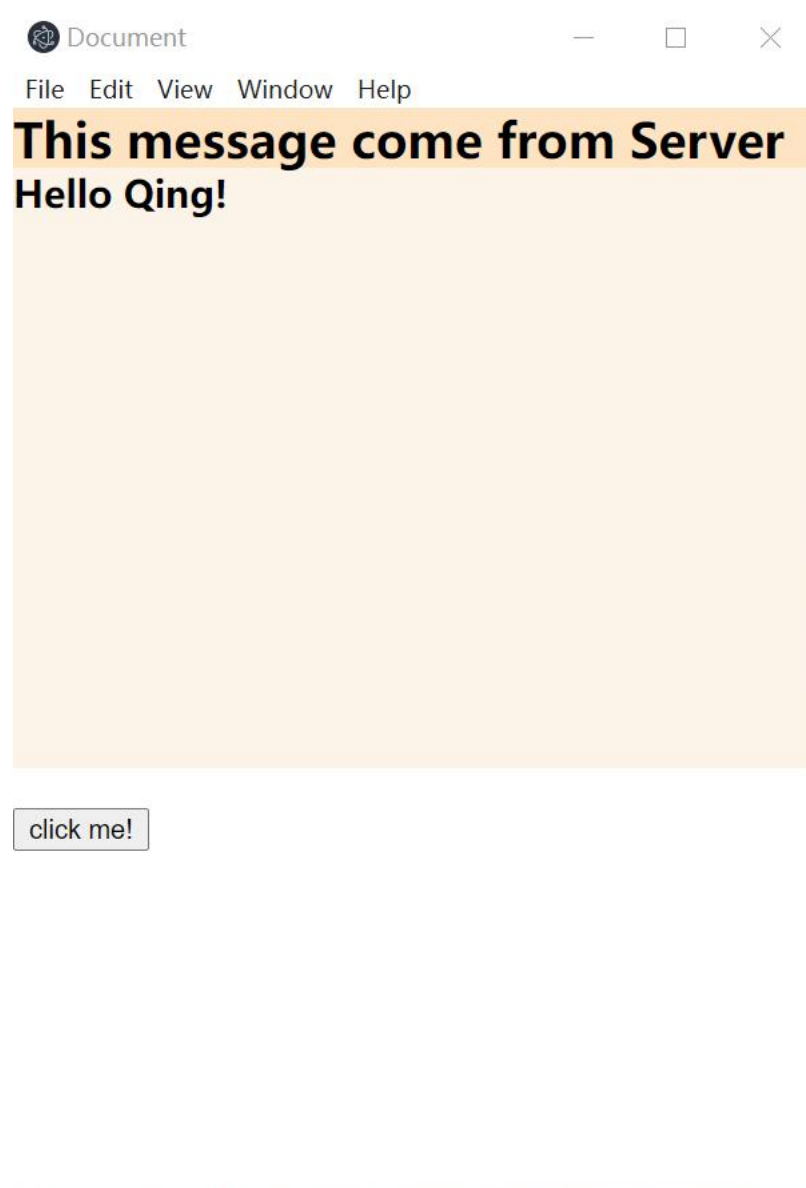
A screenshot of a Windows command prompt window. The title bar shows the path "C:\WINDOWS\system32\cmd.exe". The command prompt shows the following sequence of commands and output:
C:\Users\gaoqing>cd C:\Users\gaoqing\Desktop\lab5
C:\Users\gaoqing\Desktop\lab5>electron-packager . app --win --out presenterTool --arch=x64 --electron-version 16.0.4 --overwrite --ignore=node_modules
Packaging app for platform win32 x64 using electron v16.0.4
Wrote new app to presenterTool\app-win32-x64
C:\Users\gaoqing\Desktop\lab5>
The window has a standard Windows interface with minimize, maximize, and close buttons in the title bar.

h. Find app

名称	修改日期	类型	大小
.vscode	2021/12/6 18:10	文件夹	
node_modules	2021/12/6 18:20	文件夹	
presenterTool	2021/12/7 9:54	文件夹	
client	2021/12/7 9:17	JavaScript 文件	
index	2021/12/7 9:20	Microsoft Edge HT...	
index-1	2021/12/7 9:17	Microsoft Edge HT...	
main	2021/11/28 23:02	C++ 源文件	
main	2021/12/7 9:28	JavaScript 文件	
main.wasm	2021/11/28 14:02	WASM 文件	
main-1	2021/12/5 22:10	JavaScript 文件	
package	2021/12/6 18:20	JSON 源文件	
package-lock	2021/12/6 18:20	JSON 源文件	
server	2021/11/28 22:59	JavaScript 文件	

名称	修改日期	类型	
locales	2021/12/7 9:54	文件夹	
resources	2021/12/7 9:54	文件夹	
swiftshader	2021/12/7 9:54	文件夹	
app	2021/12/7 9:54	应用程序	
chrome_100_percent.pak	2021/12/7 9:54	PAK 文件	
chrome_200_percent.pak	2021/12/7 9:54	PAK 文件	
d3dcompiler_47.dll	2021/12/7 9:54	应用程序扩展	
ffmpeg.dll	2021/12/7 9:54	应用程序扩展	
icudtl.dat	2021/12/7 9:54	DAT 文件	
libEGL.dll	2021/12/7 9:54	应用程序扩展	
libGLSv2.dll	2021/12/7 9:54	应用程序扩展	
LICENSE	2021/12/7 9:54	文件	
LICENSES.chromium	2021/12/7 9:54	Microsoft Edge HT...	
resources.pak	2021/12/7 9:54	PAK 文件	
snapshot_blob.bin	2021/12/7 9:54	BIN 文件	
v8_context_snapshot.bin	2021/12/7 9:54	BIN 文件	
version	2021/12/7 9:54	文件	
vk_swiftshader.dll	2021/12/7 9:54	应用程序扩展	
vk_swiftshader_icd	2021/12/7 9:54	JSON 源文件	
vulkan-1.dll	2021/12/7 9:54	应用程序扩展	

- i. The app is started successfully, and the communication between server and client is realized



```
C:\Users\gaoqing\Desktop\lab5>node server
Time: 551
Hello server! I am Gao~
```

- 2. The new function of notification is realized through electron

