

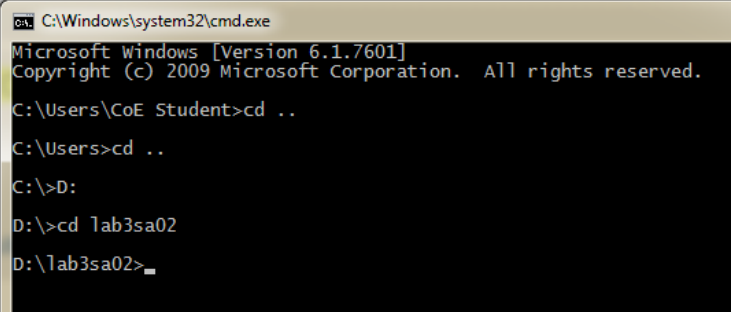
Lab 3SA02: Cross Platform Desktop Application Development

วิธีการทำลอง

1. ติดตั้ง **Electron** โดยใช้คำสั่ง “ **npm install -g electron** ” ผ่านหน้าต่าง **Command Prompt(cmd)**
2. สร้างโฟลเดอร์สำหรับเก็บไฟล์ **Project**
3. เปิดหน้าต่าง **Command Prompt(cmd)** แล้วเข้าไปยังโฟลเดอร์ที่สร้างไว้ แล้วพิมพ์คำสั่ง

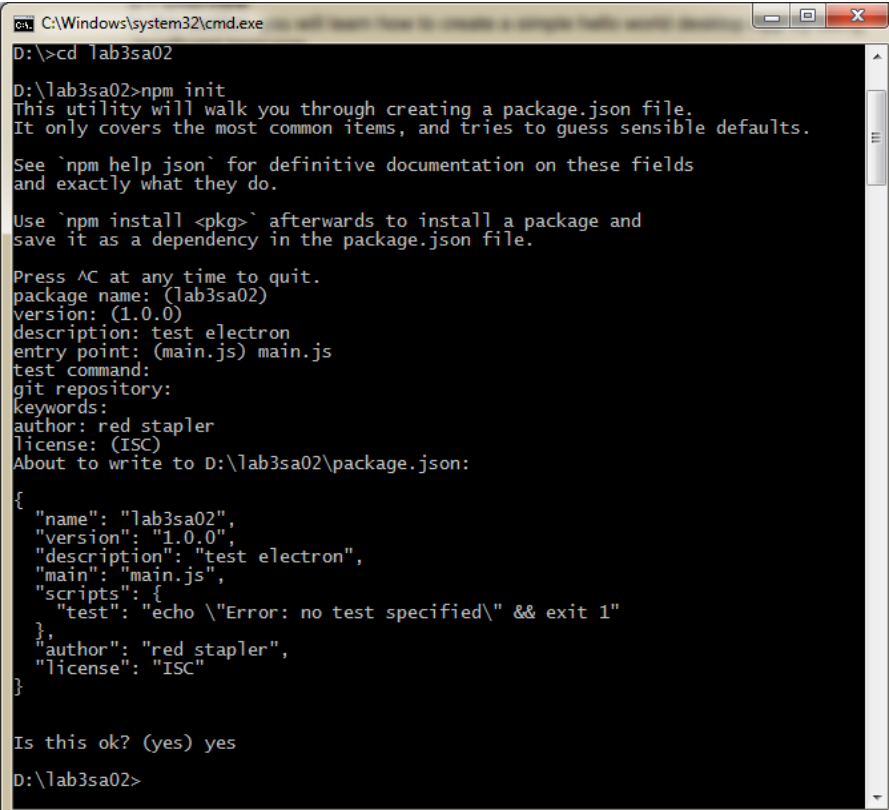
“ **npm init** ” เพื่อสร้าง **Project** โดยกำหนดรายละเอียดตามที่ต้องการ โดยในที่นี้จะระบุรายละเอียดดังนี้

- Package name: (lab3sa02)
- Version: (1.0.0)
- description: test electron
- entry point: (main.js) main.js
- test command:
- git repository:
- keywords:
- author: red stapler
- license: (ISC)



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\CoE Student>cd ..
C:\Users>cd ..
C:\>D:
D:\>cd lab3sa02
D:\lab3sa02>_
```



```
C:\Windows\system32\cmd.exe
D:\>cd lab3sa02
D:\lab3sa02>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

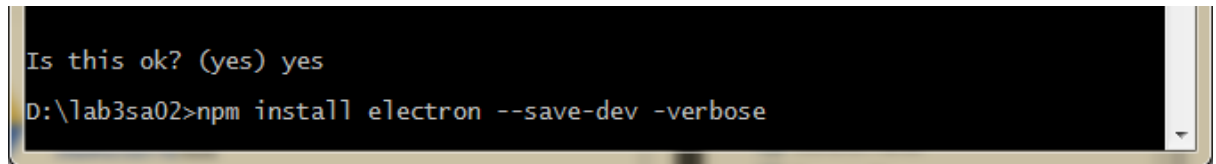
See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (lab3sa02)
version: (1.0.0)
description: test electron
entry point: (main.js) main.js
test command:
git repository:
keywords:
author: red stapler
license: (ISC)
About to write to D:\lab3sa02\package.json:
{
  "name": "lab3sa02",
  "version": "1.0.0",
  "description": "test electron",
  "main": "main.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "red stapler",
  "license": "ISC"
}

Is this ok? (yes) yes
D:\lab3sa02>
```

4. จากนั้นพิมพ์คำสั่ง “**npm install electron --save-dev --verbose**” เพื่อติดตั้ง **Electron** เข้าสู่โฟลเดอร์ ของ **Project** ที่สร้างไว้ก่อนหน้านี้



```
Is this ok? (yes) yes
D:\lab3sa02>npm install electron --save-dev --verbose
```

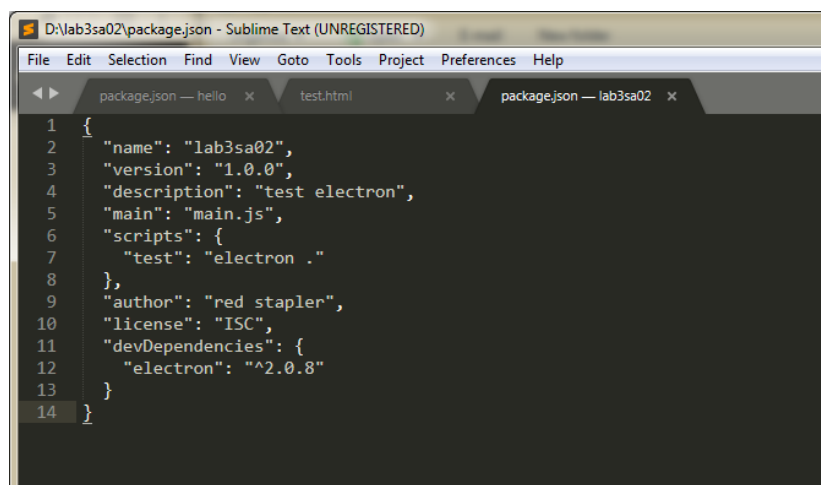
5. สร้างไฟล์ **main.js** ขึ้นมาเพื่อเป็นการจัดการคำสั่งและเรียกหน้าเว็บไซต์โดยเรียกใช้ไฟล์ **test.html**



```
main.js - Visual Studio Code
File Edit Selection View Go Debug Tasks Help

1 const electron = require('electron');
2 const app = electron.app;
3 const path = require('path');
4 const url = require('url');
5 const BrowserWindow = electron.BrowserWindow;
6 var mainWindow;
7 app.on('ready',function(){
8   mainWindow = new BrowserWindow({width: 1024, height: 768,backgroundColor: '#2e2c29'});
9   //mainWindow.loadURL('https://github.com');
10  mainWindow.loadURL(url.format({
11    pathname: path.join(__dirname, 'test.html'),
12    protocol: 'file:',
13    slashes: true
14  }));
15 });
```

6. แก้ไขไฟล์ **package.json** ในส่วน **scripts** เพื่อให้เรียกใช้ **Electron** ได้ดังนี้

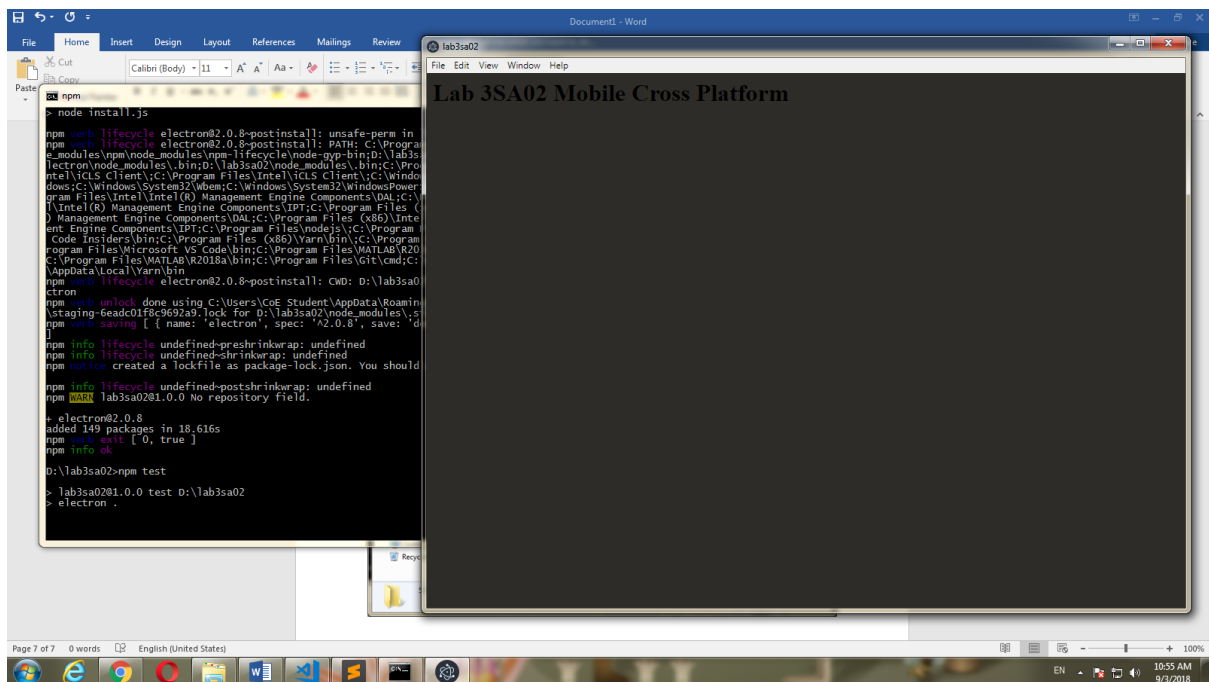


```
D:\lab3sa02\package.json - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

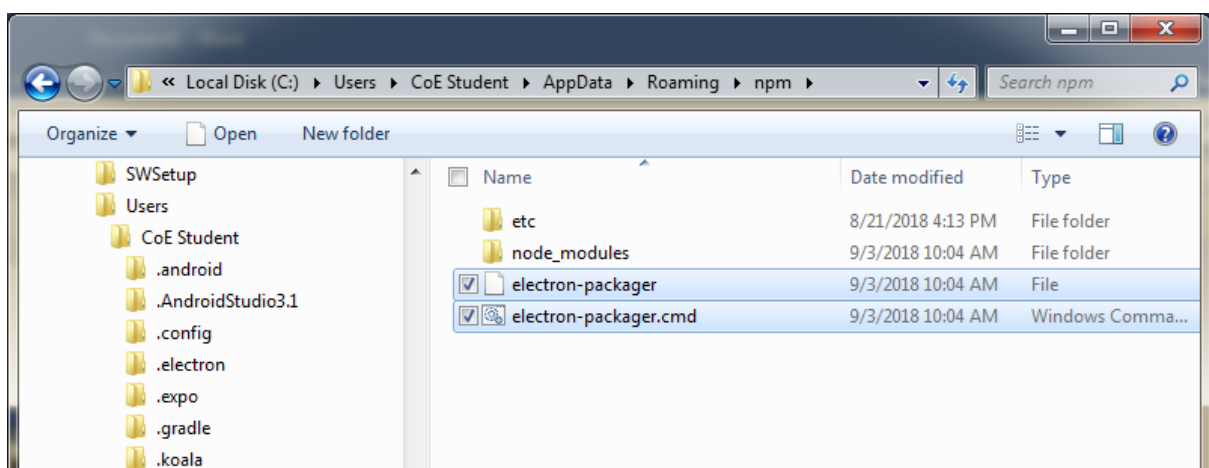
package.json — hello x test.html x package.json — lab3sa02 x

1 {
2   "name": "lab3sa02",
3   "version": "1.0.0",
4   "description": "test electron",
5   "main": "main.js",
6   "scripts": {
7     "test": "electron ."
8   },
9   "author": "red stapler",
10  "license": "ISC",
11  "devDependencies": {
12    "electron": "^2.0.8"
13  }
14 }
```

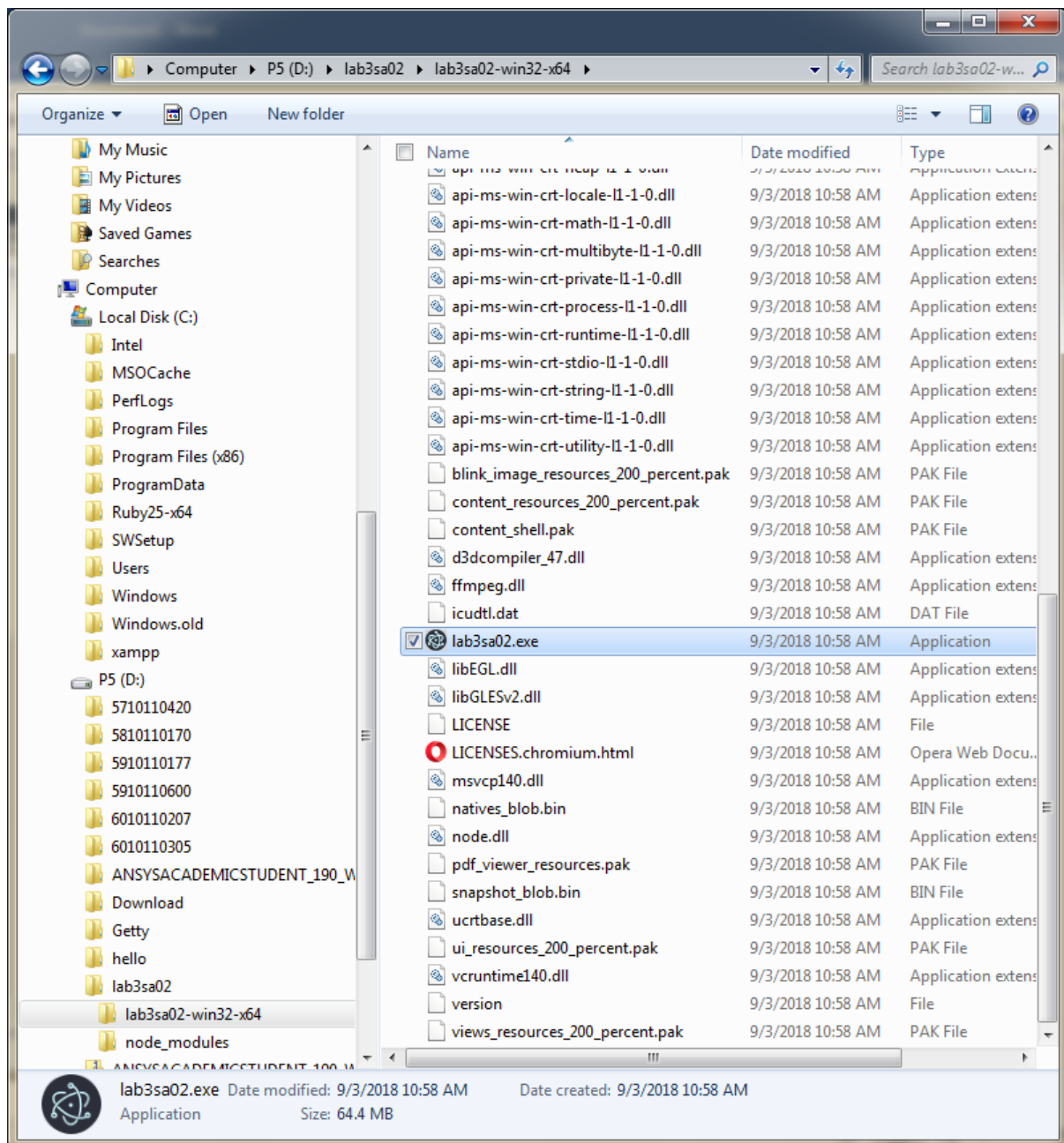
7. ทดลองเรียกใช้ **Electron** โดยพิมพ์คำสั่ง “**npm test**” ที่หน้าต่าง **Command Prompt(cmd)** จะได้หน้าตาเว็บไซต์ดังนี้



8. เมื่อทำการปรับแต่ง **Project** เสร็จแล้ว เราสามารถ **packaging** ไฟล์ต่างๆ เป็น **executable file** เพื่อที่จะให้ผู้ใช้สามารถใช้งานได้สะดวก โดยจะต้องติดตั้งตัว **Electron-packager** ในที่จะ **copy** ไฟล์มาจากโฟลเดอร์ **npm** แต่ผู้ใช้สามารถติดตั้งได้ผ่านหน้าต่าง **Command Prompt(cmd)** โดยใช้คำสั่ง “**npm install electron-packager**”



9. **Packaging** ด้วยคำสั่ง “**electron packager .**” แล้วจะได้ไฟล์นามสกุล **.exe** ซึ่งสามารถดับเบิลคลิกเพื่อเรียกใช้ได้ทันที



คำถามหลังการทดลอง

- Describe how your Electron app works (Note: see reference [1] on the first page) and according to the Fig. 2, answer the following questions in your just-finished experiment :

- ☐ The main process is in file(s): **main.js**
- ☐ The render process is in file(s): **test.html**

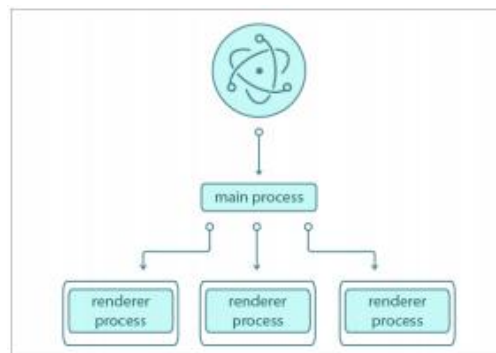


Fig. 2 How Electron App Works

ตอบ Electron เป็น framework ที่เอาไว้ทำ desktop application โดย Electron จะแบ่งออกเป็น 2 ส่วนใหญ่ๆคือ Main Process และ Renderer Process โดยที่ Main Process คือตัว Core หลักของ Electron โดยมันจะจัดการ Application ของเราโดยการสร้าง BrowserWindow Instances ขึ้นมาและในแต่ละ Instance จะทำงานโดย Renderer Process ของตัวเอง หมายความว่าถ้าหน้าต่างไหนถูกปิดไป Renderer Process นั้นๆก็จะถูก Terminated ไปด้วย แต่ Main Process จะยังคงทำงานอยู่เพื่อจัดการกับ Instances อื่นๆ

- Explain why the Electron tool can give benefits to your programming in the future.

ตอบ เนื่องจาก Electron สามารถรองรับการทำงานเข้ากับ platform ต่างๆได้เกือบทุก platform บน desktop application ซึ่งมันอาจจะเป็นทางเลือกในการแก้ปัญหาจากการที่โปรแกรมไม่สามารถรันบนระบบปฏิบัติการรุ่นใหม่ๆได้ ทำให้เป็นตัวเลือกในการประกอบการตัดสินใจที่จะนำมาใช้ในการพัฒนาซอฟต์แวร์ได้