

LAB 6: GENERATING INSTALLERS

In this exercise, you will:

- Electron-builder installation and configuration.
- Generate installer for macOS.
- Generate installer for windows.

1. Check out the `Lab-Generating-Installers` branch from the remote repository.

```
| git checkout origin/Lab-Generating-Installers
```

Note: The HEAD of this branch provides the completed code for the entire exercise.

2. Run `npm install` to ensure all the dependencies have been installed.
3. Create a new local branch for your work based on the "starting point..." commit.

```
| git log --oneline  
| git checkout <hash for starting point commit>  
| git checkout -b Lab-Electron-Generating-Installers-mine
```

4. Start the app.

```
| npm run start
```

Electron-builder Installation and configuration

5. Install electron-builder

```
| npm install --save-dev electron-builder
```

6. Create build field in `package.json` (`package.json` is present at the root level of project).

```
"build": {  
  
  "appId": "com.dev6.electron",  
  
  "mac": {
```

```
    "category": "public.app-category.educational"

  },

  "dmg": {

    "contents": [

      {

        "x": 110,

        "y": 150

      },

      {

        "x": 240,

        "y": 150,

        "type": "link",

        "path": "/Applications"

      }

    ]

  },

  "linux": {

    "target": [

      "AppImage",

      "deb"

    ]

  }
```

```
    },  
    "win": {  
      "target": [  
        {  
          "target": "nsis",  
          "arch": [  
            "ia32"  
          ]  
        }  
      ],  
      "icon": "build/icon.ico"  
    }  
  },  
},
```

Note: To refer the actual placement of build field in package.json file. Please refer to below screenshot.

```

{
  "name": "electron-shirtastic-app",
  "version": "1.0.0",
  "description": "Electron Bootcamp Demo",
  "main": "main.js",
  "scripts": {
    "start": "electron .",
    "pack": "electron-builder --pack",
    "postinstall": "electron-builder install-app-deps",
    "buildWin": "electron-builder --win",
    "buildMac": "electron-builder --mac"
  },
  "keywords": [],
  "author": "gagank@dev6.com",
  "license": "ISC",
  "build": {
    "appId": "com.dev6.electron",
    "mac": {
      "category": "public.app-category.educational"
    },
    "dmg": { ... },
    "linux": { ... },
    "win": { ... }
  },
  "devDependencies": {
    "devtron": "^1.4.0",
    "electron": "4.0.1",
    "electron-builder": "^20.38.5",
    "electron-reload": "^1.4.0"
  },
  "dependencies": {
    "axios": "^0.18.0",
    "electron-is-dev": "^1.0.1",
    "electron-log": "^3.0.1",
    "json-server": "^0.14.2"
  }
}

```

7. Add couple of new entries in scripts field.

pack script only generates the package directory without packaging it.
postinstall script will ensure your native dependencies are always matched electron version.

buildWin script will generate the distributables for window platform.

buildMac script will generate the distributables for macOS platform.

```
"scripts": {  
  
  "start": "electron .",  
  
  "pack": "electron-builder --pack",  
  
  "postinstall": "electron-builder install-app-deps",  
  
  "buildwin": "electron-builder --win",  
  
  "buildMac": "electron-builder -mac -c.mac.identity=null"  
  
},
```

8. Restart the app

| npm run start

Generate installer for macOS

9. Now it is time to generate distributable (installer) for macOS. Run below command on terminal

| npm run buildMac

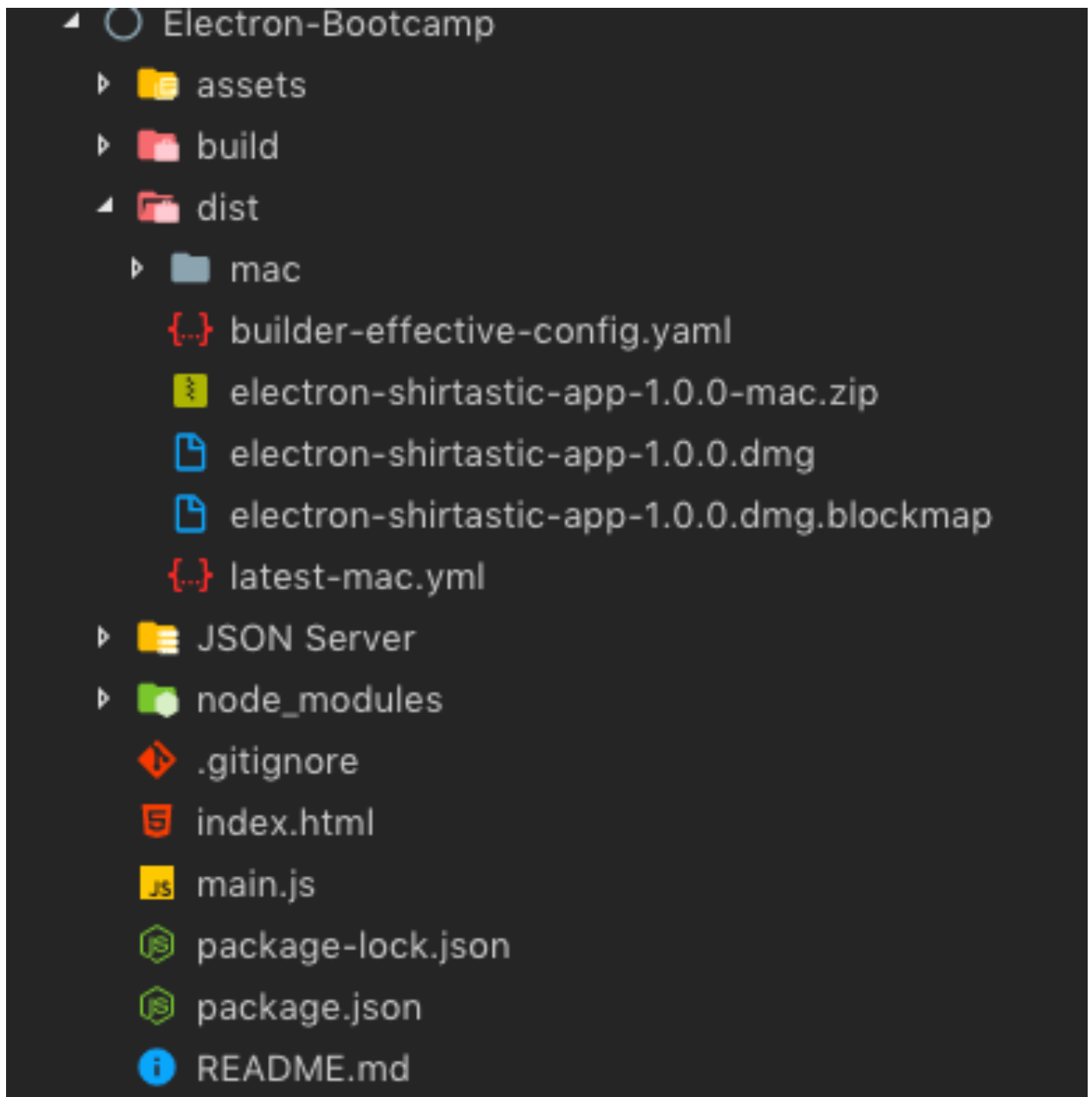
```
Gagan-Kaurs-MacBook-Pro:Electron-Bootcamp gagan_kaur$ npm run buildMac  
  
> electron-shirtastic-app@1.0.0 buildMac /Users/gagan_kaur/Projects/electron/Electron-Bootcamp  
> electron-builder --mac -c.mac.identity=null  
  
• electron-builder version=20.38.5  
• loaded configuration file=package.json ("build" field)  
• writing effective config file=dist/builder-effective-config.yaml  
• no native production dependencies  
• packaging platform=darwin arch=x64 electron=4.0.1 appOutDir=dist/mac  
• skipped macOS code signing reason=identity explicitly is set to null  
• building target=macOS zip arch=x64 file=dist/electron-shirtastic-app-1.0.0-mac.zip  
• building target=DMG arch=x64 file=dist/electron-shirtastic-app-1.0.0.dmg  
• building block map blockMapFile=dist/electron-shirtastic-app-1.0.0.dmg.blockmap  
• building embedded block map file=dist/electron-shirtastic-app-1.0.0-mac.zip
```

Note: We have passed an extra flag for buildMac script `-c.mac.identity=null`. By doing so you can skip the code signing process for build purpose on macOS. However, for distribution of electron app on Mac App Store you have to sign the app. Below is the screenshot for code signing build process logs. You can read more about how to generate code certificates by visiting this URL <https://www.electron.build/code-signing> and <https://www.electron.build/code-signing#where-to-buy-code-signing-certificate>.

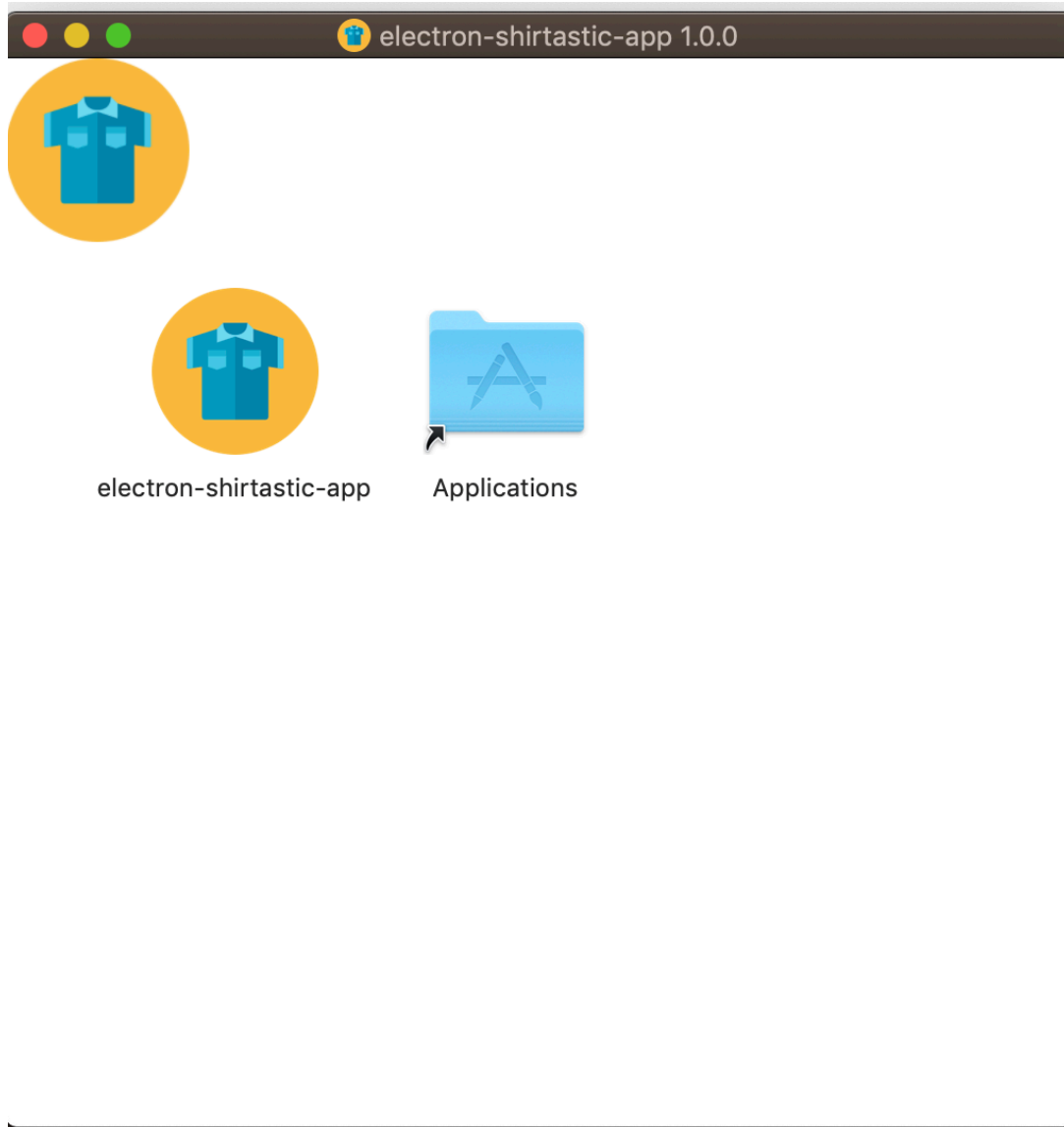
```
Gagan-Kaurs-MacBook-Pro:Electron-Bootcamp gagan_kaur$ npm run buildMac
> electron-shirtastic-app@1.0.0 buildMac /Users/gagan_kaur/Projects/electron/Electron-Bootcamp
> electron-builder --mac

  • electron-builder version=20.38.5
  • loaded configuration file=package.json ("build" field)
  • writing effective config file=dist/builder-effective-config.yaml
  • no native production dependencies
  • packaging platform=darwin arch=x64 electron=4.0.1 appOutDir=dist/mac
  • Mac Developer is used to sign app - it is only for development and testing, not for production
  • signing file=dist/mac/electron-shirtastic-app.app identityName=Mac Developer: gagank@dev6.com identityHash=
  • provisioningProfile=none
  • building target=macOS zip arch=x64 file=dist/electron-shirtastic-app-1.0.0-mac.zip
  • building target=DMG arch=x64 file=dist/electron-shirtastic-app-1.0.0.dmg
  • building block map blockMapFile=dist/electron-shirtastic-app-1.0.0.dmg.blockmap
  • building embedded block map file=dist/electron-shirtastic-app-1.0.0-mac.zip
```

10. The result of above command will be a dist folder is generated at the root level of project which contains the distributable. Below is the screenshot for reference to cross check the location of dist folder.



11. You can test the application by clicking on the dmg file. It will produce below screen to appear. Now you can drag the electron-shirtastic-app to Applications folder. And, it will start appearing in your launchpad. Please verify it by going to Launchpad.

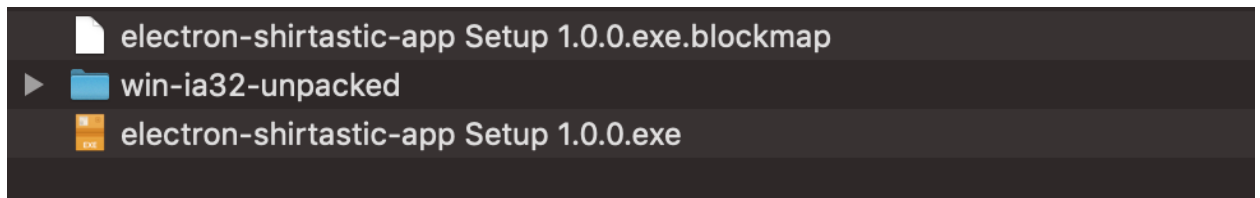


Generate installer for windows

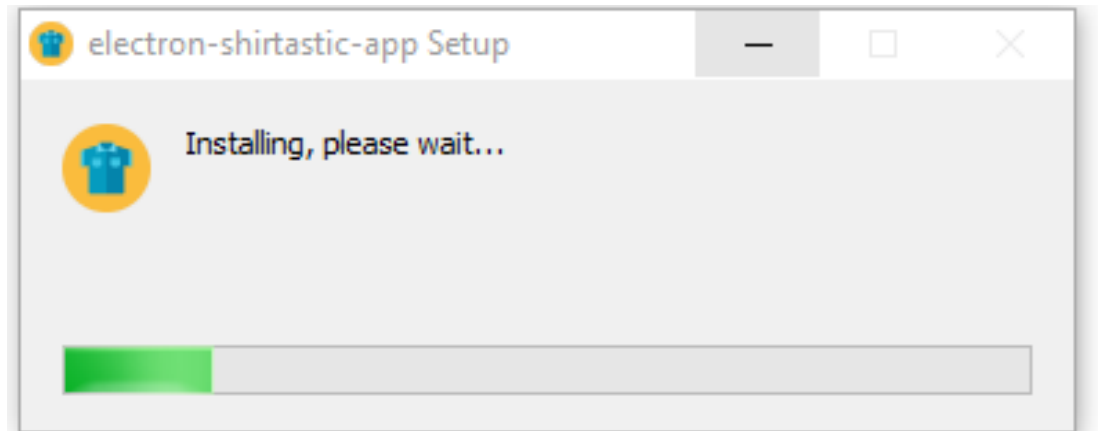
12. To generate the installer for windows, run the following command.

```
| npm run buildWin
```

13. It will generate below artifacts inside dist folder.



14. You can test the application by clicking on exe file. As soon you click on it, installation will start for windows.



Note: You can read more about how to do multi-platform build by visiting this URL <https://www.electron.build/multi-platform-build>.
