# LAB 7: AUTOMATIC UPDATES

In this exercise, you will:

- Set up http-server for hosting updates.
- Configure build field for making use of electron-builder to consume updates.
- Learn how to test auto-updates on macOS.
- Learn how to test auto-updates on windows.

1. Check out the `Lab-Silent-Updates` branch from the remote repository.

```
git checkout origin/Lab-Silent-Updates
```

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-Silent-Updates-mine
```

4. Start the app.

```
npm run start
```

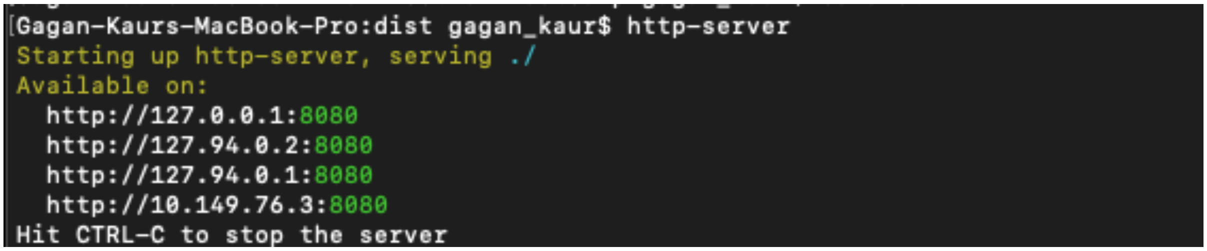
### Set up http-server for hosting updates.

5. Install http-server

```
npm install --save http-server
```

6. Navigate to dist folder in a new terminal window. And run below command.

```
http-server
```



```
[Gagan-Kaurs-MacBook-Pro:dist gagan_kaur$ http-server
Starting up http-server, serving ./
Available on:
  http://127.0.0.1:8080
  http://127.94.0.2:8080
  http://127.94.0.1:8080
  http://10.149.76.3:8080
Hit CTRL-C to stop the server
```

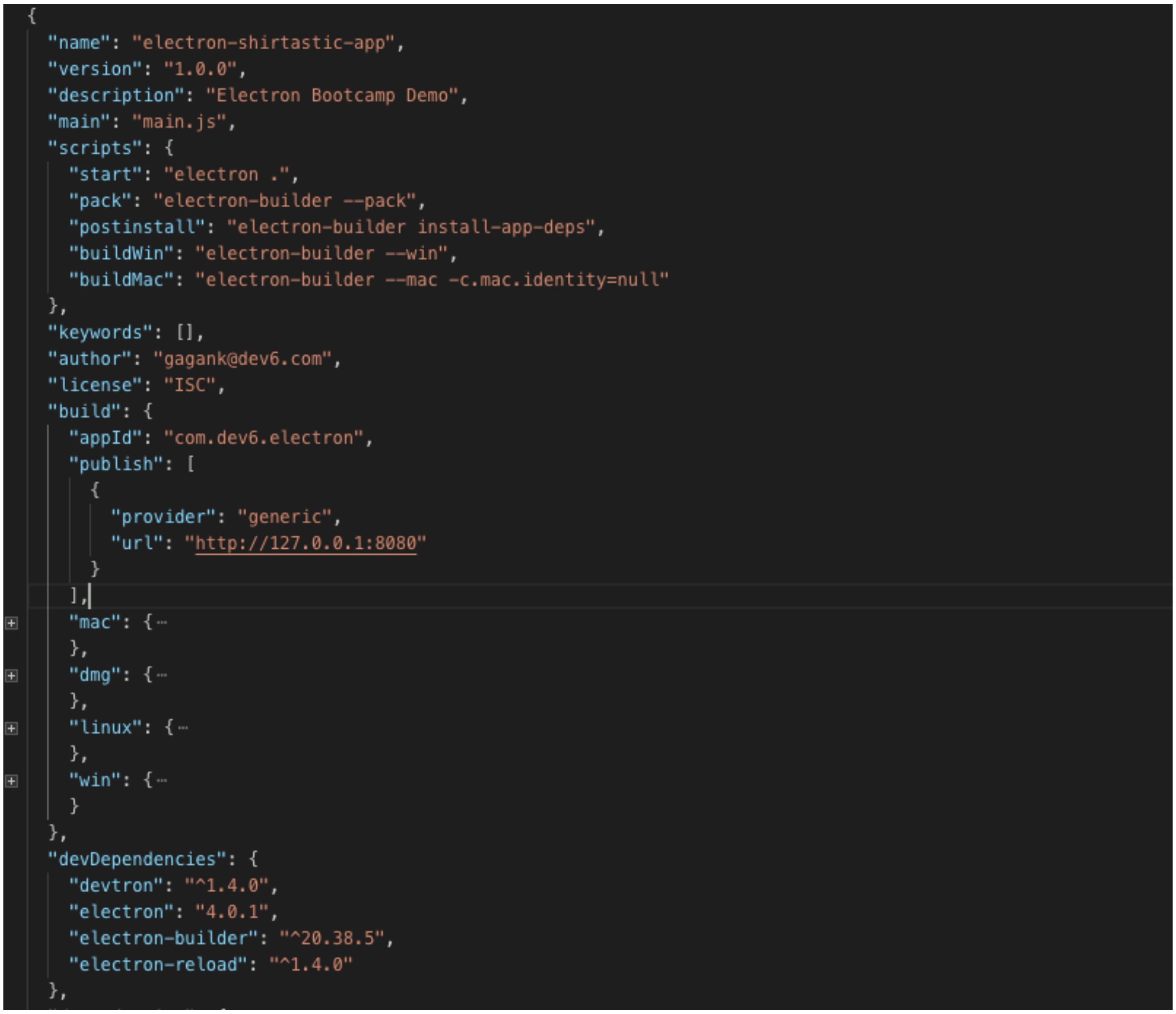
Note: Now you can check http://127.0.0.1.8080 on your browser. It will list the contents of dist folder.

## Configure electron-builder's build field in order to consume updates.

7. Open package.json and paste below lines of code inside build field.

```
"publish": [

      {

        "provider": "generic",

        "url": "http://127.0.0.1:8080"

      }

    ],
```

Note: To cross-check the actual placement of this line of code, please refer 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 -c.mac.identity=null"
  },
  "keywords": [],
  "author": "gagank@dev6.com",
  "license": "ISC",
  "build": {
    "appId": "com.dev6.electron",
    "publish": [
      {
        "provider": "generic",
        "url": "http://127.0.0.1:8080"
      }
    ],
    "mac": {…
    },
    "dmg": {…
    },
    "linux": {…
    },
    "win": {…
    }
  },
  "devDependencies": {
    "devtron": "^1.4.0",
    "electron": "4.0.1",
    "electron-builder": "^20.38.5",
    "electron-reload": "^1.4.0"
  },
```

Note: Please refer to this URL for more configuration options
https://www.electron.build/auto-update

### Testing auto-updates on macOS

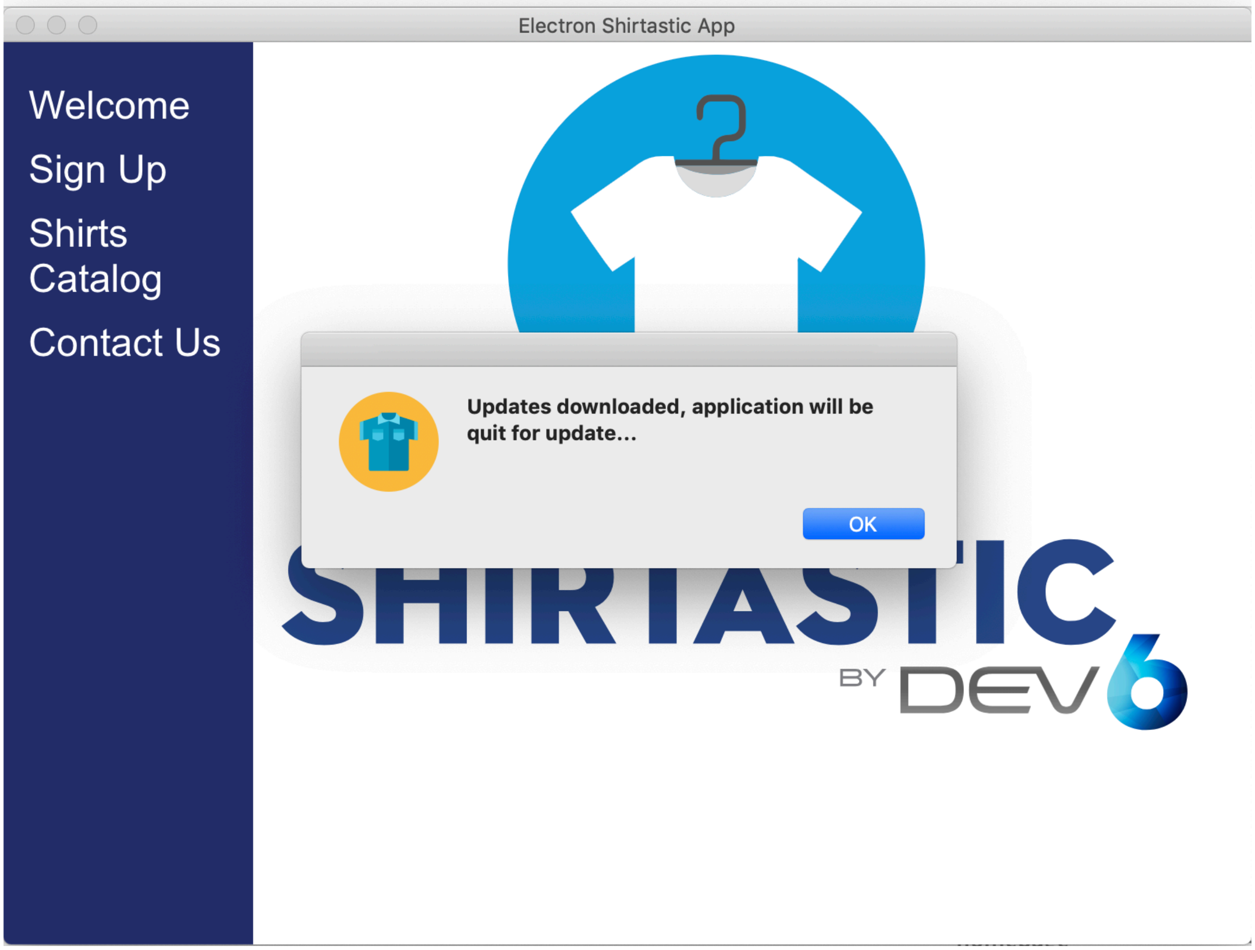
8. First create a new distributable by running below command by adding version 1.0.0 in package.json

```
npm run buildMac
```

9. Then run the dmg file and install the electron app.
10. Now go back to same terminal window and re-run the buildMac script, but this time change the version to 2.0.0

```
npm run buildMac
```

11. Now, re launch the app from launchpad, and as soon you do so, you will see a pop up like below screen.



Updates downloaded, application will be quit for update...

12. If you press on Ok, your application will be quit and relaunched with new version installed.

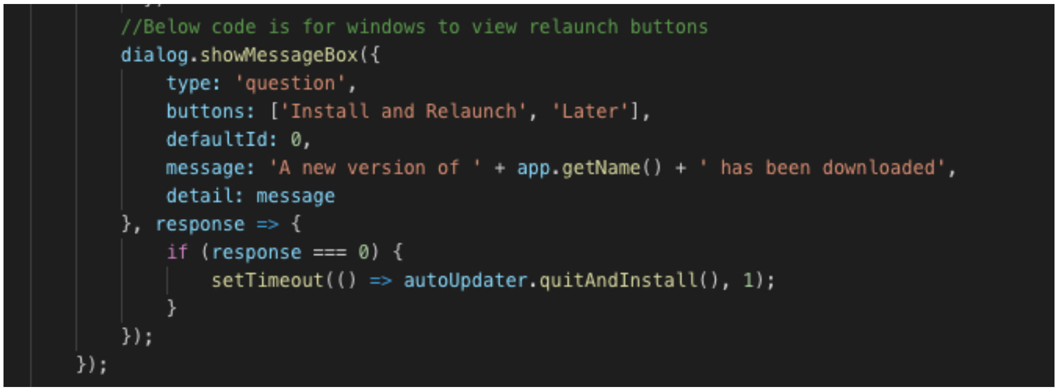
---

Note: If you closely observe, there is a dev-app-update.yml file at the root level. The purpose of this file is when you test updates in development mode, auto update module requires it internally to function smoothly.

---

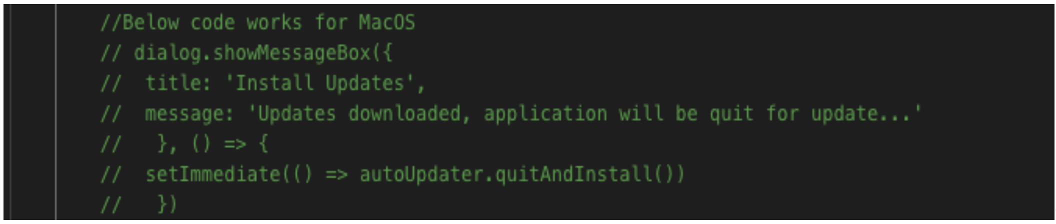
## Testing auto updates on Windows

13. Open autoupdater.js under assets/js folder. And uncomment the code mentioned in below screenshot.



```
        //Below code is for windows to view relaunch buttons
        dialog.showMessageBox({
            type: 'question',
            buttons: ['Install and Relaunch', 'Later'],
            defaultId: 0,
            message: 'A new version of ' + app.getName() + ' has been downloaded',
            detail: message
        }, response => {
            if (response === 0) {
                setTimeout(() => autoUpdater.quitAndInstall(), 1);
            }
        });
    });
```

14. And, comment this code.



```
        //Below code works for MacOS
        // dialog.showMessageBox({
        //  title: 'Install Updates',
        //  message: 'Updates downloaded, application will be quit for update...'
        //   }, () => {
        //  setImmediate(() => autoUpdater.quitAndInstall())
        //   })
```

s

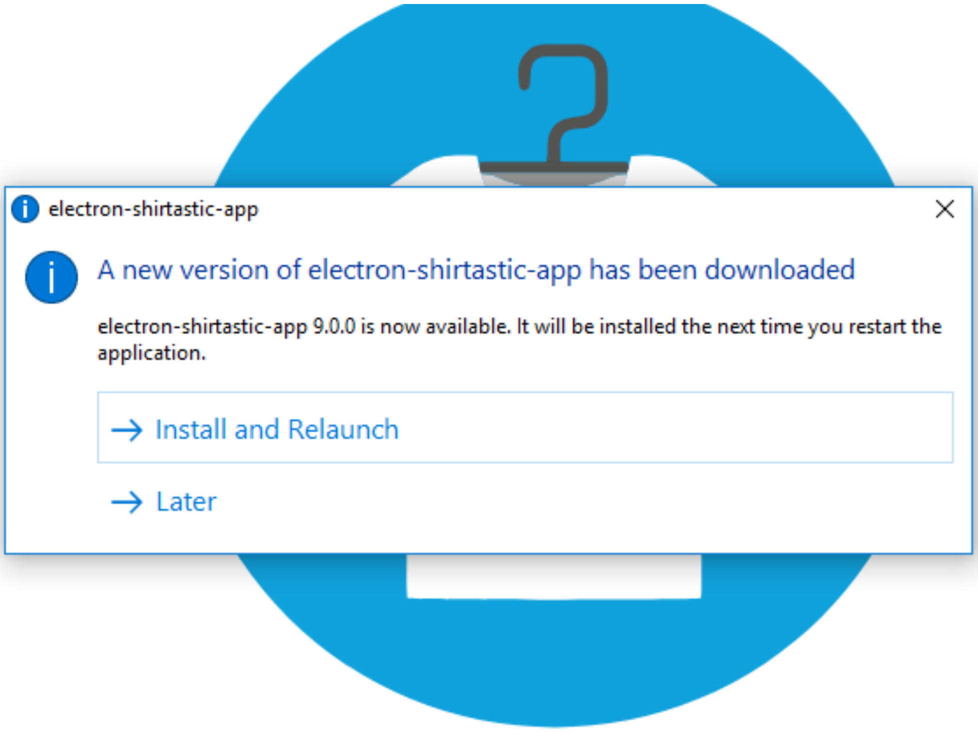
15. First create a new distributable by running below command by adding version 1.0.0 in package.json

```
npm run buildWin
```

16. Then run the exe file and install the electron app.
17. Now go back to same terminal window and re-run the buildWin script, but this time change the version to 2.0.0

```
npm run buildWin
```

18. Now, re launch the app from Application installed, and as soon you do so, you will see a pop up like below.



electron-shirtastic-app

A new version of electron-shirtastic-app has been downloaded

electron-shirtastic-app 9.0.0 is now available. It will be installed the next time you restart the application.

19. If you select Install and Relaunch, the application will be killed and restart with new version installed. And, if you select Later option, the update will be ignored, but it will download the next time when you restart the app.

Note: You can read more about the various options available by visiting this URL https://www.electron.build/auto-update.