⚠ You're currently viewing a free sample. [Start a free trial (/checkout/packt-subscription-monthly-launch-offer?freeTrial)](/checkout/packt-subscription-monthly-launch-offer?freeTrial) to access the full title and Packt library.

Configuring a new project

Let's start our journey by configuring a new Electron project and naming it `markdown-editor` since we are building a markdown editor application. You can create a corresponding folder with the following commands:

Copy

```
mkdir markdown-editor
cd markdown-editor
```

As you may recall from Chapter 1 (/book/mobile/9781838552206/1), *Building Your First Electron Application*, we need to initialize a new project with the `npm init` command. You should also install `electron`, the core library that provides an application shell. In addition, your project needs an `electron-builder` library, which allows you to publish and distribute features for multiple platforms. Let's get started:

1 Run the following commands to set up a new project:

Copy

```
npm init -y
npm i -D electron
npm i -D electron-builder
```

The `npm init` command should generate a `package.json` file with the following content:

Copy

```
{
  "name": "markdown-editor",
  "version": "1.0.0",
  "main": "index.js",
  "devDependencies": {
    "electron": "^7.0.0",
    "electron-builder": "^21.2.0"
  }
}
```

The `-D` switch means that the libraries should be installed in the `devDependencies` section.

2 Now, create an `index.js` file with a bare minimum amount of JavaScript code in it so that you can run an empty application:

Copy

```
const { app, BrowserWindow } = require('electron');

let window;

app.on('ready', () => {
    window = new BrowserWindow({
      width: 800,
      height: 600,
      webPreferences: {
        nodeIntegration: true
      }
    });
    window.loadFile('index.html');
});
```

3 Finally, create the `index.html` file, which is the HTML template for our new project, next to `index.js`. For now, just put in a dummy string as the content for the `body` element:

Copy

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
      initial-scale=1.0">
    <meta
      http-equiv="Content-Security-Policy"
      content="script-src 'self' 'unsafe-inline';"
    />
    <title>Document</title>
</head>
<body>
    <h1>Editor</h1>
</body>
</html>
```

At this point in time, our focus is on quickly setting up a project structure that we can turn into a markdown editor application.

4 The last piece of the puzzle lies in supporting the `npm start` script so that we can run and test our application without having to know all the command parameters and switches off by heart. Let's update the `package.json` file and extend the `scripts` section, as shown in the following code:

Copy

```json
{
  "name": "markdown-editor",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "start": "electron ."
  },
  "devDependencies": {
    "electron": "^7.0.0",
    "electron-builder": "^21.2.0"
  }
}
```

☐ Note

Note that your versions of the libraries may vary.

⁵ We are ready to create our Electron application. For all the updates to the files that we are going to create throughout this chapter, the testing process will run the following command:

Copy

```
npm start
```

Press *Ctrl + C* to stop the running application.

We will look at more project configuration options and live reloading in Chapter 3 (/book/mobile/9781838552206/3), *Integrating with Angular, React, and Vue*. For the time being, you should just be stopping the application with *Ctrl + C* and starting it with the `npm start` or `npm run start` command every time you change the code and want to see it live.

Now that the project is up and running, let's switch to the user interface and integrate the editor component with our Electron application.