



! 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.

Integrating the editor component

For our project, we don't need to build everything from scratch, including the components that edit and format text in markdown format. There are lots of free, open source components you can use to save time and focus on building the application and delivering value to your users, rather than spending months reinventing the wheel.

For the sake of simplicity, we are going to use the `SimpleMDE` component, which stands for **Simple Markdown Editor**. You can find more details about the project on its home page: <https://simplemde.com/> (<https://simplemde.com/>). The project is open source and has an MIT license. Follow these steps to incorporate the component:

- ¹ Similarly to how we installed the Electron framework itself, you can use NPM commands to get `SimpleMDE` into your project:

```
npm install simplemde
```

[Copy](#)

Don't forget to stop the application before installing a new library.

Like any other typical JavaScript component, the `SimpleMDE` component comes with a JavaScript file and a CSS stylesheet so that we can integrate with the web page.

- ² Append the following lines to the bottom of the `head` block in the `index.html` file:

```
<head>
  <link rel="stylesheet" href="./node_modules/simplemde/dist/
    simplemde.min.css">
  <script src="./node_modules/simplemde/dist/
    simplemde.min.js"></script>
</head>
```

[Copy](#)

Note how we are referencing the `node_modules` folder from the `index.html` page. Now, to update the `SimpleMDE` component to a newer version, you just need to run `npm install simplemde` once again. There's no need to update the web page each time; as soon as you build and run the application, it will use the updated libraries.

- ³ Now, let's run the newly installed component inside the Electron shell. According to the component's requirements, we need an empty `textarea` element defined on the page and a script block that turns that element into a markdown editor at runtime. Take a look at the following code, which shows a basic implementation of this:

```
<textarea id="editor"></textarea>

<script>
  var editor = new SimpleMDE({
    element: document.getElementById('editor')
  });
</script>
```

[Copy](#)

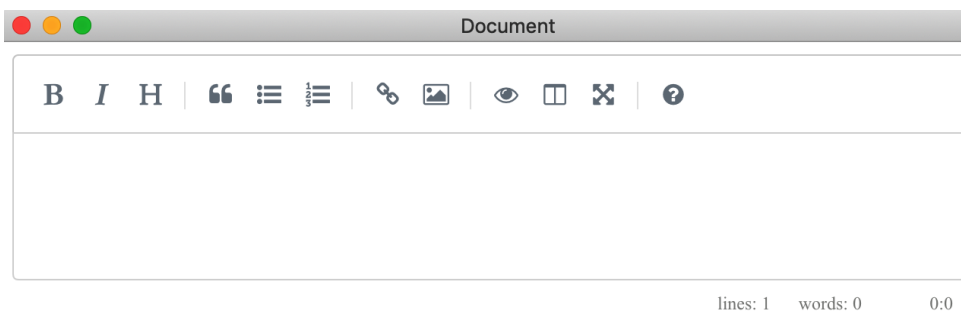
- ⁴ At this point, the content of the HTML page for your application should look as follows:

[Copy](#)

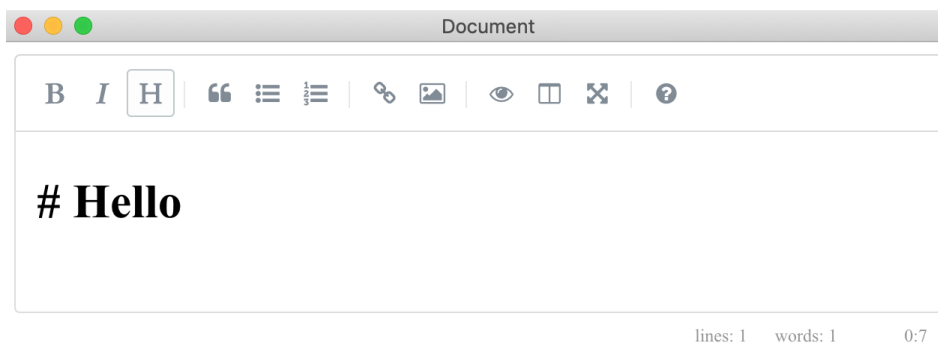
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
  <meta
    http-equiv="Content-Security-Policy"
    content="script-src 'self' 'unsafe-inline';"
  />
  <title>Document</title>
  <link rel="stylesheet" href="./node_modules/simplemde
    /dist/simplemde.min.css">
  <script src="./node_modules/simplemde/dist
    /simplemde.min.js"></script>
</head>
<body>
  <textarea id="editor"></textarea>
```

5 Save your changes and run the application.

6 You should see a window with the markdown editor component in the middle, a toolbar with a set of default buttons used to format the text, and word and line counter labels at the bottom:



The `SimpleMDE` component provides a nice live formatting feature so that your end users can see the final formatting alongside the markdown syntax. To see that in action, type something into the editor and press the `H` button on the toolbar. This turns the block into a `Heading` or `<h1>` element:



Feel free to experiment with the controls and see how they affect the text. Check the formatting options and Preview mode, which allows you to see what the final document looks like when it's rendered to HTML markup.

Note

There are many other options and features that you can enable or customize. Be sure to check out the documentation for **SimpleMDE** here:

<https://github.com/sparksuite/simplemde-markdown-editor>

(<https://github.com/sparksuite/simplemde-markdown-editor>). The out-of-the-box experience is pretty basic, and you may want to toggle additional settings later on.

One of the most important things you need to address is keeping your web application inside the Electron window. Let's see what it takes to match the content to the screen size.

◀ Previous Section (/book/mobile/9781838552206/2/ch02lvl1sec10/configuring-a-new-prc

Next Section ▶ (/book/mobile/9781838552206/2/ch02lvl1sec12/fitting-the-screen-size)
