

Shell Process

Transcrição

Para encerrar a construção da janela de **Sobre**, vamos melhorar o seu visual, adicionando um CSS, e acrescentar mais informações:

```
<!-- sobre.html -->

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Sobre o Alura Timer</title>
  <link rel="stylesheet" href="css/sobre.css">
</head>
<body>

  <div class="container">
    <p>O Alura timer é para você guardar seus tempos de estudo na Alura.</p>
    <p>Versão: 1.0.0</p>
    <p>Feito por Douglas Quintanilha</p>
    <p>Versão do Electron 1.6.1</p>
    <a href="#" id="link-fechar">Fechar</a>

  </div>
  <script src="js/sobre.js"></script>
</body>
</html>
```

Além disso, vamos colocar na página o ícone do Alura Timer, você pode baixá-lo [aqui](https://s3.amazonaws.com/caelum-online-public/electron/cap02/stages/icone-alura-timer.zip) (<https://s3.amazonaws.com/caelum-online-public/electron/cap02/stages/icone-alura-timer.zip>). Dentro de **app**, vamos criar a pasta **img**, e colocar a imagem dentro dessa pasta. Após isso, basta colocar a imagem na página:

```
<!-- sobre.html -->

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Sobre o Alura Timer</title>
  <link rel="stylesheet" href="css/sobre.css">
</head>
<body>

  <div class="container">
    
    <p>O Alura timer é para você guardar seus tempos de estudo na Alura.</p>
    <p>Versão: 1.0.0</p>
    <p>Feito por Douglas Quintanilha</p>
    <p>Versão do Electron 1.6.1</p>
    <a href="#" id="link-fechar">Fechar</a>

  </div>
  <script src="js/sobre.js"></script>
</body>
</html>
```

```

</div>
<script src="js/sobre.js"></script>
</body>
</html>

```

Falta criar o CSS, dentro da pasta `app/css/`, criamos o arquivo `sobre.css` com o seguinte conteúdo:

```

.container{
    text-align: center;
}

.container p {
    margin: 5px 0;
}

.container img{
    width: 50px;
}

```

No autor do aplicativo, podemos colocar um link para o seu Twitter:

```

<!-- sobre.html -->

<p>Feito por <a href="https://twitter.com/dquintanilhas">Douglas Quintanilha</a></p>

```

Mas ao testar e clicar no link para o Twitter, a página é aberta na própria janela de **Sobre**, isso acontece porque estamos literalmente dentro de uma janela de browser, já que as janelas do Electron são janelas de browser. Mas esse não é o comportamento que esperávamos, já que o certo é que quando clicarmos no link, o browser padrão do sistema operacional seja aberto com a página.

Então, o que podemos fazer é não colocar o endereço diretamente na página, e sim pedir para o Electron abrir o browser do sistema operacional, e a partir desse browser o Electron abrirá o endereço.

Abrir o browser padrão do sistema operacional com Electron

Mas como pediremos para o Electron abrir a página para nós? No `sobre.js`, precisamos escutar o evento de `click` do link, logo vamos adicionar o `id` para ele, e remover o endereço:

```

<!-- sobre.html -->

<p>Feito por <a href="#" id="link-twitter">Douglas Quintanilha</a></p>

```

E no `sobre.js`, selecionamos o link e escutamos seu evento de `click`:

```

// sobre.js

// código de fechar a janela omitido

let linkTwitter = document.querySelector("#link-twitter");

```

```
linkTwitter.addEventListener('click', function () {
});
```

Queremos abrir o browser do sistema operacional, e conseguimos fazer isso no próprio processo de *renderer*, através do `shell`, que vamos importar no nosso JavaScript:

```
// sobre.js

const { ipcRenderer, shell } = require('electron');

// código de fechar a janela omitido

let linkTwitter = document.querySelector("#link-twitter");

linkTwitter.addEventListener('click', function () {
});
```

O `shell` possui uma função chamada `openExternal`, que aceita receber uma URL! É ela que utilizaremos, passando o link para o Twitter:

```
// sobre.js

const { ipcRenderer, shell } = require('electron');

// código de fechar a janela omitido

let linkTwitter = document.querySelector("#link-twitter");

linkTwitter.addEventListener('click', function () {
    shell.openExternal("https://www.twitter.com/dquintanilhas");
});
```

Agora, ao clicar no link do Twitter, o browser padrão do nosso sistema operacional é aberto, exibindo a página!

Alterando dinamicamente a versão do Electron exibida na página

Por último, vamos alterar a versão do Electron exibida na página dinamicamente. Por que? Porque a versão do Electron pode ser atualizada, pode ser que mais para frente queiramos mostrar outras versões, não só a do Electron, como a do Node, do projeto, etc.

Para trocar a versão do Electron dinamicamente, primeiro devemos remover o número fixo que se encontra na página e adicionar um `span` com um `id`:

```
<!-- sobre.html -->

<p>Versão do Electron <span id="versao-electron"></span></p>
```

E para descobrir a versão do Electron, vamos utilizar o `process`, que é um módulo do Node, que nos entrega algumas informações do sistema. Primeiramente, vamos importá-lo e selecionar o `span` da versão do Electron:

```
// sobre.js

// restante do código omitido

const process = require('process');

let versaoElectron = document.querySelector('#versao-electron');
```

Agora, quando a janela carregar, ou seja, `window.onload`, nós alteramos o conteúdo de texto do `span`, ou seja, o seu o `textContent`, com a versão do Electron. Para a buscar a versão do Electron, acessamos a propriedade `versions` de `process`, acessando em seguida a propriedade `electron`, que nos retornará a sua versão:

```
// sobre.js

// restante do código omitido

const process = require('process');

let versaoElectron = document.querySelector('#versao-electron');

window.onload = function(){
    versaoElectron.textContent = process.versions.electron;
}
```

Podemos descobrir também a versão do Node, com `process.version`, e do browser que está rodando, com `process.versions.chrome`.