05

O menu de cursos

Transcrição

Agora que já entendemos como funciona o ***tray menu***, queremos criar os itens do menu a partir dos cursos que estão no diretório **alura-timer/data**, aqueles cursos que estão em arquivos JSON.

Para não ficar o código todo centralizado no **main.js**, vamos criar um módulo dentro da pasta **alura-timer**, responsável pela criação de *templates*, o **template.js**. Ao chamar a função desse módulo, será retornado o *template* pronto, e a partir desse *template*, criamos o menu:

```
// main.js

const { app, BrowserWindow, ipcMain, Tray, Menu }  = require('electron');
// importando o templateGenerator
const templateGenerator = require('./template')

app.on('ready', () => {
    console.log('Aplicacão iniciada');
    mainWindow = new BrowserWindow({
        width: 600,
        height: 400,
    });
    tray = new Tray(__dirname + '/app/img/icon-tray.png');

    // gerando template com o templateGenerator
    let template = templateGenerator.geraTrayTemplate();

    // criando o menu com o template gerado
    let trayMenu = Menu.buildFromTemplate(template);

    tray.setContextMenu(trayMenu);

    mainWindow.loadURL(`file://${__dirname}/app/index.html`);
});

// restante do código comentado
```

Agora, vamos criar o **template.js**, já exportando a função `geraTrayTemplate`:

```
// template.js

module.exports = {
    geraTrayTemplate() {

    }
}
```

Essa função deve gerar um *template* nos moldes que criamos no vídeo passado, ou seja, um array com vários objetos, cada um representando um curso. Vamos primeiramente criar os dois itens, o de cursos e o separador, e retornar o

template ao final da função:

```
// template.js

module.exports = {
    geraTrayTemplate() {
        let template = [
            { 'label': 'Cursos' },
            { type: 'separator' }
        ];

        return template;
    }
}
```

Ao testar, vemos que já conseguimos gerar um *template*, mas queremos que os cursos também apareçam no menu. E para isso precisamos saber onde estão esses cursos.

Se queremos criar um objeto para cada curso dentro do array do *template*, primeiro devemos selecionar todos os cursos que já estão criados, na pasta **alura-timer/data**. Então, o que podemos fazer é pedir para o módulo **data.js** buscar o nome de todos os cursos:

```
// template.js

const data = require('./data');

module.exports = {
    geraTrayTemplate() {
        let template = [
            { 'label': 'Cursos' },
            { type: 'separator' }
        ];

        // pegando o nome de todos os cursos
        let cursos = data.pegaNomeDosCursos();

        return template;
    }
}
```

Agora, no módulo **data.js**, nós criamos a função `pegaNomeDosCursos` :

```
// data.js

const jsonfile = require('jsonfile-promised');
const fs = require('fs');

module.exports = {
    salvaDados(curso, tempoEstudado) {
        // código omitido
    },
    adicionaTempoAoCurso(arquivoDoCurso, tempoEstudado) {
        // código omitido
```

```
    },
    criaArquivoDeCurso(nomeArquivo, conteudoArquivo) {
        // código omitido
    },
    pegaDados(curso) {
        // código omitido
    },
    // nova função pegaNomeDosCursos
    pegaNomeDosCursos() {

    }
}
```

Cada arquivo JSON dentro da pasta **data** possui o nome do seu respectivo curso. Então, para selecionar o nome dos cursos, podemos perguntar para o Node o nome de todos os arquivos dessa pasta, e depois nós removemos a extensão do nome desses arquivos (**.json**).

Selecionando os arquivos dos cursos

Como estamos trabalhando com o Node, podemos perguntar para ele o nome de todos os arquivos da pasta **data**, com o auxílio da função `readdirSync`, do *file system* (`fs`), passando para ela o diretório onde se encontram os arquivos:

```
// data.js

// restante do código omitido

pegaNomeDosCursos() {
    let arquivos = fs.readdirSync(__dirname + '/data/');
}
```

Com isso, temos um array com o nome de todos os arquivos da pasta **data**. Com isso em mãos, podemos criar uma variável `cursos` e atribuir a ela a função `map`, para mapear cada item do array e remover a extensão `.json` de cada um deles.

```
// data.js

// restante do código omitido

pegaNomeDosCursos() {
    let arquivos = fs.readdirSync(__dirname + '/data/');
    let cursos = arquivos.map((arquivo) => {

    });
}
```

Dentro da função, nós vamos retornar o arquivo sem a sua extensão. Para remover a extensão, podemos chamar a função `substr`, que corta um pedaço da string. Queremos cortar a string a partir do ponto, logo vamos passar o índice `0`, que significa o começo da string, e o índice do ponto, que descobriremos através da função `lastIndexOf`:

```
// data.js
```

```
// restante do código omitido

pegaNomeDosCursos() {
    let arquivos = fs.readdirSync(__dirname + '/data/');
    let cursos = arquivos.map((arquivo) => {
        return arquivo.substr(0, arquivo.lastIndexOf('.'));
    });
}
```

Assim, retornaremos o nome do arquivo do seu início (representado pelo índice 0) até o último ponto. E com o nome dos cursos em mãos, podemos retorná-los na função:

```
// data.js

// restante do código omitido

pegaNomeDosCursos() {
    let arquivos = fs.readdirSync(__dirname + '/data/');
    let cursos = arquivos.map((arquivo) => {
        return arquivo.substr(0, arquivo.lastIndexOf('.'));
    });

    return cursos;
}
```

Criando um objeto no template para cada curso

Com os cursos em mãos, temos que criar no *template* um objeto para cada curso. Para isso usaremos o `forEach`:

```
// template.js

const data = require('./data');

module.exports = {
    geraTrayTemplate() {
        let template = [
            { 'label': 'Cursos' },
            { type: 'separator' }
        ];

        // pegando o nome de todos os cursos
        let cursos = data.pegaNomeDosCursos();
        cursos.forEach((curso) => {
            let menuItem = {
                label: curso,
                type: 'radio'
            }
        });
        return template;
    }
}
```

Criamos um item, um objeto com `label` igual ao nome do curso e `type` *radio*, exatamente como havíamos criado no vídeo passado. Por fim, nós adicionamos o item ao *template*:

```
// template.js

const data = require('./data');

module.exports = {
    geraTrayTemplate() {
        let template = [
            { 'label': 'Cursos' },
            { type: 'separator' }
        ];

        // pegando o nome de todos os cursos
        let cursos = data.pegaNomeDosCursos();
        cursos.forEach((curso) => {
            let menuItem = {
                label: curso,
                type: 'radio'
            }
            template.push(menuItem);
        });
        return template;
    }
}
```

Ao executar a aplicação, os cursos aparecem corretamente no menu! Resta agora que, ao clicar em um curso, ele seja carregado na aplicação. Veremos isso no próximo vídeo.