

Exibindo o tempo salvo

Transcrição

Por último, falta que, quando abrirmos a aplicação, o tempo estudado do curso seja carregado.

Carregando o tempo salvo na inicialização da aplicação

Para isso, no **renderer.js**, nós importamos o módulo **data.js** e precisamos que, no momento em que a aplicação for aberta (`window.onload`), ele busque os dados do curso. Vamos chamar a função `pegaDados` e passar o curso por argumento:

```
// renderer.js

// restante do código comentado

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

window.onload = () => {
  data.pegaDados(curso.textContent);
}
```

Agora, no **data.js**, precisamos criar essa função:

```
// data.js

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

module.exports = {
  // outras funções omitidas
  pegaDados(curso) {

  }
}
```

Essa função irá até o arquivo do curso (o seu caminho completo), e o lerá através da função `readFile` do `jsonfile`, retornando essa *promise*:

```
// data.js

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

module.exports = {
  // outras funções omitidas
  pegaDados(curso) {
    let arquivoDoCurso = __dirname + '/data/' + curso + '.json';
    return jsonfile.readFile(arquivoDoCurso);
  }
}
```

```
    }  
  }  
}
```

Por fim, no **renderer.js**, precisamos lidar com essa *promise*. Então, encadeamos um `.then`, que assim que os dados forem lidos do JSON, eles serão retornados para nós na função:

```
// renderer.js  
  
// restante do código comentado  
  
const data = require('.././data');  
  
window.onload = () => {  
  data.pegadados(curso.textContent)  
    .then((dados) => {  
    })  
  }  
}
```

O retorno nada mais é do que um objeto, com as propriedades `ultimoEstudo` e `tempo`. Logo, basta atualizar o `textContent` de `tempo` para a propriedade do objeto retornado:

```
// renderer.js  
  
// restante do código comentado  
  
const data = require('.././data');  
  
window.onload = () => {  
  data.pegadados(curso.textContent)  
    .then((dados) => {  
      tempo.textContent = dados.tempo;  
    })  
  }  
}}
```

Ao abrir ou atualizar a aplicação, o tempo é carregado corretamente! Assim podemos dar continuidade aos nossos estudos exatamente do tempo que paramos.