

## Salvando o tempo de estudo

### Transcrição

Agora que sabemos que o arquivo já está sendo criado, vamos implementar a função responsável por salvar os dados dentro desse arquivo.

### Salvando o tempo estudado no arquivo do curso

No módulo `data.js`, vamos criar a função `adicionaTempoAoCurso`, que recebe o arquivo do curso e o seu tempo de estudo por parâmetro:

```
// data.js

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

module.exports = {
  salvaDados(curso, tempoEstudado) {
    let arquivoDoCurso = __dirname + '/data/' + curso + '.json';
    if(fs.existsSync(arquivoDoCurso)) {
      // Salva os dados
    } else {
      this.criaArquivoDeCurso(arquivoDoCurso, {})
        .then(() => {
          // Salva os dados
        });
    }
  },
  adicionaTempoAoCurso(arquivoDoCurso, tempoEstudado) {

  },
  criaArquivoDeCurso(nomeArquivo, conteudoArquivo){
    return jsonfile.writeFile(nomeArquivo, conteudoArquivo)
      .then(() => {
        console.log('Arquivo Criado')
      }).catch((err) => {
        console.log(err);
      });
  }
}
```

Nessa função, já sabemos o que fazer, faremos a mesma coisa que fizemos na função `criaArquivoDeCurso`, chamando a função `writeFile` do `jsonfile`:

```
// data.js

adicionaTempoAoCurso(arquivoDoCurso, tempoEstudado) {
  jsonfile.writeFile(arquivoDoCurso, tempoEstudado)
    .then(() => {
      console.log('Tempo salvo com sucesso');
    });
}
```

```

        }).catch((err) => {
            console.log(err);
        })
    }
}

```

Só que não vamos salvar apenas o tempo do curso e sim um objeto, que conterá o último horário que o curso foi estudado, além do tempo de estudo:

```

// data.js

adicionaTempoAoCurso(arquivoDoCurso, tempoEstudado) {
    let dados = {
        ultimoEstudo: new Date().toString(),
        tempo: tempoEstudado
    }
    jsonfile.writeFile(arquivoDoCurso, tempoEstudado)
        .then(() => {
            console.log('Tempo salvo com sucesso');
        }).catch((err) => {
            console.log(err);
        })
}

```

Agora basta salvar esse objeto no arquivo do curso, no lugar do tempo estudado:

```

// data.js

// restante do código comentado

adicionaTempoAoCurso(arquivoDoCurso, tempoEstudado) {
    let dados = {
        ultimoEstudo: new Date().toString(),
        tempo: tempoEstudado
    }
    jsonfile.writeFile(arquivoDoCurso, dados)
        .then(() => {
            console.log('Tempo salvo com sucesso');
        }).catch((err) => {
            console.log(err);
        })
}

```

E na função `salvaDados`, chamamos essa função;

```

// data.js

// restante do código comentado

salvaDados(curso, tempoEstudado) {
    let arquivoDoCurso = __dirname + '/data/' + curso + '.json';
    if(fs.existsSync(arquivoDoCurso)) {
        this.adicionaTempoAoCurso(arquivoDoCurso, tempoEstudado);
    } else {

```

```
this.criaArquivoDeCurso(arquivoDoCurso, {})
  .then(() => {
    this.adicionaTempoAoCurso(arquivoDoCurso, tempoEstudado);
  });
}
```

Podemos testar e ver que os dados estão sendo salvos perfeitamente! Como o JSON está sendo escrito em uma única linha em cada arquivo, podemos deixá-lo um pouco mais legível, formatando-o com espaços. Para isso, passamos mais um parâmetro para a função `writeFile`, o parâmetro `{spaces: 2}`:

```
// data.js

// restante do código comentado

adicionaTempoAoCurso(arquivoDoCurso, tempoEstudado) {
  let dados = {
    ultimoEstudo: new Date().toString(),
    tempo: tempoEstudado
  }
  jsonfile.writeFile(arquivoDoCurso, dados, {spaces: 2})
    .then(() => {
      console.log('Tempo salvo com sucesso');
    }).catch((err) => {
      console.log(err);
    })
}
```

Agora conseguimos criar um persistência dos cursos nos seus arquivos JSON, faltando apenas buscar esse tempo quando a aplicação for aberta. Veremos isso no próximo vídeo!