14

Async/await

Transcrição

Olá meus alunos! Este vídeo foi um pedido de vocês!

TypeScript se baseia no ES2015, todavia, na versão ES2017 foi introduzida a sintaxe `async/await`. Ela funciona da seguinte maneira. Dentro de uma uma função ou método `async`, isto é, uma função ou método declarado como `async NomeDoMetodoOuFuncao`, podemos tratar o retorno de promises de uma maneira muito especial.

Por padrão, capturamos o retorno de uma promise dentro da função `then`. Mas se dentro de uma função async, usamos a instrução `await` antes da chamada de um método **que retorne uma promise**, podemos capturar seu retorno sem a necessidade da chamada de `then`, como se ela fosse uma função síncrona tradicional.

Vejamos um exemplo:

// o método importDados é um método async!

```
@throttle()
async importaDados() {

    try {

       // usou await antes da chamada de this.service.obterNegociacoes()

        const negociacoesParaImportar = await this._service
            .obterNegociacoes(res => {

                if(res.ok) {
                    return res;
                } else {
                    throw new Error(res.statusText);
                }
            });

        const negociacoesJaImportadas = this._negociacoes.paraArray();

        negociacoesParaImportar
            .filter(negociacao =>
                !negociacoesJaImportadas.some(jaImportada =>
                    negociacao.ehIgual(jaImportada)))
            .forEach(negociacao =>
            this._negociacoes.adiciona(negociacao));

        this._negociacoesView.update(this._negociacoes);

    } catch(err) {
        this._mensagemView.update(err.message);
    }
}
```

Mas se não chamamos mais then, não chamaremos também `catch`, certo? Então, como conseguiremos tratar possíveis erros? Quando usamos async/wait, por mais que o código seja assíncrono, podemos usar `try` e `catch` para lidar com possíveis exceções em nosso código. Por mais que nosso código pareça um código síncrono, ele continua sendo um código assíncrono.

A boa notícia é que mesmo o TypeScript suportando apenas o ES2015 ele introduziu em sua sintaxe o async/await do ES2017 a partir da sua versão 2.3. Isso não quer dizer que somos obrigados a utilizá-la, mas seu uso melhor bastante a legibilidade do nosso código.