03

Definindo uma interface para a API

Transcrição

Como os dados recebidos são do tipo `any[]`, o TypeScript não consegue detectar erros como a escrita da propriedade errada. Podemos resolver isso criando uma `Interface`:

```
// app/ts/models/NegociacaoParcial.ts

export interface NegociacaoParcial {

    vezes: number,
    montante: number;
}
```

Veja que a interface define duas propriedades e seus tipos. Não é à toa que essas propriedades são as mesmas da nossa API externa. Agora, sem a necessidade de instanciarmos uma classe, até porque interfaces não podem ser instanciadas, podemos usar o tipo `NegociacaoParcial[]` em vez de `any[]` nos dados retornados pela nossa API. Isso será suficiente para que o TypeScript consiga analisar nosso código e checar por erros, inclusive fornecendo autocomplete:

```
    importarDados() {

        function isOK(res: Response) {

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

        fetch('http://localhost:8080/dados')
            .then(res => isOK(res))
            .then(res => res.json())
            .then((dados: NegociacaoParcial[]) => {
                dados
                    .map(dado => new Negociacao(new Date(), dado.vezes, dado.montante))
                    .forEach(negociacao => this._negociacoes.adiciona(negociacao));
                this._negociacoesView.update(this._negociacoes);
            })
            .catch(err => console.log(err.message));
    }
```

Essa solução não blinda da mudança inesperada da API, por exemplo, mudar `montante` para `valor`. Continuaríamos com um erro em runtime. Mas uma vez detectado o erro, podemos alterar nossa interface `NegociacaoParcial` mudando de `montante` para `valor` e isso resultará em um erro de compilação em todos os lugares que usam a interface. Perfeito, pois enquanto o desenvolvedor não ajustar para o formato da API que mudou, seu código não compilará. As chances de ele esquecer de mudar são nulas, pois seu código só compilará após a correção.