05

Ativando strictNullChecks

Transcrição

Muitas vezes, atribuímos `null` e `undefined` à variáveis para realizarmos alguma espécie de controle. Mas esses tipos podem causar problemas em runtime em nosso código se não tivermos cuidado com eles:

```
import { NegociacoesView, MensagemView } from '../views/index';
import { Negociacoes, Negociacao } from '../models/index';

export class NegociacaoController {

    private _inputData: JQuery;
    private _inputQuantidade: JQuery;
    private _inputValor: JQuery;
    private _negociacoes = new Negociacoes();
    private _negociacoesView = new NegociacoesView('#negociacoesView');
    private _mensagemView = new MensagemView('#mensagemView');

    constructor() {
        this._inputData = $('#data');
        this._inputQuantidade = $('#quantidade');
        this._inputValor = $('#valor');
        this._negociacoesView.update(this._negociacoes);

        let nome: string = '';
        nome = null; // algo permitido
    }
```

Contudo, o TypeScript possui o modo `strickNullChecks`. Neste modo, `null` e `undefined` não fazem parte do domínio dos tipos e só podem ser atribuídos a eles mesmos. Com a exceção de `undefined`, que pode ser atribuído a `void`. Isso pode ser interessante para evitarmos valores nulos e indefinidos em nosso projeto.

Vamos ativá-lo em `tsconfig.json`:

```
{
    "compilerOptions": {
        "target": "es6",
        "outDir": "app/js",
        "noEmitOnError": true,
        "noImplicitAny": true,
        "removeComments": true,
        "module": "system",
        "strictNullChecks": true
    },
    "include": [
        "app/ts/**/*"
    ]
}
```

Agora, nosso código deixará de compilar.

```
import { NegociacoesView, MensagemView } from '../views/index';
import { Negociacoes, Negociacao } from '../models/index';

export class NegociacaoController {

    private _inputData: JQuery;
    private _inputQuantidade: JQuery;
    private _inputValor: JQuery;
    private _negociacoes = new Negociacoes();
    private _negociacoesView = new NegociacoesView('#negociacoesView');
    private _mensagemView = new MensagemView('#mensagemView');

    constructor() {
        this._inputData = $('#data');
        this._inputQuantidade = $('#quantidade');
        this._inputValor = $('#valor');
        this._negociacoesView.update(this._negociacoes);

        let nome: string = '';
       // erro de compilação
        nome = null;
    }
   // código anterior omitido
```

```
Type 'null' is not assignable to type 'string'.
```

Vamos remover o teste que fizemos em nosso código e voltar nossa atenção para o arquivo `Views.ts` , que deixou de compilar após a ativação desse novo tipo de checagem:

```
export abstract class View<T> {

    protected _elemento: JQuery;
    private _escape: boolean;

    constructor(seletor: string, escapar?: boolean) {

        console.log(escape);
        this._elemento = $(seletor);
        // erro de compilação
        this._escapar = escapar;
    }
   // código posterior omitido
```

```
Type 'boolean | undefined' is not assignable to type 'boolean'.
  Type 'undefined' is not assignable to type 'boolean'.
```

```
export abstract class View<T> {

    protected _elemento: JQuery;
    private _escape: boolean;
```

```
    // tirou o tipo opcional que recebia undefined caso não fosse passado para um valor padrão, 
     constructor(seletor: string, escapar: boolean = false) {

         console.log(escape);
         this._elemento = $(seletor);
         this._escapar = escapar;
     }
```

Mas ainda há um problema com outra classe, a classe `Negociacoes`:

```
// app/ts/models/Negociacoes.ts
// código anterior omitido

    paraArray(): Negociacao[] {

       // erro de compilação
        return [].concat(this._negociacoes);
    }

// código posterior omitido
```

```
argument of type 'Negociacao[]' is not assignable to parameter of type 'never[]'.
  Type 'Negociacao' is not assignable to type 'never'.
```

Ele é resolvido indicando o tipo do `[]` com a sintaxe `as`:

```
    paraArray(): Negociacao[] {

        return ([] as Negociacao[]).concat(this._negociacoes);
    }
```