

Parâmetros opcionais

Transcrição

Vamos realizar uma pequena alteração na classe `View`. Hoje, interpretamos nosso template sem nos preocuparmos em escapá-lo. Nada impede de alguém colocar a tag `script` dentro do nosso template. Para evitar que isso aconteça, vamos implementar uma simples lógica de escape que remove a tag `script` e todo seu conteúdo dos templates das nossas views, mas deixaremos que o desenvolvedor escolha se deseja ou não realizar essa operação:

```
export abstract class View<T> {

    protected _elemento: JQuery;
    private _escapar: boolean;

    constructor(seletor: string, escapar: boolean) {

        this._elemento = $(seletor);
        this._escapar = escapar;
    }

    update(model: T) {

        let template = this.template(model);
        if(this._escapar)
            template = template.replace(/<script>[\s\S]*?</script>/, '');

        this._elemento.html(template);
    }

    abstract template(model: T): string;
}
```

No `constructor()` de `View` recebemos um novo parâmetro do tipo `boolean` que guardamos na propriedade da classe `escape`. Consultamos a propriedade no método `update`. Perfeito, mas isso fará com que `NegociacaoController` deixe de compilar, pois somos obrigados a passar o valor `true` ou `false`:

```
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();
    // erro de compilação
    private _negociacoesView = new NegociacoesView('#negociacoesView');

    // erro de compilação
    private _mensagemView = new MensagemView('#mensagemView');
```

Não queremos obrigar aos desenvolvedores terem que passar `false` quando não desejarem usar essa funcionalidade, queremos simplesmente que o parâmetro seja ignorado.

Podemos conseguir isso indicando que o parâmetro `escapar` é opcional, adicionando `?` ao final de sua declaração:

```
export abstract class View<T> {  
  
    protected _elemento: JQuery;  
    private _escape: boolean;  
  
    // tornando o parâmetro opcional!  
    constructor(seletor: string, escapar?: boolean) {  
  
        this._elemento = $(seletor);  
        this._escapar = escapar;  
    }  
    // código posterior omitido
```

Excelente! Nosso código continua compilando. Um ponto importante é que o valor de `escapar` quando não for passado, será `undefined`. Os tipos `null` e `undefined` podem ser atribuídos para outros tipos como `boolean`.