

TS TypeScript – Decorators em atributos

Na classe que **orquestra a inclusão de uma negociação** eu tenho **declarado como eu vou pegar o valor dos inputs** e **atribuir nos atributos da classe** para que eu **monte um objeto de negociação** e o adicione:

```
export class NegociacaoController {  
    private inputData: HTMLInputElement;  
    private inputQuantidade: HTMLInputElement;  
    private inputValor: HTMLInputElement;  
  
    private negociacoes = new Negociacoes();  
    private negociacoesView = new NegociacoesView('#negociacoesView', true);  
    private mensagemView = new MensagemView('#mensagemView');  
  
    constructor() {  
        this.inputData = <HTMLInputElement>document.querySelector('#data');  
        this.inputQuantidade = document.querySelector('#quantidade') as HTMLInputElement;  
        this.inputValor = document.querySelector('#valor') as HTMLInputElement;  
        this.negociacoesView.update(this.negociacoes);  
    }  
}
```

Negociações

Data
dd/mm/aaaa
Quantidade
1
Valor
0,0

Incluir

DATA	QUANTIDADE
------	------------

Mas olha como esse código está repetitivo, eu estou **usando querySelector para pegar elementos por um ID**, e eu tenho que repetir ele 3 vezes no meu construtor, e se eu tivesse mais campos ainda de formulário para receber? e se eu quiser usar em outro lugar?

Decorators em atributos

Eu posso resolver isso usando um **Decorator** de atributo que vai conter essa lógica de pegar um elemento pelo seletor. O esqueleto dele **é bem parecido com um decorator comum**, eu vou **ter meu target**, minha propertyKey, mas tem um problema agora.

O **target** apenas tem o valor do **“prototype”** caso o método não seja estático, ou o método construtor caso o **“local decorado”** seja estático. Isso significa que eu **NÃO TENHO UMA INSTÂNCIA** da classe que eu quero **injetar o elemento**, se eu não tenho essa **instância** como eu vou atribuir esse **valor do elemento selecionado no atributo da classe**?

```
export function domInjector(seletor : string) {  
    return function(  
        target: any,  
        propertyKey : string,  
    ) {}  
}
```

```
export class NegociacaoController {  
    private inputData: HTMLInputElement;  
    private inputQuantidade: HTMLInputElement;  
    private inputValor: HTMLInputElement;  
  
    private negociacoes = new Negociacoes();  
    private negociacoesView = new NegociacoesView('#negociacoesView', true);  
    private mensagemView = new MensagemView('#mensagemView');  
  
    constructor() {  
        this.inputData = <HTMLInputElement>document.querySelector('#data');  
        this.inputQuantidade = document.querySelector('#quantidade') as HTMLInputElement;  
        this.inputValor = document.querySelector('#valor') as HTMLInputElement;  
        this.negociacoesView.update(this.negociacoes);  
    }  
}
```

Uma solução possível é **modificar o prototype** desse **“target”**, já que a gente **o possui**, podemos **definir as propriedades** dele com o **Object de TypeScript**. Basicamente **estamos pegando o elemento através do seletor passado** e **definindo ele como um getter** da **classe** quando ela for chamada em tempo de execução:

