

≡ 18

## toHaveBeenCalled

Vejamos a chamada de `toHaveBeenCalled` :

```
it(`#${LikeWidgetComponent.prototype.like.name}  
  should trigger (@Output liked) when called`, () => {  
  spyOn(component.liked, 'emit');  
  fixture.detectChanges();  
  component.like();  
  expect(component.liked.emit).toHaveBeenCalled();  
}  
);
```

**COPIAR CÓDIGO**

Marque a alternativa verdadeira sobre o código anterior.

*Seleciona uma alternativa*

**A**

O teste falhará, pois a chamada de `toHaveBeenCalled` não recebeu um spy como parâmetro.

**B**

O teste com `toHaveBeenCalled` funcionará, pois a função `spyOn` modificou o método `component.liked` transformando-o em um spy.

**C**

O teste falhará porque o método `emit` não faz parte da API de uma output property.

