

Teste de Software

**IAN DA SILVA SANTOS
CONCEIÇÃO**

Parte 1)

Iniciando abrindo o site do StackOverflow

The screenshot shows the Stack Overflow homepage. At the top, there's a navigation bar with the Stack Overflow logo, links for 'About', 'Products', and 'OverflowAI', a search bar, and 'Log in' and 'Sign up' buttons. Below the navigation bar, a blue banner announces 'The 2024 Developer Survey results are live!' with a 'See the results' button. The main content area features a large illustration of a developer at a desk with the text 'Every developer has a tab open to Stack Overflow.' and a 'Log in' button. Below this, there are sections for 'COLLECTIVES' and 'TEAMS'. The 'TEAMS' section highlights 'Stack Overflow FOR TEAMS' and 'overflow AI'.

vamos na barra de pesquisa e adicionamos nosso filtro para obter mais precisão.

O filtro utilizado foi: "[unit-testing] or [junit] or [pytest]"

Adicionando a String, vamos para essa tela:

The screenshot shows the Stack Overflow 'All Questions' page. The header includes the Stack Overflow logo, navigation links (About, Products, OverflowAI), a search bar with the query '[unit-testing] or [junit] or [pytest]', and 'Log in' and 'Sign up' buttons. A blue banner at the top says 'The 2024 Developer Survey results are live! See the results'. The left sidebar contains navigation links: Home, Questions, Tags, Users, Companies, LABS, Jobs, Discussions, COLLECTIVES, and TEAMS. The main content area is titled 'All Questions' and shows a list of questions tagged with 'unit-testing', 'junit', or 'pytest'. The first question is 'How do I test a class that has private methods, fields or inner classes?' with 3230 votes and 59 answers. The second question is 'How do you assert that a certain exception is thrown in JUnit tests?' with 2311 votes and 35 answers. The third question is 'How do you test that a Python function throws an exception?' with 1243 votes and 19 answers. The fourth question is 'How to mock void methods with Mockito' with 1213 votes and 11 answers. The fifth question is 'How do I generate a stream from a string?' with 1031 votes and 14 answers. The right sidebar has a 'Featured on Meta' section and a 'Related Tags' section.

A seguir, clicamos no filtro lateral e acrescentamos o filtro de ordem decrescente de pontuação: "Highest score"

The screenshot shows the Stack Overflow 'All Questions' page with the filter sidebar open. The sidebar has three sections: 'Filter by' with options 'No answers', 'No accepted answer', and 'Has bounty'; 'Sorted by' with options 'Newest', 'Recent activity', 'Highest score' (selected), 'Most frequent', and 'Bounty ending soon'; and 'Tagged with' with 'My watched tags' and 'The following tags:' (selected). The 'The following tags:' section shows 'unit-testing', 'junit', and 'pytest' selected. The 'Apply filter' button is at the bottom left, and the 'Cancel' button is at the bottom right.

A seguir, selecionamos uma pergunta cadastrada no StackOverflow que tenha ao menos uma resposta aceita e no mínimo 400 votos.

Dentro todas as disponíveis, selecionamos uma:

3230 votes	How do I test a class that has private methods, fields or inner classes?	How do I use JUnit to test a class that has internal private methods, fields or nested classes? It seems bad to change the access modifier for a method just to be able to run a test.	java unit-testing junit tdd	Community wiki 20 revs, 11 users 48% Raedwald
✓ 59 answers				
1.3m views				
2311 votes	How do you assert that a certain exception is thrown in JUnit tests?	How can I use JUnit idiomatically to test that some code throws an exception? While I can certainly do something like this: <code>@Test public void testFooThrowsIndexOutOfBoundsException() { boolean ...</code>	java exception junit junit4 assert	SCdF 58.7k asked Oct 1, 2008 at 6:56
✓ 35 answers				
2.0m views				
1243 votes	How do you test that a Python function throws an exception?	How does one write a unit test that fails only if a function doesn't throw an expected exception?	python unit-testing exception	Daryl Spitzer 148k asked Sep 24, 2008 at 20:00
✓ 19 answers				
958k views				
1213 votes	How to mock void methods with Mockito	How to mock methods with void return type? I implemented an observer pattern but I can't mock it with Mockito because I don't know how. And I tried to find an example on the Internet but didn't ...	java unit-testing mocking mockito void	ibrahimylimaz 18.8k asked Feb 16, 2010 at 20:54
✓ 11 answers				
1.3m views				
1031 votes	How do I generate a stream from a string?	I need to write a unit test for a method that takes a stream which comes from a text file. I would like to do something like this: <code>Stream s = GenerateStreamFromString("a,b\n c,d");</code>	c# unit-testing string stream	Omu 70.9k asked Dec 10, 2009 at 8:14
✓ 14 answers				
854k views				
1015 votes	Running unittest with typical test directory structure	The very common directory structure for even a simple Python module seems to be to separate the unit tests into their own test directory: <code>new_project/ antigravity/ antigravity.py test/ ...</code>	python unit-testing	Major Major 10.7k asked Dec 13, 2009 at 16:10
✓ 27 answers				
440k views				
994 votes	What's the difference between faking, mocking, and stubbing?	I know how I use these terms, but I'm wondering if there are accepted definitions for faking, mocking, and stubbing for unit tests? How do you define these for your tests? Describe situations where ...	unit-testing mocking language-agnostic terminology definition	tvanfosson 530k asked Dec 6, 2008 at 15:17
✓ 17 answers				
318k views				

A selecionada será “How to test the type of a thrown exception in jest”

How to test the type of a thrown exception in Jest

Asked 6 years, 11 months ago Modified 10 months ago Viewed 750k times



I'm working with some code where I need to test the type of an exception thrown by a function (is it `TypeError`, `ReferenceError`, etc.?).

571



My current testing framework is AVA and I can test it as a second argument `t.throws` method, like here:



```
it('should throw Error with message \'UNKNOWN ERROR\' when no params were passed', (t) => {
  const error = t.throws(() => {
    throwError();
  }, TypeError);

  t.is(error.message, 'UNKNOWN ERROR');
});
```

I started rewriting my tests in Jest and couldn't find how to easily do that. Is it even possible?

javascript unit-testing jestjs

Share Follow

edited Sep 24, 2020 at 18:05



Peter Mortensen

31.4k ● 22 ● 109 ● 132

asked Sep 4, 2017 at 18:43



bartsmykla

5,881 ● 2 ● 11 ● 6

Esse foi o Questionamento colocado no StackOverflow por um usuário.

E essa foi a resposta mais avaliada e aceita pelo Site:



815



In Jest you have to pass a function into `expect(function).toThrow(<blank or type of error>)`.

Example:

```
test("Test description", () => {
  const t = () => {
    throw new TypeError();
  };
  expect(t).toThrow(TypeError);
});
```

Or if you also want to check for error message:

```
test("Test description", () => {
  const t = () => {
    throw new TypeError("UNKNOWN ERROR");
  };
  expect(t).toThrow(TypeError);
  expect(t).toThrow("UNKNOWN ERROR");
});
```

If you need to test an existing function whether it throws with a set of arguments, you have to wrap it inside an anonymous function in `expect()`.

Example:

```
test("Test description", () => {
  expect(() => {http.get(yourUrl, yourCallbackFn)}).toThrow(TypeError);
});
```

Share Follow

edited Feb 9, 2021 at 13:28

answered Sep 11, 2017 at 12:06



PeterDanis

9,086 ● 2 ● 14 ● 24

Parte 2)

Com base nesse questionamento feito por um usuário membro do Stack Overflow, pegamos e criamos um código para que possamos mostrar como solucionar o problema proposto por ele utilizando a resposta mais bem avaliada para aquele questionamento no Stack Overflow

Nosso código foi esse:

```
1 function throwError() {
2   throw new TypeError('This is a tpe error');
3 }
4
5 module.exports = throwError;
```

E a solução para ela utilizando a resposta do usuário do Stack Overflow foi essa:

```
exemplo.test.js > ...
1  const throwError = require('./exemplo');
2
3  test("should throw TypeError", () => {
4    const t = () => {
5      throwError();
6    };
7    expect(t).toThrow(TypeError);
8    //expect(t).toThrow('This is a type error');
9  });
```

Trazendo o resultado esperado no Jest do terminal da IDE:

```
PS C:\Users\Ian\Documents\TesteJest> npm test

> testejest@1.0.0 test
> jest

PASS ./sum.test.js
PASS ./exemplo.test.js
PASS ./geradorErro.test.js

Test Suites: 3 passed, 3 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        2.984 s
Ran all test suites.
```

Caso o tipo da exceção lançada não fosse a mesma, daria erro, retornando no comando "npm test" que havia um erro e mostrando onde o erro se encontrava no código para que o usuário possa fazer a devida correção;

Para conseguirmos fazer o teste com o Jest, precisamos inicialmente instalar o Jest em nossa máquina através do terminal.

Abaixo fica todo o processo que precisamos fazer para instalar o Jest:

Iniciando

Instale Jest usando seu gerenciador de pacotes favorito:

npm Yarn pnpm

```
npm install --save-dev jest
```

Vamos começar por escrever um teste para uma função hipotética que soma dois números. Primeiro, crie um arquivo `sum.js`:

```
function sum(a, b) {  
  return a + b;  
}  
module.exports = sum;
```

Em seguida, crie um arquivo chamado `sum.test.js`. Este irá conter o nosso teste real:

```
const sum = require('./sum');  
  
test('adds 1 + 2 to equal 3', () => {  
  expect(sum(1, 2)).toBe(3);  
});
```

Adicione a seguinte seção ao seu `package.json`:

```
{  
  "scripts": {  
    "test": "jest"  
  }  
}
```

Por fim, execute `yarn test` ou `npm run test` e o Jest irá imprimir a seguinte mensagem:

```
PASS   ./sum.test.js  
✓ adds 1 + 2 to equal 3 (5ms)
```


Para finalizar, criamos também outro código, referenciando o mesmo questionamento colocado no Stack Overflow sobre “**Como testar o tipo de uma exceção lançada por uma função?**”

O código criado foi o seguinte:

```
geradorErro.js > ...
1  // errorGenerator.js
2  function errorGenerator(type) {
3      if (type === 'type') {
4          throw new TypeError('This is a type error');
5      } else if (type === 'reference') {
6          throw new ReferenceError('This is a reference error');
7      } else {
8          throw new Error('This is a generic error');
9      }
10 }
11
12 module.exports = errorGenerator;
```

E o código de teste em Jest para esse código foi esse:

```

geradorErro.test.js > ...
1  // errorGenerator.test.js
2  const errorGenerator = require('./geradorErro');
3
4  test('throws TypeError when type is "type"', () => {
5    const t = () => {
6      errorGenerator('type');
7    };
8    expect(t).toThrow(TypeError);
9    expect(t).toThrow('This is a type error');
10  });
11
12  test('throws ReferenceError when type is "reference"', () => {
13    const t = () => {
14      errorGenerator('reference');
15    };
16    expect(t).toThrow(ReferenceError);
17    expect(t).toThrow('This is a reference error');
18  });
19
20  test('throws generic Error when type is not specified', () => {
21    const t = () => {
22      errorGenerator('other');
23    };
24    expect(t).toThrow(Error);
25    expect(t).toThrow('This is a generic error');
26  });

```

Esses testes fazem as seguintes coisas:

Primeiro teste (**throws TypeError when type is "type"**):

- Verifica se a função **geradorErro** lança um `TypeError` quando chamada com o argumento 'type'.
- Verifica também a mensagem do erro para garantir que ela é a esperada.

Segundo teste (**throws ReferenceError when type is "reference"**):

- Verifica se a função **geradorErro** lança um `ReferenceError` quando chamada com o argumento 'reference'.
- Verifica também a mensagem do erro para garantir que ela é a esperada.

Terceiro teste (**throws generic Error when type is not specified**):

- Verifica se a função **geradorErro** lança um erro genérico (Error) quando chamada com qualquer outro argumento.
- Verifica também a mensagem do erro para garantir que ela é a esperada.

Dessa forma, também respondemos o questionamento proposto no Stack OverFlow.

O Repositório do GitHub é o seguinte:

https://github.com/Ian-Silva/Teste_Software_2024_Conceicao_Ian/tree/main