

Aula 20 - Testes Automatizados com JUnit

Docupedia Export

Author: Sílio Leonardo (SO/OPM-TS21-BR)

Date: 18-Oct-2024 15:57

Table of Contents

1 Instalando JUnit com Maven

4

- [Instalando JUnit com Maven](#)

1 Instalando JUnit com Maven

Para instalar a biblioteca de testes automatizados chamada JUnit precisamos apenas modificar o pom.xml:

pom.xml

```
1  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3      <modelVersion>4.0.0</modelVersion>
4      <groupId>com.junitexample</groupId>
5      <artifactId>my-junit-app</artifactId>
6      <version>0.1</version>
7
8      <dependencyManagement>
9
10         <dependencies>
11             <dependency>
12                 <groupId>org.junit</groupId>
13                 <artifactId>junit-bom</artifactId>
14                 <version>5.11.2</version>
15                 <type>pom</type>
16                 <scope>import</scope>
17             </dependency>
18         </dependencies>
19     </dependencyManagement>
20
21     <dependencies>
22
23         <dependency>
24             <groupId>org.junit.jupiter</groupId>
25             <artifactId>junit-jupiter</artifactId>
26             <scope>test</scope>
27         </dependency>
28
29     </dependencies>
30
31     <build>
```

```
32     <plugins>
33
34         <plugin>
35             <artifactId>maven-surefire-plugin</artifactId>
36             <version>3.5.0</version>
37         </plugin>
38
39     </plugins>
40 </build>
41 </project>
```

Vamos implementar um teste para validação de datas:

src/main/java/com/junitexample/DateValidator.java

```
1  package com.junitexample;
2
3  public class DateValidator {
4      public Boolean validate(int day, int month, int year) {
5          if (day < 1 || day > 31)
6              return false;
7
8          if (month < 1 || month > 12)
9              return false;
10
11         if (day < 29)
12             return true;
13
14         // for days 31, 30 or 29
15         if (day == 31)
16             return month == 1 ||
17                    month == 3 || month == 5 ||
18                    month == 7 || month == 8 ||
19                    month == 10 || month == 12;
20
21         if (day == 30)
22             return month != 2;
23
24         // for day 29
```

```
25         if (month != 2)
26             return true;
27
28         return
29             year % 4 == 0 && (
30                 year % 100 != 0 ||
31                 year % 400 == 0);
32     }
33 }
```

Essa seria uma implementação útil a ser usada em um grande software de cadastro. Mas, mudanças no código e requisitos podem criar Bugs, nesta, e em outras funcionalidades. Por isso, é importante testá-las sempre e garantir que o projeto está funcionando. Vamos mostrar como escrever um teste no projeto.

src/test/java/com/junitexample/DateValidator.java

```
1  package com.junitexample;
2
3  import static org.junit.jupiter.api.Assertions.assertEquals;
4  import org.junit.jupiter.api.Test;
5
6  import com.junitexample.DateValidator;
7
8  // A classe deve terminar com 'Test' no nome.
9  class MyTest {
10     private final DateValidator validator = new DateValidator();
11
12     @Test
13     void validate() {
14         assertEquals(
15             validator.validate(29, 2, 2024),
16             true
17         );
18
19         assertEquals(
20             validator.validate(29, 2, 2025),
21             false
22         );
23
24         assertEquals(
```

```
25         validator.validate(30, 2, 2024),
26         false
27     );
28
29     assertEquals(
30         validator.validate(30, 12, 1800),
31         true
32     );
33
34     assertEquals(
35         validator.validate(29, 2, 1800),
36         false
37     );
38
39     assertEquals(
40         validator.validate(29, 2, 1600),
41         true
42     );
43
44     assertEquals(
45         validator.validate(4, 1, 2023),
46         true
47     );
48
49     assertEquals(
50         validator.validate(31, 4, 2000),
51         false
52     );
53
54     assertEquals(
55         validator.validate(29, 4, 2001),
56         true
57     );
58 }
59 }
```

Agora basta rodar:

1

```
..\apache-maven-3.9.6\bin\mvn.cmd clean test
```

```
[INFO] -----  
[INFO] T E S T S  
[INFO] -----  
[INFO] Running com.junitexample.MyTest  
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.019 s - in com.junitexample.MyTest  
[INFO]  
[INFO] Results:  
[INFO]  
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0  
[INFO]  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 3.287 s  
[INFO] Finished at: 2024-05-15T14:04:36-03:00  
[INFO] -----
```

Ou use a extensão Java Test Runner (que vem junto com o Java Extension Pack) e rode os testes clicando no executar ao lado da classe desejada.