

Junit5中实现参数化测试



城下秋草

关注

 0.192 2018.09.05 22:13:19 字数 663 阅读 4,709

从Junit5开始，对参数化测试支持进行了大幅度的改进和提升。下面我们就一起来详细看看Junit5参数化测试的方法。

部署和依赖

和Junit4相比，Junit5框架更多在向测试平台演进。其核心组成也从以前的一个Junit的jar包更换成由多个模块组成。本文所需要依赖模块如下：

- junit-jupiter-engine: Junit的核心测试引擎
- junit-jupiter-params: 编写参数化测试所需要的依赖包
- junit-platform-launcher: 从IDE(Intelij/Eclipses)等运行时所需要的启动器

另外，为了从Maven命令行工具中运行Juint,还需要junit-platform-surefire-provider包的依赖。

maven的pom.xml文件中添加如下来进行安装

```
1  <dependencies>
2      <dependency>
3          <groupId>org.junit.jupiter</groupId>
4          <artifactId>junit-jupiter-engine</artifactId>
5          <version>5.2.0</version>
6          <scope>test</scope>
7      </dependency>
8      <dependency>
9          <groupId>org.junit.jupiter</groupId>
10         <artifactId>junit-jupiter-params</artifactId>
11         <version>5.2.0</version>
12         <scope>test</scope>
13     </dependency>
14     <dependency>
15         <groupId>org.junit.platform</groupId>
16         <artifactId>junit-platform-launcher</artifactId>
17         <version>1.2.0</version>
18     </dependency>
19 </dependencies>
20
21 <build>
22     <plugins>
23         <plugin>
24             <groupId>org.apache.maven.plugins</groupId>
25             <artifactId>maven-surefire-plugin</artifactId>
26             <version>2.22</version>
27         </plugin>
28     </plugins>
29 </build>
```

Junit5 参数源详解

value source

value source是最简单的参数源，通过注解可以直接指定携带的运行参数。

- String values: @ValueSource(strings = {"foo", "bar", "baz"})
- Double values: @ValueSource(doubles = {1.5D, 2.2D, 3.0D})

推荐阅读

- idea 开发必备插件
阅读 1,856
- springboot项目整合阿里开源框架
EasyExcel，附带整理好的Excel操作...
阅读 1,725
- java中使用jwt
阅读 104
- springboot + jta + mysql + mongo
分布式（多种数据源）事务
阅读 2,643
- HashMap原理解析
阅读 36
- PGK JV

示例代码如下：

```
1 import static org.junit.jupiter.api.Assertions.assertEquals;
2
3 import org.junit.jupiter.params.ParameterizedTest;
4 import org.junit.jupiter.params.provider.ValueSource;
5
6 public class ValueSourcesExampleTest {
7
8     @ParameterizedTest
9     @ValueSource(ints = {2, 4, 8})
10    void testNumberShouldBeEven(int num) {
11        assertEquals(0, num % 2);
12    }
13
14    @ParameterizedTest
15    @ValueSource(strings = {"Radar", "Rotor", "Tenet", "Madam", "Racecar"})
16    void testStringShouldBePalindrome(String word) {
17        assertEquals(isPalindrome(word), true);
18    }
19
20    @ParameterizedTest
21    @ValueSource(doubles = {2.D, 4.D, 8.D})
22    void testDoubleNumberBeEven(double num) {
23        assertEquals(0, num % 2);
24    }
25
26    boolean isPalindrome(String word) {
27        return word.toLowerCase().equals(new StringBuffer(word.toLowerCase()).reverse().toString());
28    }
29 }
```



推荐阅读

- idea 开发必备插件
阅读 1,856
- springboot项目整合阿里开源框架
EasyExcel， 附带整理好的Excel操作...
阅读 1,725
- java中使用jwt
阅读 104
- springboot + jta + mysql + mongo
分布式（多种数据源）事务
阅读 2,643
- HashMap原理解析
阅读 36

输出

```
1 [INFO] -----
2 [INFO] T E S T S
3 [INFO] -----
4 [INFO] Running qiucuo.learning.ParaTest
5 [INFO] Tests run: 11, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.155 s - in qiucuo.learn
6 [INFO]
7 [INFO] Results:
8 [INFO]
9 [INFO] Tests run: 11, Failures: 0, Errors: 0, Skipped: 0
```

Enum Source

枚举参数源，允许我们将参数值由给定Enum枚举类型传入。并可以通过制定约束条件或正则匹配来筛选传入参数

```
1 import static org.junit.jupiter.api.Assertions.assertFalse;
2 import static org.junit.jupiter.api.Assertions.assertTrue;
3
4 import java.util.EnumSet;
5 import java.util.concurrent.TimeUnit;
6 import org.junit.jupiter.params.ParameterizedTest;
7 import org.junit.jupiter.params.provider.EnumSource;
8 import org.junit.jupiter.params.provider.EnumSource.Mode;
9
10 public class EnumSourcesExampleTest {
11
12     @ParameterizedTest(name = "[{index}] TimeUnit: {arguments}")
13     @EnumSource(TimeUnit.class)
14     void testTimeUnitMinimumNanos(TimeUnit unit) {
15         assertTrue(unit.toMillis(2000000L) > 1);
16     }
17
18     @ParameterizedTest
19     @EnumSource(value = TimeUnit.class, names = {"SECONDS", "MINUTES"})
20     void testTimeUnitJustSecondsAndMinutes(TimeUnit unit) {
```



```
26
27 @ParameterizedTest
28 @EnumSource(value = TimeUnit.class, mode = Mode.EXCLUDE, names = {"SECONDS", "MINUTES"})
29 void testTimeUnitExcludingSecondsAndMinutes(TimeUnit unit) {
30     assertFalse(EnumSet.of(TimeUnit.SECONDS, TimeUnit.MINUTES).contains(unit));
31     assertTrue(EnumSet
32         .of(TimeUnit.DAYS, TimeUnit.HOURS, TimeUnit.MILLISECONDS, TimeUnit.NANOSECONDS,
33             TimeUnit.MICROSECONDS).contains(unit));
34 }
35
36 @ParameterizedTest
37 @EnumSource(value = TimeUnit.class, mode = Mode.MATCH_ALL, names = ".*SECONDS")
38 void testTimeUnitIncludingAllTypesOfSecond(TimeUnit unit) {
39     assertFalse(EnumSet.of(TimeUnit.DAYS, TimeUnit.HOURS, TimeUnit.MINUTES).contains(unit));
40     assertTrue(EnumSet
41         .of(TimeUnit.SECONDS, TimeUnit.MILLISECONDS, TimeUnit.NANOSECONDS,
42             TimeUnit.MICROSECONDS).contains(unit));
43 }
44
45 }
```

输出:

```
1 [INFO] -----
2 [INFO] T E S T S
3 [INFO] -----
4 [INFO] Running qiucao.learning.ParaTest
5 [INFO] Tests run: 18, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.206 s - in qiucao.learr
6 [INFO]
7 [INFO] Results:
8 [INFO]
9 [INFO] Tests run: 18, Failures: 0, Errors: 0, Skipped: 0
10
```

Method Source

通过其他的Java方法函数来作为参数源。引用的方法返回值必须是Stream, Iterator 或者Iterable.

```
1 import static org.junit.jupiter.api.Assertions.assertAll;
2 import static org.junit.jupiter.api.Assertions.assertNotNull;
3 import static org.junit.jupiter.api.Assertions.assertTrue;
4
5 import java.util.stream.IntStream;
6 import java.util.stream.Stream;
7 import org.junit.jupiter.params.ParameterizedTest;
8 import org.junit.jupiter.params.provider.Arguments;
9 import org.junit.jupiter.params.provider.MethodSource;
10
11 public class MethodSourceExampleTest {
12     @ParameterizedTest
13     @MethodSource("stringGenerator")
14     void shouldNotBeNullString(String arg){
15         assertNotNull(arg);
16     }
17
18     @ParameterizedTest
19     @MethodSource("intGenerator")
20     void shouldBeNumberWithinRange(int arg){
21         assertAll(
22             () -> assertTrue(arg > 0),
23             () -> assertTrue(arg <= 10)
24         );
25     }
26
27     @ParameterizedTest(name = "[{index}] user with id: {0} and name: {1}")
28     @MethodSource("userGenerator")
29     void shouldUserWithIdAndName(long id, String name){
30         assertNotNull(id);
31         assertNotNull(name);
32     }
33 }
```

推荐阅读

- idea 开发必备插件
阅读 1,856
- springboot项目整合阿里开源框架
EasyExcel, 附带整理好的Excel操作...
阅读 1,725
- java中使用jwt
阅读 104
- springboot + jta + mysql + mongo
分布式（多种数据源）事务
阅读 2,643
- HashMap原理解析
阅读 36

写下你的评论...

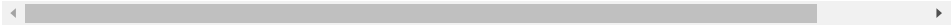
 评论0  赞4 ...

```
39     return IntStream.range(1,10);
40 }
41
42 static Stream<Arguments> userGenerator(){
43     return Stream.of(Arguments.of(1L, "Sally"), Arguments.of(2L, "Terry"), Arguments.of(3L, "Fred"));
44 }
45 }
```



输出:

```
1 [INFO] -----
2 [INFO] T E S T S
3 [INFO] -----
4 [INFO] Running qiucao.learning.ParaTest
5 [INFO] Tests run: 16, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.191 s - in qiucao.learning.ParaTest
6 [INFO]
7 [INFO] Results:
8 [INFO]
9 [INFO] Tests run: 16, Failures: 0, Errors: 0, Skipped: 0
```



Argument Source

通过参数类来作为参数源。这里引用的类必须实现ArgumentsProvider接口。示例如下:

```
1 import static org.junit.jupiter.api.Assertions.assertFalse;
2 import static org.junit.jupiter.api.Assertions.assertTrue;
3
4 import java.util.stream.Stream;
5 import org.junit.jupiter.api.extension.ExtensionContext;
6 import org.junit.jupiter.params.ParameterizedTest;
7 import org.junit.jupiter.params.provider.Arguments;
8 import org.junit.jupiter.params.provider.ArgumentsProvider;
9 import org.junit.jupiter.params.provider.ArgumentsSource;
10
11 public class ArgumentsSourceExampleTest {
12
13     @ParameterizedTest
14     @ArgumentsSource(CustomArgumentsGenerator.class)
15     void testGeneratedArguments(double number) throws Exception {
16         assertFalse(number == 0.0);
17         assertTrue(number > 0);
18         assertTrue(number < 1);
19     }
20
21     static class CustomArgumentsGenerator implements ArgumentsProvider {
22
23         @Override
24         public Stream<? extends Arguments> provideArguments(ExtensionContext context) {
25             return Stream.of(Math.random(), Math.random(), Math.random(), Math.random(), Math.random())
26                 .map(Arguments::of);
27         }
28     }
29 }
```



CSV Source

通过指定csv格式(comma-separated-values)的注解作为参数源

```
1 import static org.junit.jupiter.api.Assertions.assertTrue;
2
3 import java.util.HashMap;
4 import java.util.Map;
5 import org.junit.jupiter.params.ParameterizedTest;
6 import org.junit.jupiter.params.provider.CsvSource;
7
8 public class CsvSourceExampleTest {
```



推荐阅读

- idea 开发必备插件
阅读 1,856
- springboot项目整合阿里开源框架
EasyExcel, 附带整理好的Excel操作...
阅读 1,725
- java中使用jwt
阅读 104
- springboot + jta + mysql + mongo
分布式 (多种数据源) 事务
阅读 2,643
- HashMap原理解析
阅读 36

```
15     idToUsername.put(3L, "Tim");
16 }
17
18 @ParameterizedTest
19 @CsvSource({"1,Selma", "2,Lisa", "3,Tim"})
20 void testUsersFromCsv(long id, String name) {
21     assertTrue(idToUsername.containsKey(id));
22     assertTrue(idToUsername.get(id).equals(name));
23 }
24 }
```

输出:

```
1 [INFO] -----
2 [INFO]  T E S T S
3 [INFO] -----
4 [INFO] Running qiucao.learning.ParaTest
5 [INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.164 s - in qiucao.learn
6 [INFO]
7 [INFO] Results:
8 [INFO]
9 [INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0
```

CSV File Source

除了使用csv参数源，这里也支持使用csv文件作为参数源

假设users.csv 文件包含如下csv格式的数据

```
1,Selma
2,Lisa
3,Tim
```

代码示例如下

```
1 import static org.junit.jupiter.api.Assertions.assertTrue;
2
3 import java.util.HashMap;
4 import java.util.Map;
5 import org.junit.jupiter.params.ParameterizedTest;
6 import org.junit.jupiter.params.provider.CsvFileSource;
7 import org.junit.jupiter.params.provider.CsvSource;
8
9 public class CsvFileSourceExampleTest {
10
11     Map<Long, String> idToUsername = new HashMap<>();
12
13     {
14         idToUsername.put(1L, "Selma");
15         idToUsername.put(2L, "Lisa");
16         idToUsername.put(3L, "Tim");
17     }
18
19     @ParameterizedTest
20     @CsvFileSource(resources = "/users.csv")
21     void testUsersFromCsv(long id, String name) {
22         assertTrue(idToUsername.containsKey(id));
23         assertTrue(idToUsername.get(id).equals(name));
24     }
25 }
```

输出:

写下你的评论...

 评论0

 赞4



推荐阅读

- idea 开发必备插件
阅读 1,856
- springboot项目整合阿里开源框架
EasyExcel， 附带整理好的Excel操作...
阅读 1,725
- java中使用jwt
阅读 104
- springboot + jta + mysql + mongo
分布式（多种数据源）事务
阅读 2,643
- HashMap原理解析
阅读 36

```
3 [INFO] -----
4 [INFO] Running qiucuo.learning.ParaTest
5 [INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.199 s - in qiucuo.learnj
6 [INFO]
7 [INFO] Results:
8 [INFO]
9 [INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0
```

参数转换

JUnit allows us to convert arguments to the target format we need in our tests.

There are two possible conversion types:

隐式转换

JUnit提供了很对内建的格式转化支持，特别是string和常用的数据类型

以下是支持和string型进行转换的类型

- Boolean
- Byte
- Character
- Short
- Integer
- Long
- Float
- Double
- Enum subclass
- Instant
- LocalDate
- LocalDateTime
- LocalTime
- OffsetTime
- OffsetDateTime
- Year
- YearMonth
- ZonedDateTime

显式转换

Junit5中可以使用@ConvertWith(MyConverter.class) 注解来实现 SimpleArgumentConverter.

代码示例：

```
1
2
3 import static org.junit.jupiter.api.Assertions.assertEquals;
4 import static org.junit.jupiter.api.Assertions.assertNotNull;
5 import static org.junit.jupiter.api.Assertions.assertTrue;
6
7 import java.time.LocalDate;
8 import java.time.Month;
9 import java.util.UUID;
```

推荐阅读

- idea 开发必备插件
阅读 1,856
- springboot项目整合阿里开源框架
EasyExcel， 附带整理好的Excel操作...
阅读 1,725
- java中使用jwt
阅读 104
- springboot + jta + mysql + mongo
分布式（多种数据源）事务
阅读 2,643
- HashMap原理解析
阅读 36

```
16
17 @ParameterizedTest
18 @ValueSource(strings = "2017-07-11")
19 void testImplicitArgumentConversion(LocalDate date) throws Exception {
20     assertTrue(date.getYear() == 2017);
21     assertTrue(date.getMonth().equals(Month.JULY));
22     assertTrue(date.getDayOfMonth() == 11);
23 }
24
25 @ParameterizedTest
26 @ValueSource(strings = "B4627B3B-ACC4-44F6-A2EB-FCC94DAB79A5")
27 void testImplicitArgumentConversion(@ConvertWith(ToUUIDArgumentConverter.class) UUID uuid)
28     throws Exception {
29     assertNotNull(uuid);
30     assertTrue(uuid.getLeastSignificantBits() == -6706989278516512347L);
31 }
32
33 static class ToUUIDArgumentConverter extends SimpleArgumentConverter {
34
35     @Override
36     protected Object convert(Object source, Class<?> targetType) {
37         assertEquals(UUID.class, targetType, "may only convert to UUID");
38         return UUID.fromString(String.valueOf(source));
39     }
40 }
41
42 }
```

输出:

```
1 [INFO] -----
2 [INFO] T E S T S
3 [INFO] -----
4 [INFO] Running qiucao.learning.ParaTest
5 [INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.487 s - in qiucao.learn
6 [INFO]
7 [INFO] Results:
8 [INFO]
9 [INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
10
```

补充对照: JUnit 4中参数化测试方法

代码如下:

```
1 @RunWith(Parameterized.class)
2 public class ParameterizedTest {
3     @Parameters(name = "Run #{index}: {0}^2={1}")
4     public static Iterable<Object[]> data() {
5         return Arrays.asList(new Object[][] { { 1, 1 }, { 2, 4 }, { 3, 9 },
6             { 4, 16 }, { 5, 25 } });
7     }
8
9     private final int input;
10    private final int resultExpected;
11
12    public ParameterizedTest(final int input, final int result) {
13        this.input = input;
14        this.resultExpected = result;
15    }
16
17    @Test
18    public void testUserMapping() {
19        Calculator calc = new Calculator();
20        assertEquals(resultExpected, calc.square(input));
21    }
22 }
```

推荐阅读

- idea 开发必备插件
阅读 1,856
- springboot项目整合阿里开源框架
EasyExcel, 附带整理好的Excel操作...
阅读 1,725
- java中使用jwt
阅读 104
- springboot + jta + mysql + mongo
分布式 (多种数据源) 事务
阅读 2,643
- HashMap原理解析
阅读 36



"如果觉得我的文章对您有用，请随意打赏。您的支持将鼓励我继续创作！"

赞赏支持

还没有人赞赏，支持一下



城下秋草 资深测试人，知乎ID：城下秋草
总资产72 (约6.52元) 共写了4.2W字 获得154个赞 共85个粉丝

关注

被以下专题收入，发现更多相似内容



测试、成长、路



软件测试

推荐阅读

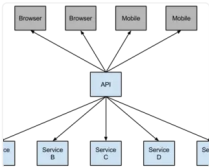
Spring Cloud

Spring Cloud为开发人员提供了快速构建分布式系统中一些常见模式的工具（例如配置管理，服务发现，断路器，智...



卡卡罗2017 阅读 81,839 评论 12 赞 122

更多精彩内容 >



Java面试宝典Beta5.0

pdf下载地址：Java面试宝典 第一章内容介绍 20 第二章JavaSE基础 21 一、Java面向对象 21 ...



王震阳 阅读 86,045 评论 27 赞 531

JMETER中文手册

JMETER中文手册 1. 简介 Apache JMeter是100%纯java桌面应用程序，被设计用来测试客户端...



捉虫师 阅读 13,207 评论 0 赞 23



写下你的评论...

评论0 赞4 ...

简书

首页

下载APP

搜索

Aa

beta

登录

注册

立师兄

阅读 4,199

评论 3

赞 45

Unit

0012

【墨竹的菜园】0012——几年前我有个梦想，我想在乡下建个带院子的房子，我美美的在墙角晒太阳睡觉，睡醒了到院子里抓...

墨竹的菜园

阅读 80

评论 0

赞 0

推荐阅读

idea 开发必备插件

阅读 1,856

springboot项目整合阿里开源框架EasyExcel，附带整理好的Excel操作...

阅读 1,725

java中使用jwt

阅读 104

springboot + jta + mysql + mongo 分布式（多种数据源）事务

阅读 2,643

HashMap原理解析

阅读 36

写下你的评论...

评论0

赞4