

SpringBoot单元测试

1.什么是单元测试？

单元测试（unit testing），是指对软件中的最小可测试单元进行检查和验证的过程就叫单元测试。

单元测试是开发者编写的一小段代码，用于检验被测代码的一个很小的、很明确的（代码）功能是否正确。执行单元测试就是为了证明某段代码的执行结果是否符合我们的预期。如果测试结果符合我们的预期，称之为测试通过，否则就是测试未通过（或者叫测试失败）。

2.单元测试有哪些好处？

- 1、可以非常简单、直观、快速的测试某一个功能是否正确。
- 2、使用单元测试可以帮我们在打包的时候，发现一些问题，因为在打包之前，所以的单元测试必须通过，否则不能打包成功。
- 3、使用单元测试，在测试功能的时候，可以不污染连接的数据库，也就是可以不对数据库进行任何改变的情况下，测试功能。

3.Spring Boot 单元测试使用

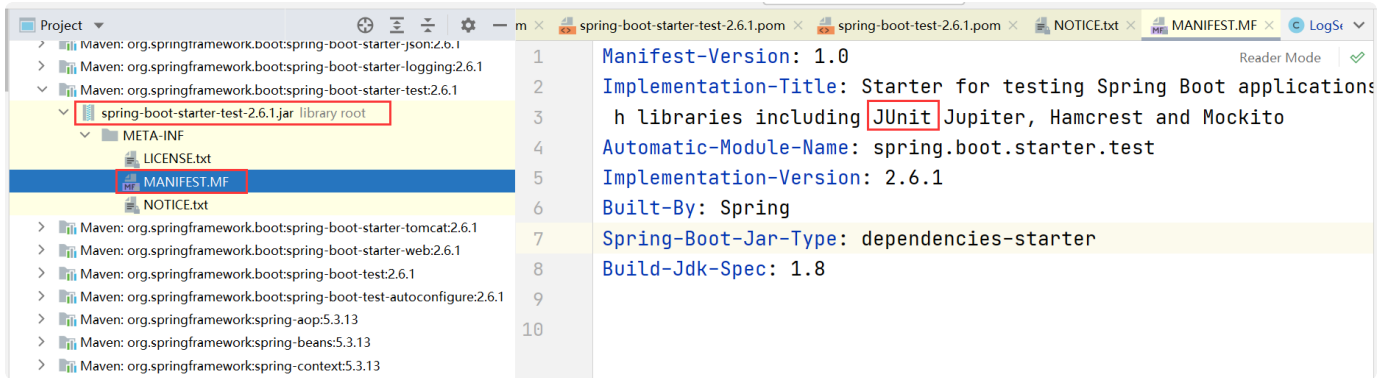
Spring Boot 项目创建时会默认单元测试框架 spring-boot-test，而这个单元测试框架主要是依靠另一个著名的测试框架 JUnit 实现的，打开 pom.xml 就可以看到，以下信息是 Spring Boot 项目创建是自动添加的：

▼

XML | 复制代码

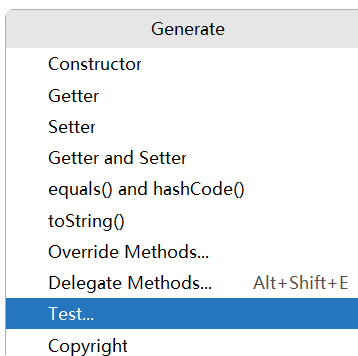
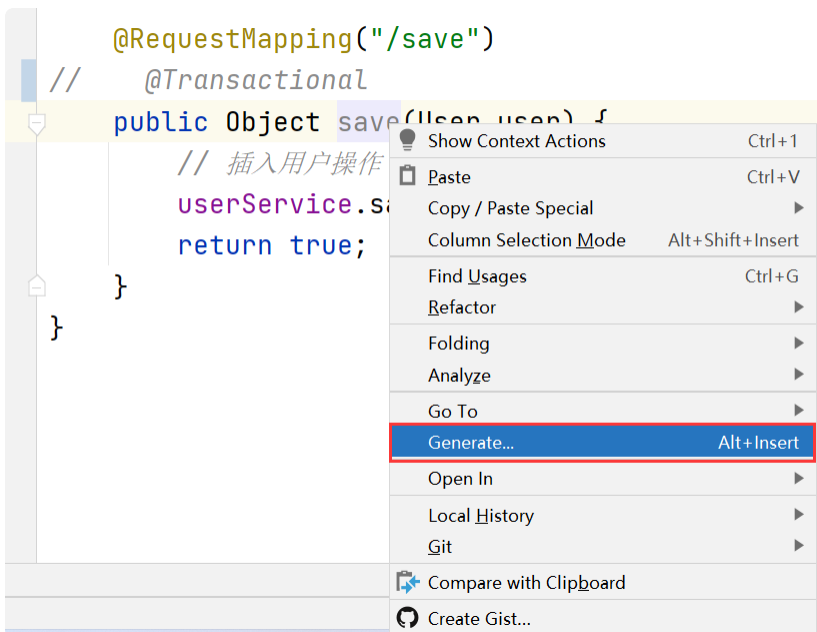
```
1 <dependency>
2   <groupId>org.springframework.boot</groupId>
3   <artifactId>spring-boot-starter-test</artifactId>
4   <scope>test</scope>
5 </dependency>
```

而 spring-boot-starter-test 的 MANIFEST.MF（Manifest 文件是用来定义扩展或档案打包的相关信息）里面有具体的说明，如下信息所示：



单元测试的实现步骤

1.生成单元测试类



Create Test

×

Testing library:

JUnit5

Class name:

UserControllerTest

Superclass:

...

Destination package:

com.example.demo.controller

...

Generate:

☐ setUp/@Before
☐ tearDown/@After

Generate test methods for:

☐ Show inherited methods

Member

☐ save(user:User):Object

?

OK

Cancel

Create Test

×

Testing library:

JUnit5

Class name:

UserControllerTest

Superclass:

...

Destination package:

com.example.demo.controller

...

Generate:

☐ setUp/@Before
☐ tearDown/@After

Generate test methods for:

☐ Show inherited methods

Member

☐ save(user:User):Object

?

OK

Cancel

最终生成的代码：

Java

复制代码

```

1  import org.junit.jupiter.api.Test;
2
3
4  class UserControllerTest {
5
6      @Test
7      void save() {
8
9      }
10 }
```

这个时候，此方法是不能调用到任何单元测试的方法的，此类只生成了单元测试的框架类，具体的业务代码要自己填充。

2 添加单元测试代码

1.添加 Spring Boot 框架测试注解：@SpringBootTest

Java | 复制代码

```
1  import org.junit.jupiter.api.Test;
2  import org.springframework.boot.test.context.SpringBootTest;
3
4  @SpringBootTest
5  class UserControllerTest {
6
7      @Test
8      void save() {
9
10     }
11 }
```

2.添加单元测试业务逻辑

```
1 import com.example.demo.model.User;
2 import org.junit.jupiter.api.Assertions;
3 import org.junit.jupiter.api.Test;
4 import org.springframework.boot.test.context.SpringBootTest;
5
6 import javax.annotation.Resource;
7
8 @SpringBootTest
9 class UserControllerTest {
10
11     @Resource
12     private UserController userController;
13
14     @Test
15     void save() {
16         User user = new User();
17         user.setId(1);
18         user.setName("Java Test");
19         user.setPassword("123");
20         boolean result = userController.save(user);
21         // 使用断言判断最终的结果是否符合预期
22         Assertions.assertEquals(true, result);
23         // Assertions.assertTrue(result);
24     }
25 }
```

简单的断言说明

方法	说明
assertEquals	判断两个对象或两个原始类型是否相等
assertNotEquals	判断两个对象或两个原始类型是否不相等
assertSame	判断两个对象引用是否指向同一个对象
assertNotSame	判断两个对象引用是否指向不同的对象
assertTrue	判断给定的布尔值是否为 true
assertFalse	判断给定的布尔值是否为 false
assertNull	判断给定的对象引用是否为 null
assertNotNull	判断给定的对象引用是否不为 null

断言：如果断言失败，则后面的代码都不会执行。

