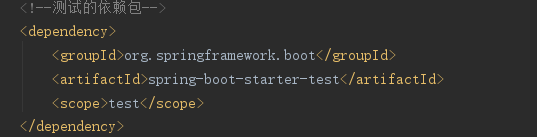
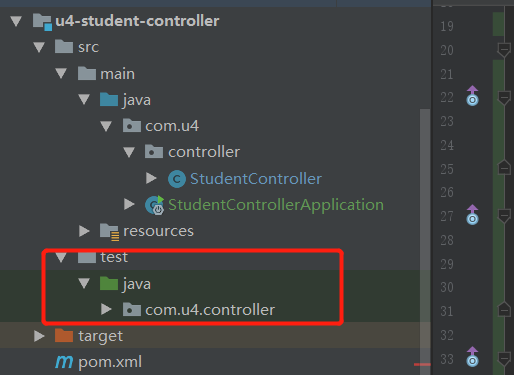
Junit4的使用

1. 首先我们引入pom.xml坐标，在这里是结合spring-boot的test测试，pom.xml坐标如下：



1. 如果是maven工程，那么必须写在test文件夹下。以利于maven工程发现。

如果是IDEA可以使用快捷键，ctrl+alt+T对此类进行快速生成测试类。

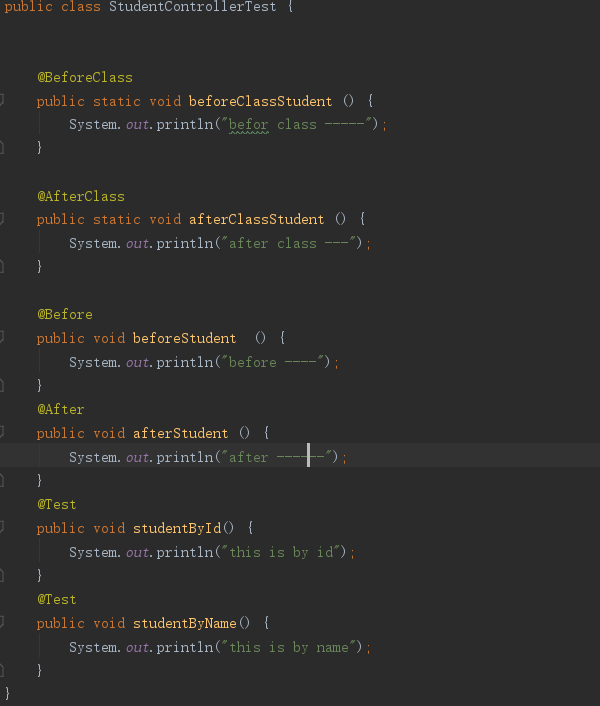
1. Junit4测试的API

@BeforeClass:在此类运行前运行，一般都是static修饰

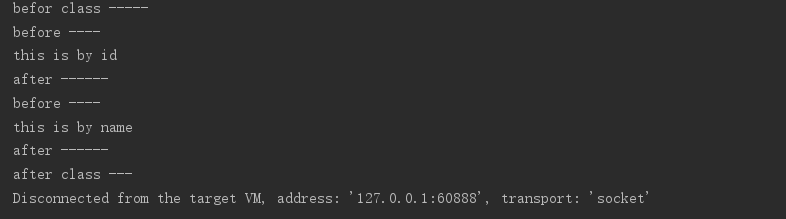
@AfterClass:在此类结束后运行，一般也是static修饰的

@Before：在类中每一个方法运行之前运行

@After：在类中每个方法运行之后运行



执行结果：



assertEquals(String msg, Object expect, Object actual);

当expect与actual相同的时候，那么无信息，不相同那么在控制台抛出异常，并且输出msg。

重载方法：

assertEquals(Object expect, Object actual);当不相同的时候，直接报异常

注意使用此方法的时候，我们要注意地址的使用情况。如果比较两个对象那么，需要重写其hashCode和equals方法

assertSame(String msg, var 1, var 2):

判断的是地址

assertTrue(String msg, A):

判断A返回的是否是真

assertFalse(String msg, B)

判断B返回是否是真

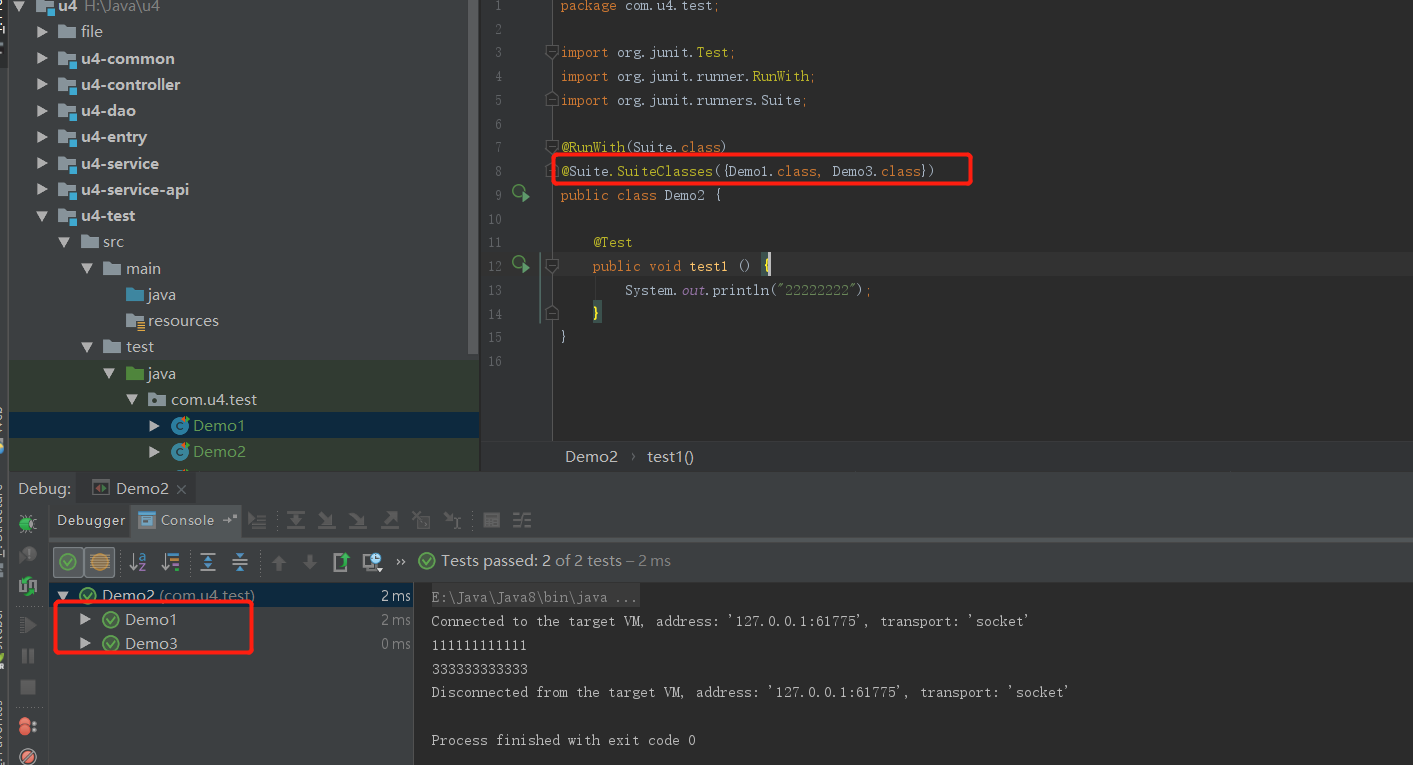
assertNotNull(String msg, A)

判断A不为空，如果为null，那么返回msg消息

assertArrayEquals(String msg, A, B)

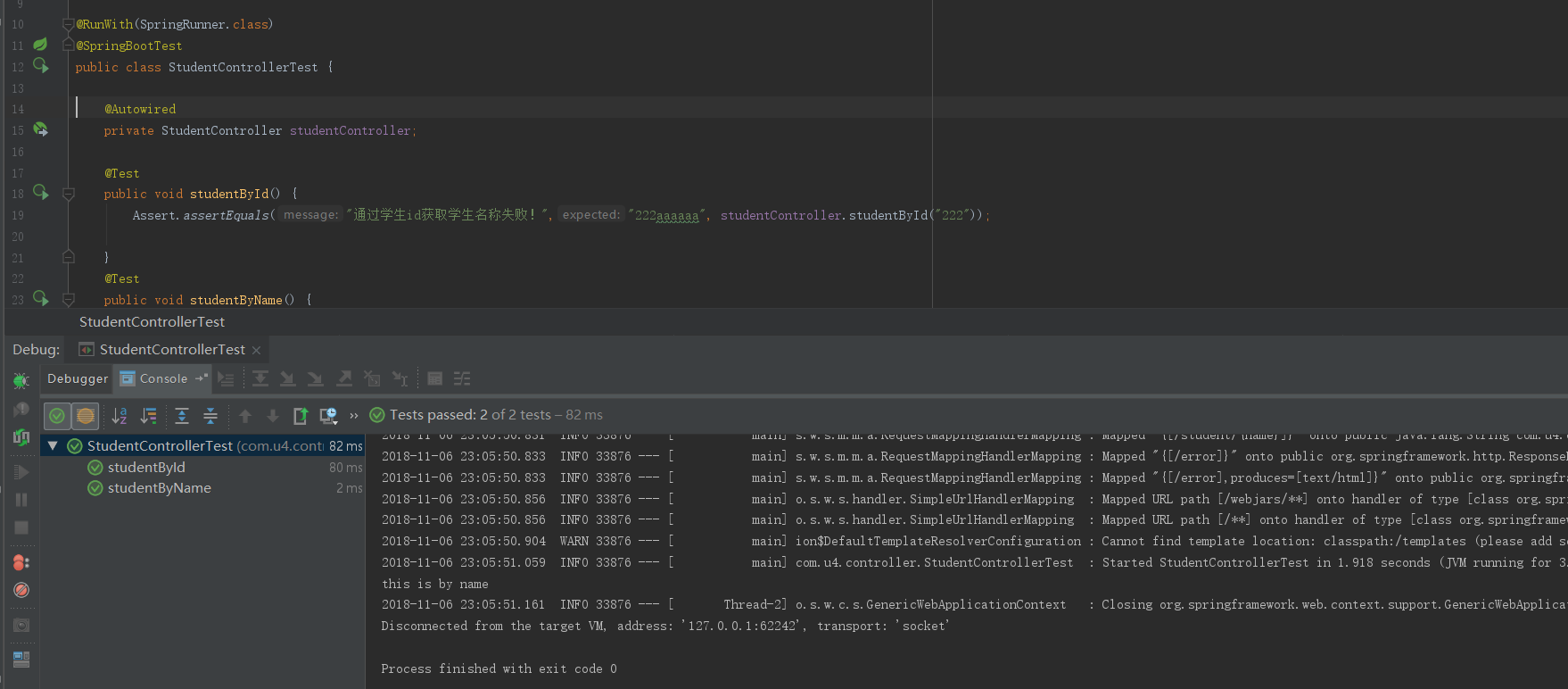
判断A和B的数组是否相同

@Suite:容器注解，[需要配合@RunWith(Suite.class)](mailto:需要配合@RunWith(Suite.class))此类是一个Suite容器，结合@SuiteClasses(Class[] clazzs) 一起使用，将clazzs数组放到Suite容器中一起运行



1. spring融合Junit4测试，首先加入注解，@RunWith(SpringRunner.class)告诉它使用spring测试的方式进行单元测试，其实也类似与一个spring的容器。@SpringBootTest会根据当前目录一直向上找，直到找到springboot的运行程序，也就是主函数。

我们可以直接加上这俩个注解直接进行注入进行单元测试。如下：



任何在spring容器中的对象都可以进行注入。

1. 测试spring mvc

首先在测试类上加入注解：@RunWith(SpringRunner.class)告诉使用spring的测试方式。

再次加入注解：@WebMvcTest(Class clazz)告诉是一个mvc的测试类。