链滴

Spock 使用教程 (二) 之与 Springboot 的整合

作者：hancaicai

原文链接：https://ld246.com/article/1573907697146

来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/bing/20181105.jpg?imageView2/1/w/960/h/540/interlace/1/q/100">
<h3 id="一-需求">一、需求</h3>
1、启动测试类的时候只加载相应的 java 类，而不是启动整个项目
2、可以 mock 相关的 java 类以及方法的返回值
<h3 id="二-实现">二、实现</h3>
<h4 id="2-1--准备">2.1 、准备</h4>
要测试的 java 类为 UserServiceImpl,依赖了 UserMapper 代码如下：
<pre><code class="highlight-chroma"><span class="highlight
cl">public class UserServiceImpl implements UserService {

    @Autowired
    private UserM
pper userMapper;
    @Override
    //根据用户ID查
用户信息
    public User sele
tUserById(Integer userId){
        User user=us
rMapper.seleteUserById(userId);
        return user;
    }
    //修改用户信息
    @Override
    public void upd
teUser(User user){
        userMapper.
pdateById(user);
    }
    //删除用户信息
    @Override
    public void dele
eUserById(Integer userId){
        userMapper.
eleteUserById(userId);
    }
    //添加用户信息
    public void inse
User(User user){
        user.setregis
erTime(new Date());
        userMapper.i
serUser(user);
    }
}

</code></pre>
<h4 id="2-2-测试代码如下-">2.2 测试代码如下：</h4>
<pre><code class="highlight-chroma"><span class="highlight
cl">@SpringBootTest
@ContextConfigu
ation(classes = UserServiceImpl.class)
@WebAppConfig

ration
class UserInfoSpec extends Specification {
    @Autowired
    UserServiceImpl userService
    @SpringBean
    UserMapper userMapper = Mock()
    def setup() {}
    @Unroll
    def "testSelectUserById"() {
        given:
        def user = new User(userId: userId, userName: userName, passWord: passWord);
        userMapper.deleteUserById(userId) >> user
        expect:
        userService.selectUserById(userId) == user
        where:
        userId | userName | passWord
        1234567L | xioming | 123456
        12309756 | xiohong | 4321134
    }
}

</code></pre>
<h3 id="三-代码讲解">三、代码讲解</h3>
<h4 id="3-1--SpringBootTest">3.1、@SpringBootTest</h4>
@SpringBootTest 注解告诉 SpringBoot 去寻找一个主配置类(例如带有 @SpringBootApplication 的配置类)，并使用它来启动 Spring 应用程序上下文。只用这个注解也是可以的，会自动注入所的 bean,我们不用 mock，缺点是当我们只测试一个类，也会加载完全无关的类，启动慢，浪费时间
<h4 id="3-2--ContextConfiguration">3.2、@ContextConfiguration</h4>
<code>@ContextConfiguration</code> loads an <code>ApplicationContext</code> for Spring integration test. <code>@ContextConfiguration</code> can load <code>ApplicationContext</code> using XML resource or the JavaConfig annotated with <code>@Configuration</code>. The <code>@ContextConfiguration</code> annotation can also load a component annotated with <code>@Component</code>, <code>@Service</code>, <code>@Repository</code> etc. We can also load classes annotated with <code>javax.inject</code>
简而言之就是使用 ContextConfiguration 注解可以只加载你想要加载的 bean,不用加载所有的 bean
官网用法链接
<h4 id="3-3---SpringBean">3.3、 @SpringBean</h4>
spock 支持的注解
<a href="https://ld246.com/forward?goto=https%3A%2F%2Fspockframework.org%2Fspock%2Fdocs%2F1.3%2Fmodule_spring.html" target="_blank" rel="nofollow ugc">官方文档解释</a

:
<code>Registers mock/stub/spy as a spring bean in the test context.To use </code>@Sprin
Bean <code>you have to use a strongly typed field</code> def <code>or</code> Object <
ode>won’ t work. You also need to directly assign the</code> Mock <code>/</code> Stub
<code>/</code> Spy <code>to the field using the standard Spock syntax. You can even use
he initializer blocks to define common behavior, however they are only picked up once they a
e attached to the</code> Specification <code>. </code>@SpringBean <code>definitions c
n replace existing Beans in your</code> ApplicationContext.
下一篇文章会解释 mock、stub 和 spy
3.4、@Unroll
spock 支持的注解和 where 一起用，相当于创建了多个 test
3.5、
<pre><code class="highlight-chroma"><span class="highlight
cl">def user = new User(userId: userId, userName: userName, passWord: passWord);
</code></pre>
spock 支持的 bean 的创建，也可以把参数直接换成常量，也支持 java 中的 bean 的实例化，
人觉得 spock 支持的比较好用，没有传的参数默认是 null.
<pre><code class="highlight-chroma"><span class="highlight
cl">userMapper.seleteUserById(userId) >> user
</code></pre>
相当于创建 mock 一个方法的返回值，当我们传的参数为 userId 时，userMapper.seleteUserBy
d 方法的返回值是 user，也可用_表示任何参数；
<pre><code class="highlight-chroma"><span class="highlight
cl">userMapper.seleteUserById(_) >> user
userMapper.selet
UserById(null) >> { throw new InternalError("ouch") }
userMapper.selet
UserById(!null)>>user
</code></pre>