**使用MockMvc测试Spring mvc Controller**

**一、MockMvc简介**

实现了对Http请求的模拟，能够直接使用网络的形式，转换到Controller的调用，这样可以使得测试速度快、不依赖网络环境，而且提供了一套验证的工具，这样可以使得请求的验证统一而且很方便。

**二、MockMvc测试常用注解**

1、@WebAppConfiguration

表明该类会使用web应用程序的默认根目录来载入ApplicationContext, 默认的更目录是"src/main/webapp", 如果需要更改这个更目录可以修改该注释的value值。使用这个Annotate会在跑单元测试的时候真实的启一个web服务，然后开始调用Controller的Rest API，待单元测试跑完之后再将web服务停掉。

图片包含 屏幕截图

描述已自动生成

2、@RunWith(SpringJUnit4ClassRunner.class):

表示使用Spring Test组件进行单元测试。

3、@ContextConfiguration(location = )

指定Bean的配置文件信息，可以有多种方式，这个例子使用的是文件路径形式，如果有多个配置文件，可以将括号中的信息配置为一个字符串数组来表示。

4、@Mock

需要被Mock的对象

5、@Transactional

不是必须的，是和@TestExecutionListeners中的

TransactionalTestExecutionListener.class配合使用，用于保证插入的数据库中的测试数据，在测试完后，事务回滚，将插入的数据给删除掉，保证数据库的干净。如果没有显示的指@Transactional，那么插入到数据库中的数据就是真实数据。

7、@InjectMocks

需要将Mock对象注入的对象, 此处就是Controller。

8、@Before

在每次Test方法之前运行的方法,目前把登陆信息放到session中处理，以及初始化mockMvc。

9、@Test

执行测试

**三、测试逻辑**

校验Controller处理之后，请求是否为成功状态，返回的内容是否包含了："{'foo':'bar'}"字符串，具体步骤如下：

1、 mockMvc调用perform，调用controller的业务处理逻辑；  
2 、perform返回ResultActions，返回操作结果，通过ResultActions，提供了统一的验证方式；  
3 、使用StatusResultMatchers对请求结果进行验证；  
4 、使用ContentResultMatchers对请求返回的内容进行验证。

**四、演示代码**

图片包含 屏幕截图

描述已自动生成