单元测试实践

1 测试的目标

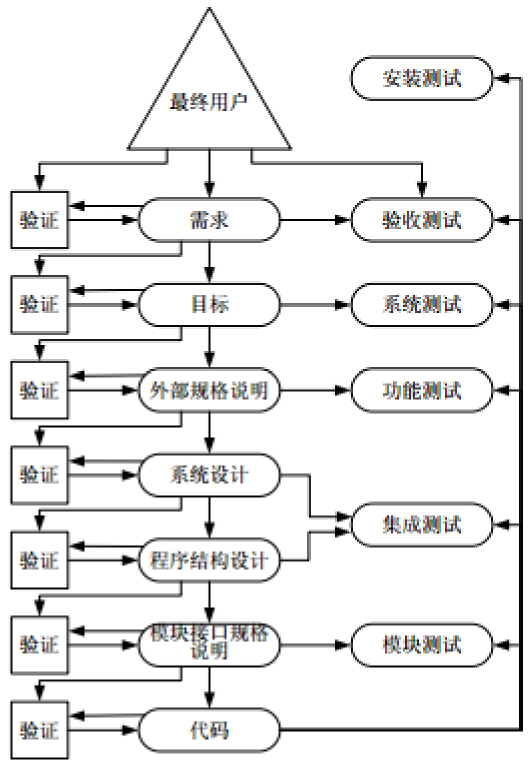
编写有效的测试保证代码的质量；对代码的质量能够度量，因为能够度量才能管理。

2 测试原则和理念

测试你的意图，而不是实际代码，已黑盒测试为主。
测试需要精心设计，它和开发一样重要。
测试和开发不需要同一个人，因为测试的是意图。
测试是提高了质量和降低了维护成本，而不是开发成本。
测试成本和时间需要进行权衡。
测试代码容易编写、容易执行、容易维护。
测试会增强你的信心，努力增强你的信心。

3 测试范围

3.1 测试的范围



- 模块测试的目的是发现程序模块与其接口规格说明之间的不一致。
- 功能测试的目的是为了证明程序未能符合其外部规格说明。
- 系统测试的目的是为了证明软件产品与其初始目标不一致。

单元测试、接口测试、集成测试、系统测试
这些测试只是粒度不一样，尽可能忽略这些复杂的定义，定义有效的测试，减少重复测试。
关注测试的有效性，而不是单元的粒度或者集成的规模。

3.2 常见错误

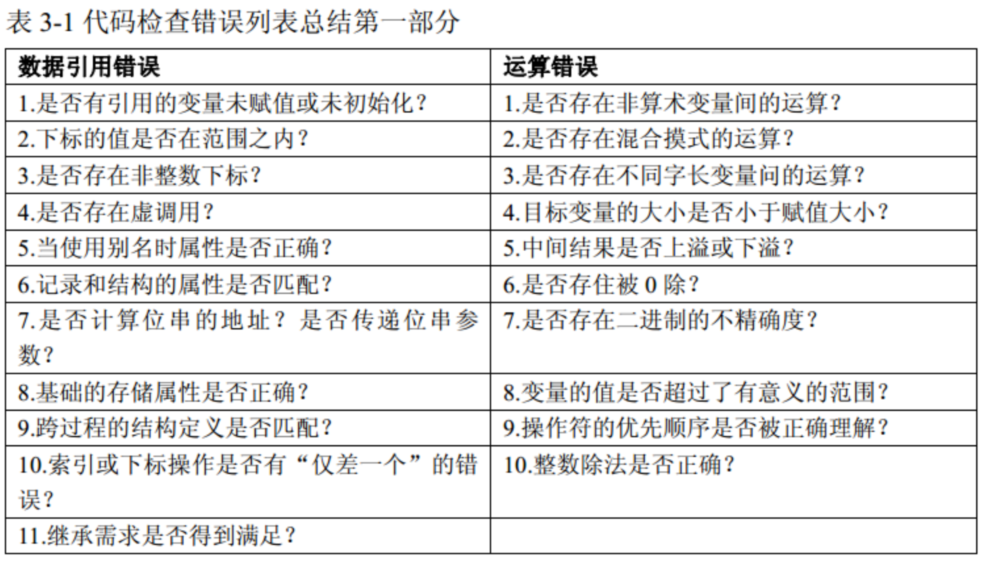


表 3-1 代码检查错误列表总结第一部分

数据引用错误	运算错误
1.是否有引用的变量未赋值或未初始化？	1.是否存在非算术变量间的运算？
2.下标的值是否在范围之内？	2.是否存在混合摸式的运算？
3.是否存在非整数下标？	3.是否存在不同字长变量间的运算？
4.是否存在虚调用？	4.目标变量的大小是否小于赋值大小？
5.当使用别名时属性是否正确？	5.中间结果是否上溢或下溢？
6.记录和结构的属性是否匹配？	6.是否存住被 0 除？
7.是否计算位串的地址？是否传递位串参数？	7.是否存在二进制的不精确度？
8.基础的存储属性是否正确？	8.变量的值是否超过了有意义的范围？
9.跨过程的结构定义是否匹配？	9.操作符的优先顺序是否被正确理解？
10.索引或下标操作是否有“仅差一个”的错误？	10.整数除法是否正确？
11.继承需求是否得到满足？	

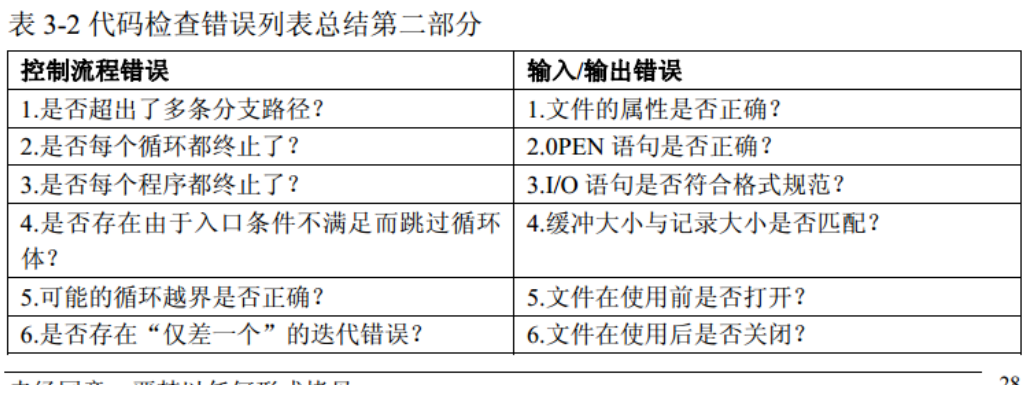


表 3-2 代码检查错误列表总结第二部分

控制流程错误	输入/输出错误
1.是否超出了多条分支路径？	1.文件的属性是否正确？
2.是否每个循环都终止了？	2.0PEN 语句是否正确？
3.是否每个程序都终止了？	3.I/O 语句是否符合格式规范？
4.是否存在由于入口条件不满足而跳过循环体？	4.缓冲大小与记录大小是否匹配？
5.可能的循环越界是否正确？	5.文件在使用前是否打开？
6.是否存在“仅差一个”的迭代错误？	6.文件在使用后是否关闭？

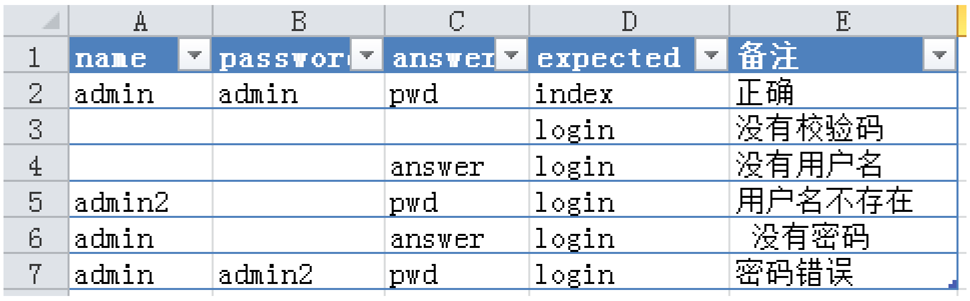
3.3 容易出错的测试点

输入最复杂
第三方的输出都要谨慎处理
第三方接口
并发
异常
关键点性能

4 测试方法和测试用例设计

4.1 好的方式

优先使用黑盒测试，白盒测试作为补充。
自上而下测试，还是自下而上的测试，根据情况选择。
认真对待输入和输出，测试用例需要覆盖判定条件。



	A	B	C	D	E
1	name	passwor	answer	expected	备注
2	admin	admin	pwd	index	正确
3				login	没有校验码
4			answer	login	没有用户名
5	admin2		pwd	login	用户名不存在
6	admin		answer	login	没有密码
7	admin	admin2	pwd	login	密码错误

4.2 常见的错误方式

没有自动测试
发现问题通常通过 debug 调试
有测试经常打印出来，需要人工对比。
好像通了，实际上没有严禁的测试用例

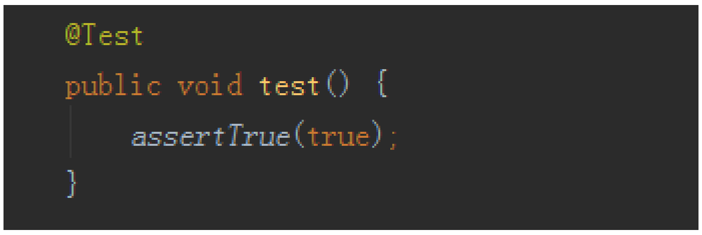
5 测试实践

本文的示例代码在:
https://github.com/superproxy/sample-test

5.1 基本测试

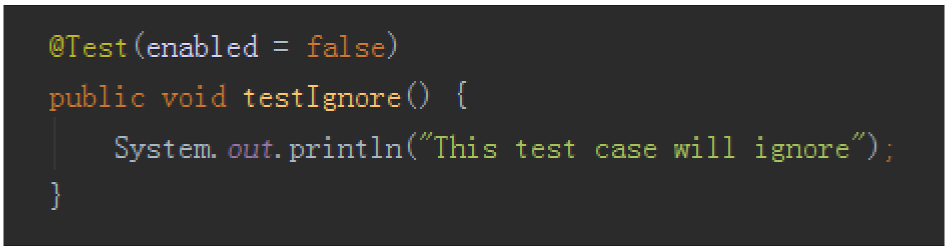
我们以 testng 为基本的测试工具，testng 提供了断言方式，如果没有满足条件，则抛出异常。
参考资料
Testng 官网 http://testng.org/
Testng 教程 http://testng.org/doc/documentation-main.html

5.1.1　简单测试



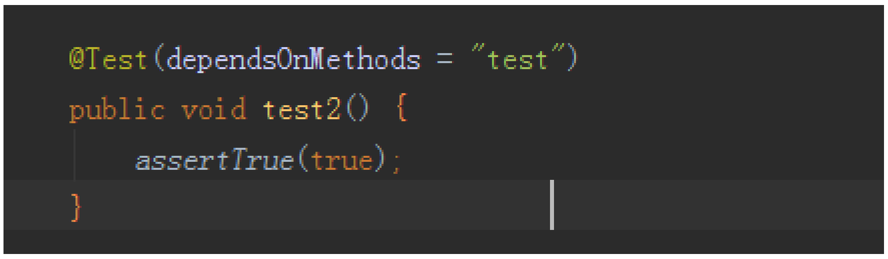
```java
@Test
public void test() {
    assertTrue(true);
}
```

5.1.2　忽略测试



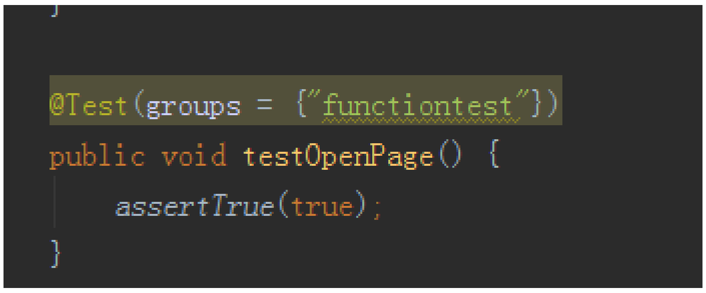
```java
@Test(enabled = false)
public void testIgnore() {
    System.out.println("This test case will ignore");
}
```

5.1.3 依赖测试



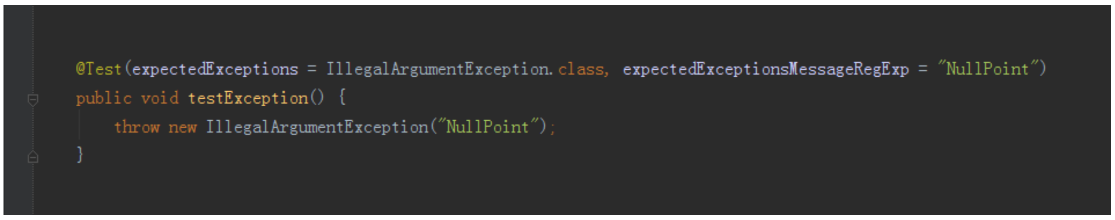
```java
@Test(dependsOnMethods = "test")
public void test2() {
    assertTrue(true);
}
```

5.1.4 分组



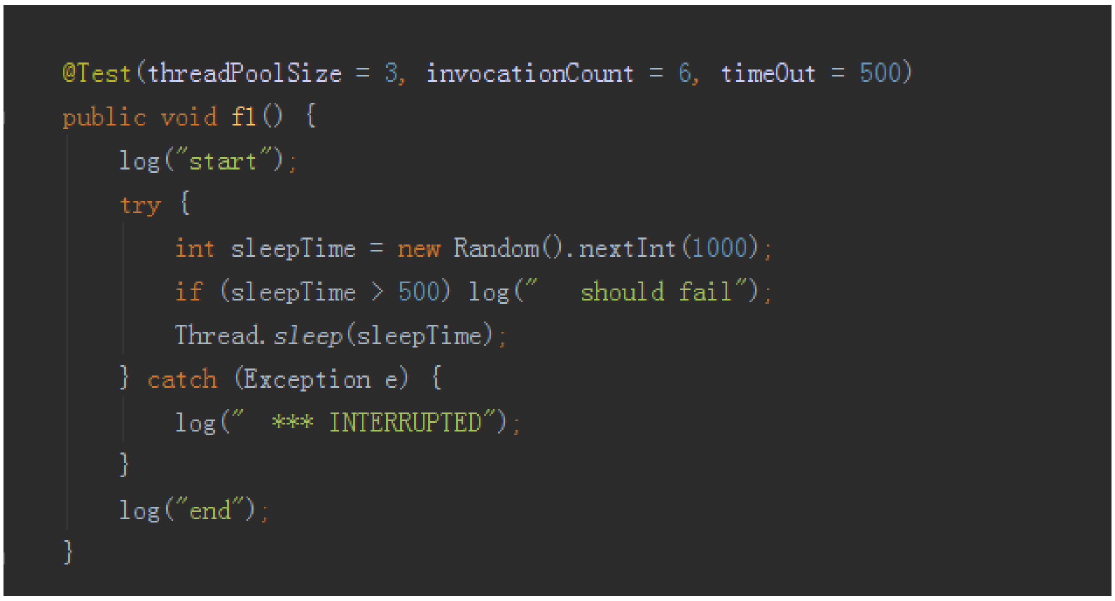
```java
}

@Test(groups = {"functiontest"})
public void testOpenPage() {
    assertTrue(true);
}
```

5.1.5 异常



```java
@Test(expectedExceptions = IllegalArgumentException.class, expectedExceptionsMessageRegExp = "NullPoint")
public void testException() {
    throw new IllegalArgumentException("NullPoint");
}
```

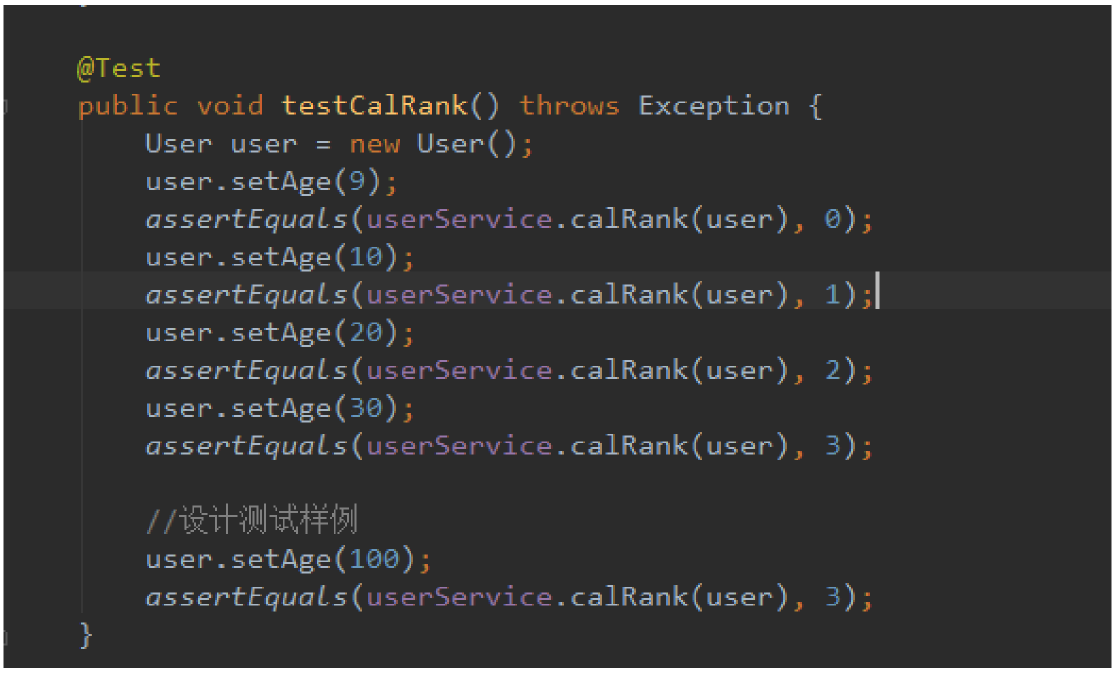
5.1.6 多线程



```java
@Test(threadPoolSize = 3, invocationCount = 6, timeOut = 500)
public void f1() {
    log("start");
    try {
        int sleepTime = new Random().nextInt(1000);
        if (sleepTime > 500) log("   should fail");
        Thread.sleep(sleepTime);
    } catch (Exception e) {
        log("  *** INTERRUPTED");
    }
    log("end");
}
```

5.2　数据集测试

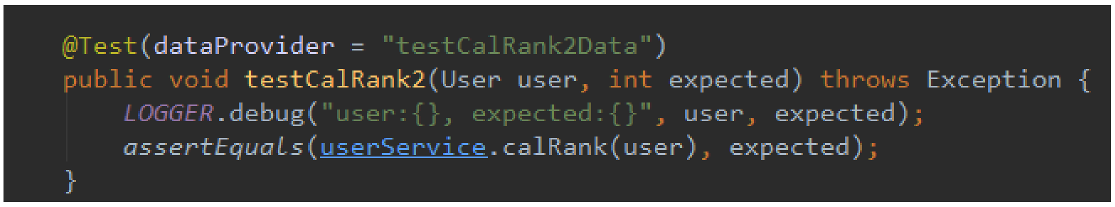
5.2.1 方式一



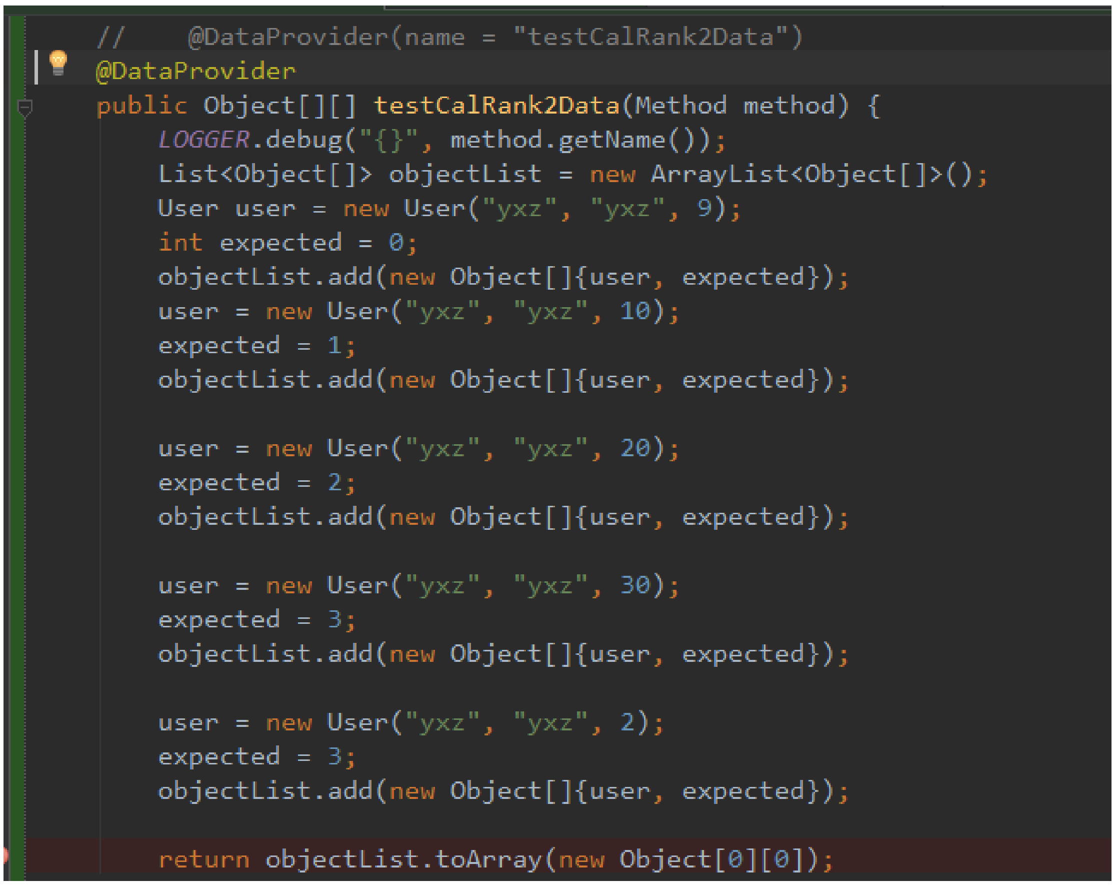
```java
@Test
public void testCalRank() throws Exception {
    User user = new User();
    user.setAge(9);
    assertEquals(userService.calRank(user), 0);
    user.setAge(10);
    assertEquals(userService.calRank(user), 1);
    user.setAge(20);
    assertEquals(userService.calRank(user), 2);
    user.setAge(30);
    assertEquals(userService.calRank(user), 3);

    //设计测试样例
    user.setAge(100);
    assertEquals(userService.calRank(user), 3);
}
```

5.2.2 方式二

使用 testng 提供的 dataProvider 方式



```java
@Test(dataProvider = "testCalRank2Data")
public void testCalRank2(User user, int expected) throws Exception {
    LOGGER.debug("user:{}, expected:{}", user, expected);
    assertEquals(userService.calRank(user), expected);
}
```



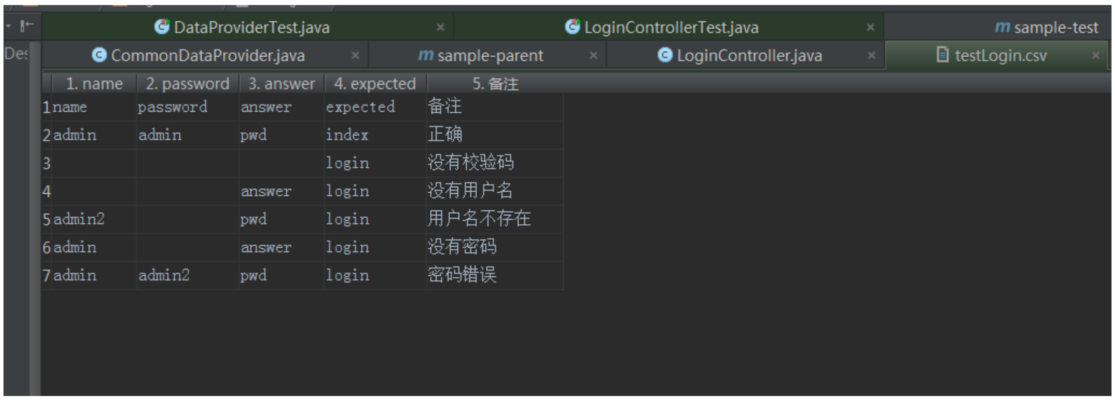
```java
//    @DataProvider(name = "testCalRank2Data")
@DataProvider
public Object[][] testCalRank2Data(Method method) {
    LOGGER.debug("{}", method.getName());
    List<Object[]> objectList = new ArrayList<Object[]>();
    User user = new User("yxz", "yxz", 9);
    int expected = 0;
    objectList.add(new Object[]{user, expected});
    user = new User("yxz", "yxz", 10);
    expected = 1;
    objectList.add(new Object[]{user, expected});

    user = new User("yxz", "yxz", 20);
    expected = 2;
    objectList.add(new Object[]{user, expected});

    user = new User("yxz", "yxz", 30);
    expected = 3;
    objectList.add(new Object[]{user, expected});

    user = new User("yxz", "yxz", 2);
    expected = 3;
    objectList.add(new Object[]{user, expected});

    return objectList.toArray(new Object[0][0]);
```

5.2.3 方式三

使用 jar 包 test-data-provider，支持从 csv，json 文件中读取文件内容。

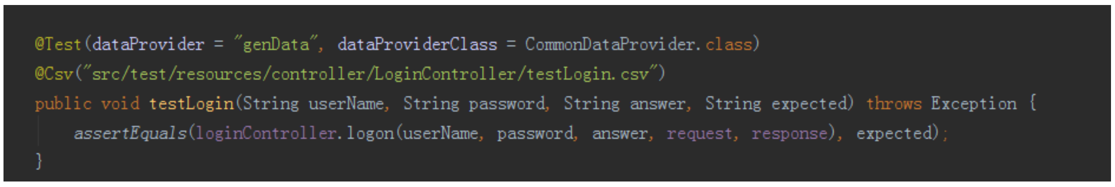
```xml
<dependency>
	<groupId>com.github.superproxy</groupId>
	<artifactId>test-data-provider</artifactId>
	<version>0.1.0</version>
</dependency>
```

1. 编写文件 csv 文件，csv 文件最好使用 utf-8



	1. name	2. password	3. answer	4. expected	5. 备注
1	name	password	answer	expected	备注
2	admin	admin	pwd	index	正确
3				login	没有校验码
4			answer	login	没有用户名
5	admin2		pwd	login	用户名不存在
6	admin		answer	login	没有密码
7	admin	admin2	pwd	login	密码错误

2.使用 Csv 注解，然后设置文件的路径，目前不支持默认路径



```java
@Test(dataProvider = "genData", dataProviderClass = CommonDataProvider.class)
@Csv("src/test/resources/controller/LoginController/testLogin.csv")
public void testLogin(String userName, String password, String answer, String expected) throws Exception {
    assertEquals(loginController.logon(userName, password, answer, request, response), expected);
}
```

5.3 Mock 模拟测试

mock 测试就是在测试过程中，对于某些不容易构造或者 不容易获取的对象，用一个虚拟的对象来创建以便测试的测试方法。

Mock 测试框架推荐使用 mockito。

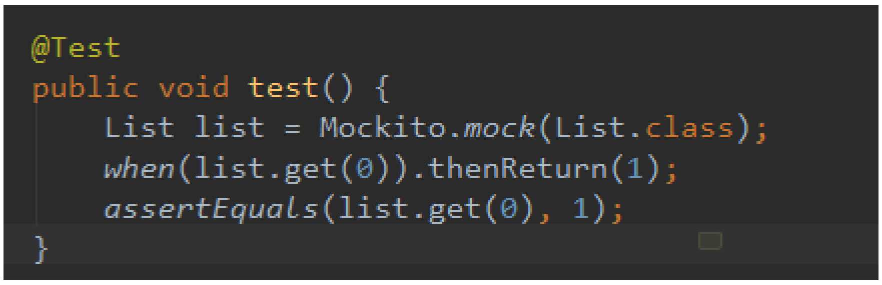
参考资料：

Mockito 官网　http://site.mockito.org/

Mockito 教程　https://github.com/mockito/mockito/wiki/FAQ

5.3.1 Mock 基本使用方式

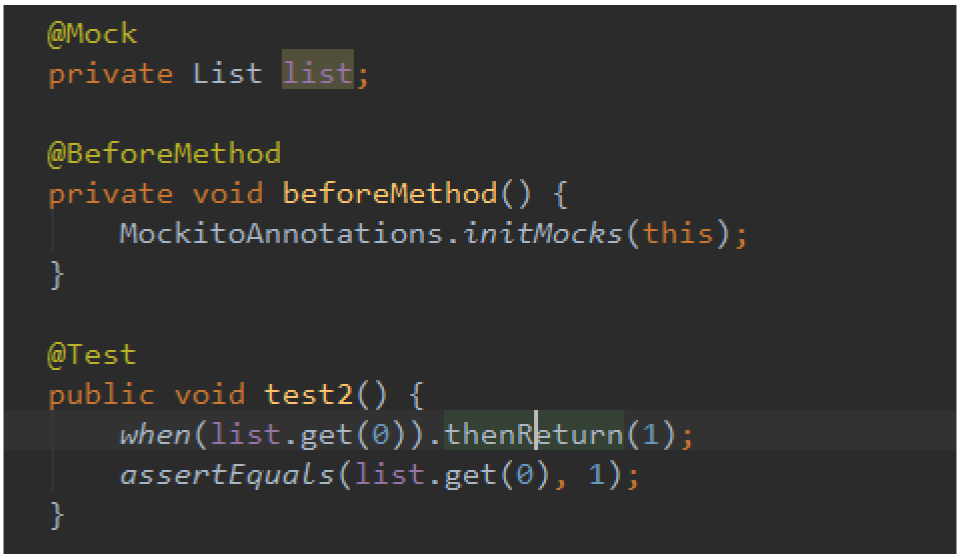
5.3.1.1方式 1



```java
@Test
public void test() {
    List list = Mockito.mock(List.class);
    when(list.get(0)).thenReturn(1);
    assertEquals(list.get(0), 1);
}
```

上面的示例模拟了 jdk 提供的集合接口 List。当调用 list.get(0)，获取集合中第一个元素，返回 1。

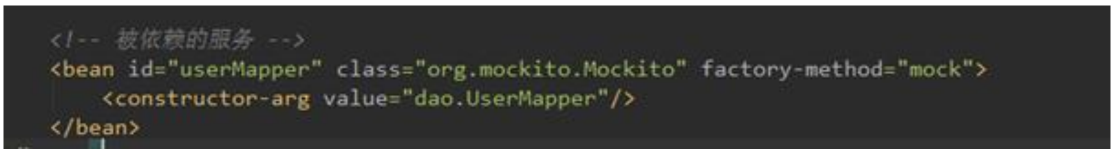
5.3.1.2方式 2



```java
@Mock
private List list;

@BeforeMethod
private void beforeMethod() {
    MockitoAnnotations.initMocks(this);
}

@Test
public void test2() {
    when(list.get(0)).thenReturn(1);
    assertEquals(list.get(0), 1);
}
```

5.3.1.3方式 3

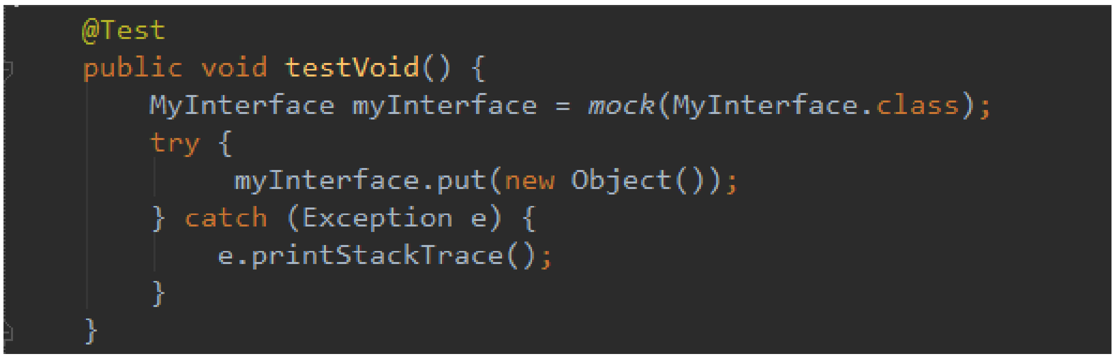
Spring 配置



```xml
<!-- 被依赖的服务 -->
<bean id="userMapper" class="org.mockito.Mockito" factory-method="mock">
    <constructor-arg value="dao.UserMapper"/>
</bean>
```

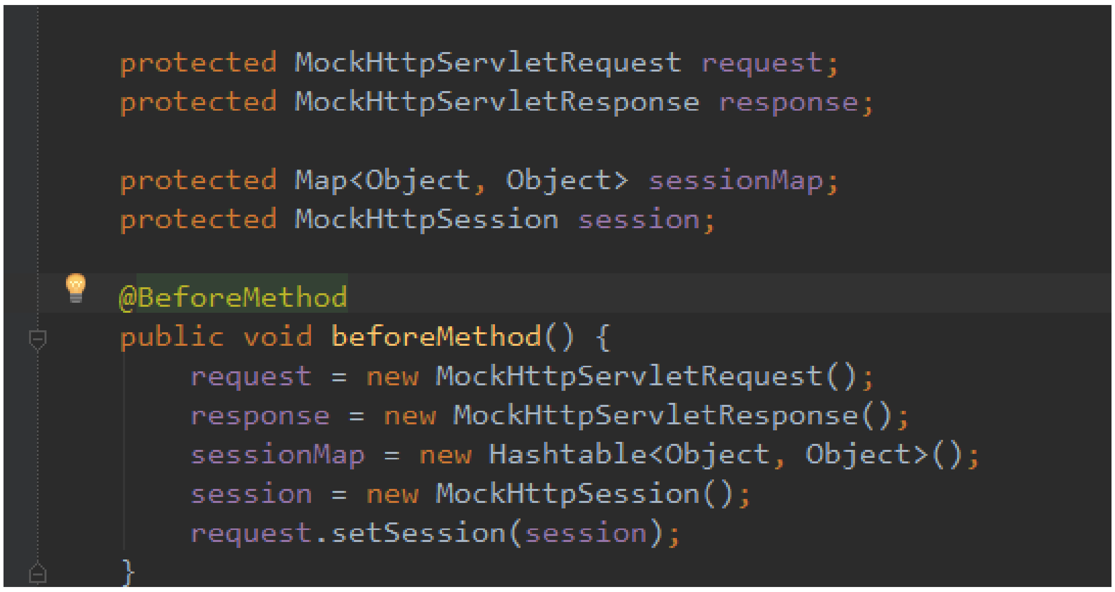
5.3.2 Mock void 返回

直接调用方法，没有异常。



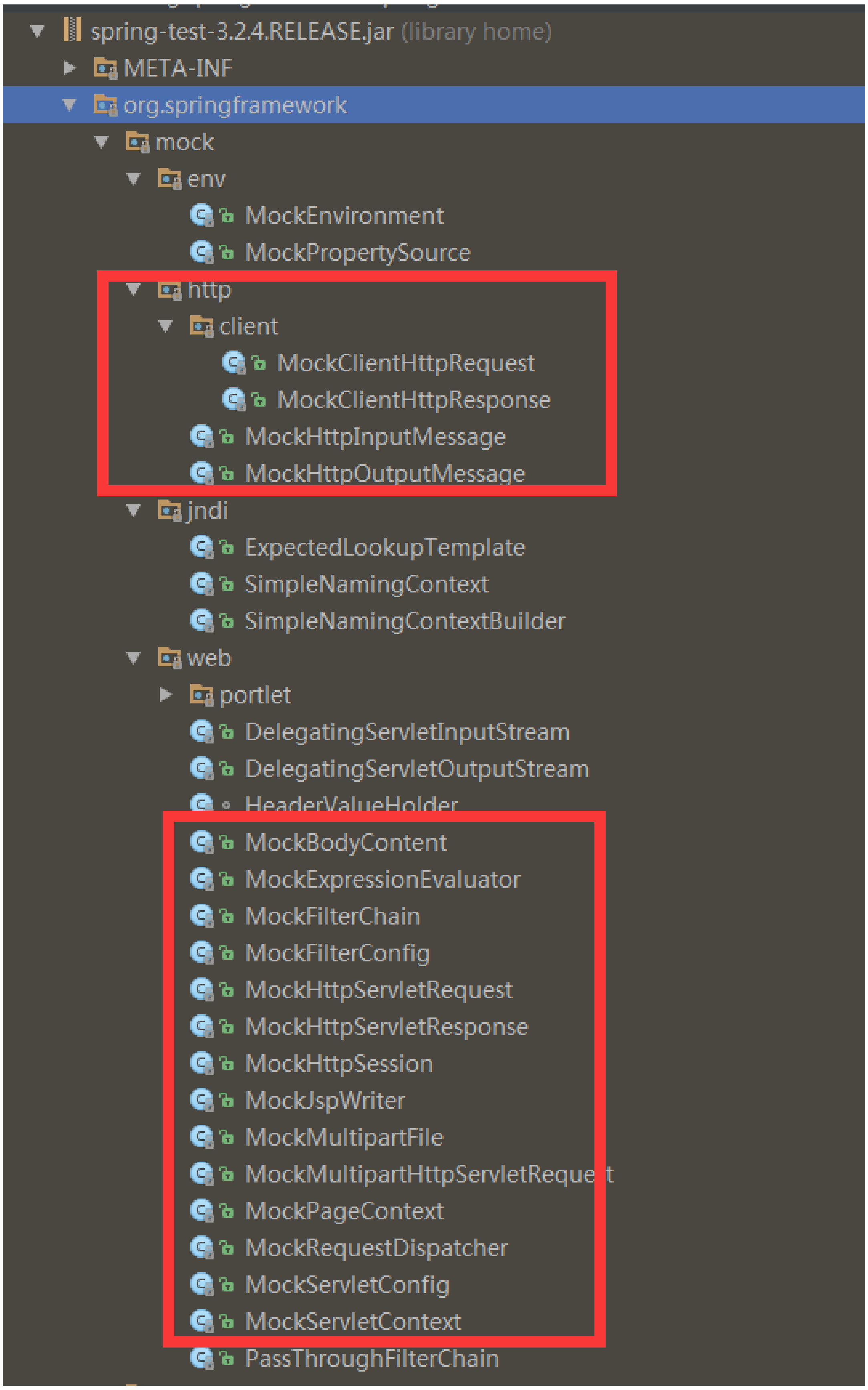
```java
@Test
public void testVoid() {
    MyInterface myInterface = mock(MyInterface.class);
    try {
        myInterface.put(new Object());
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

5.3.3 重量级对象模拟



```java
    protected MockHttpServletRequest request;
    protected MockHttpServletResponse response;

    protected Map<Object, Object> sessionMap;
    protected MockHttpSession session;

    @BeforeMethod
    public void beforeMethod() {
        request = new MockHttpServletRequest();
        response = new MockHttpServletResponse();
        sessionMap = new Hashtable<Object, Object>();
        session = new MockHttpSession();
        request.setSession(session);
    }
```

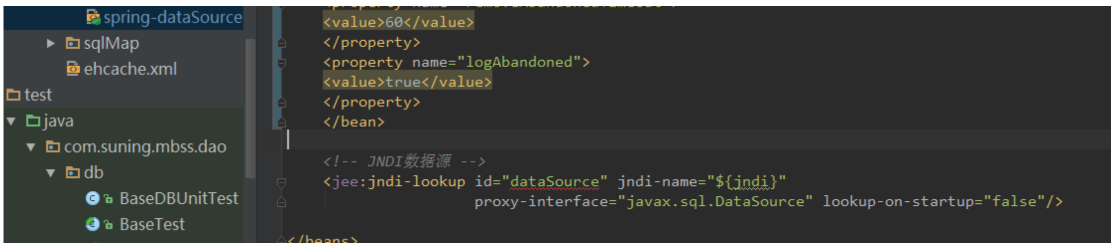
Spring TEST 支持的 MOCK 对象



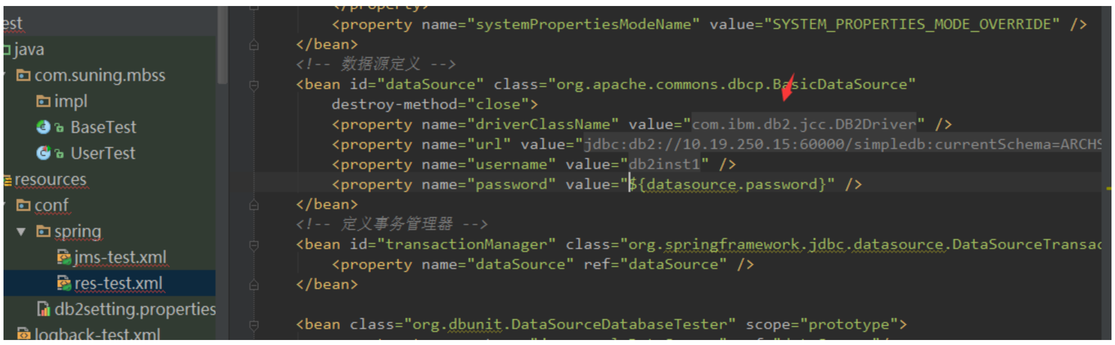
```
▼ spring-test-3.2.4.RELEASE.jar (library home)
    ▶ META-INF
    ▼ org.springframework
        ▼ mock
            ▼ env
                MockEnvironment
                MockPropertySource
            ▼ http
                ▼ client
                    MockClientHttpRequest
                    MockClientHttpResponse
                MockHttpInputMessage
                MockHttpOutputMessage
            ▼ jndi
                ExpectedLookupTemplate
                SimpleNamingContext
                SimpleNamingContextBuilder
            ▼ web
                ▶ portlet
                DelegatingServletInputStream
                DelegatingServletOutputStream
                HeaderValueHolder
                MockBodyContent
                MockExpressionEvaluator
                MockFilterChain
                MockFilterConfig
                MockHttpServletRequest
                MockHttpServletResponse
                MockHttpSession
                MockJspWriter
                MockMultipartFile
                MockMultipartHttpServletRequest
                MockPageContext
                MockRequestDispatcher
                MockServletConfig
                MockServletContext
                PassThroughFilterChain
```

5.3.1 MOCK JNDI

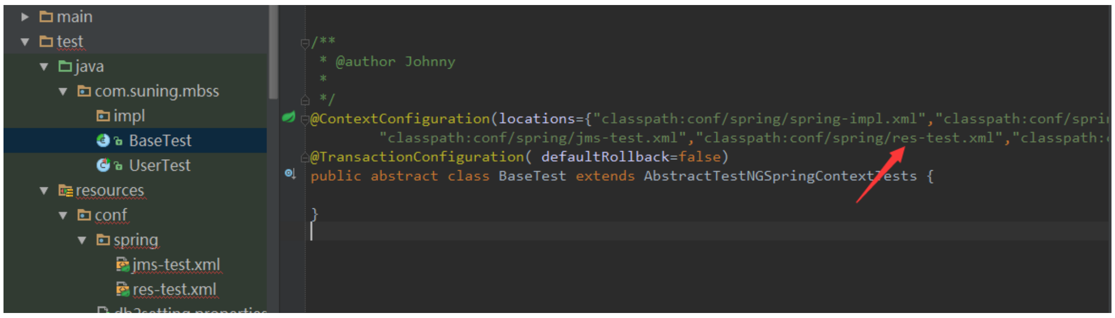
通过不同的配置文件加载不同的资源



```xml
        <value>60</value>
        </property>
        <property name="logAbandoned">
        <value>true</value>
        </property>
        </bean>

        <!-- JNDI数据源 -->
        <jee:jndi-lookup id="dataSource" jndi-name="${jndi}"
                         proxy-interface="javax.sql.DataSource" lookup-on-startup="false"/>

</beans>
```



```xml
        <property name="systemPropertiesModeName" value="SYSTEM_PROPERTIES_MODE_OVERRIDE" />
    </bean>
    <!-- 数据源定义 -->
    <bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"
        destroy-method="close">
        <property name="driverClassName" value="com.ibm.db2.jcc.DB2Driver" />
        <property name="url" value="jdbc:db2://10.19.250.15:60000/simpledb:currentSchema=ARCHS
        <property name="username" value="db2inst1" />
        <property name="password" value="${datasource.password}" />
    </bean>
    <!-- 定义事务管理器 -->
    <bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransac
        <property name="dataSource" ref="dataSource" />
    </bean>

    <bean class="org.dbunit.DataSourceDatabaseTester" scope="prototype">
```



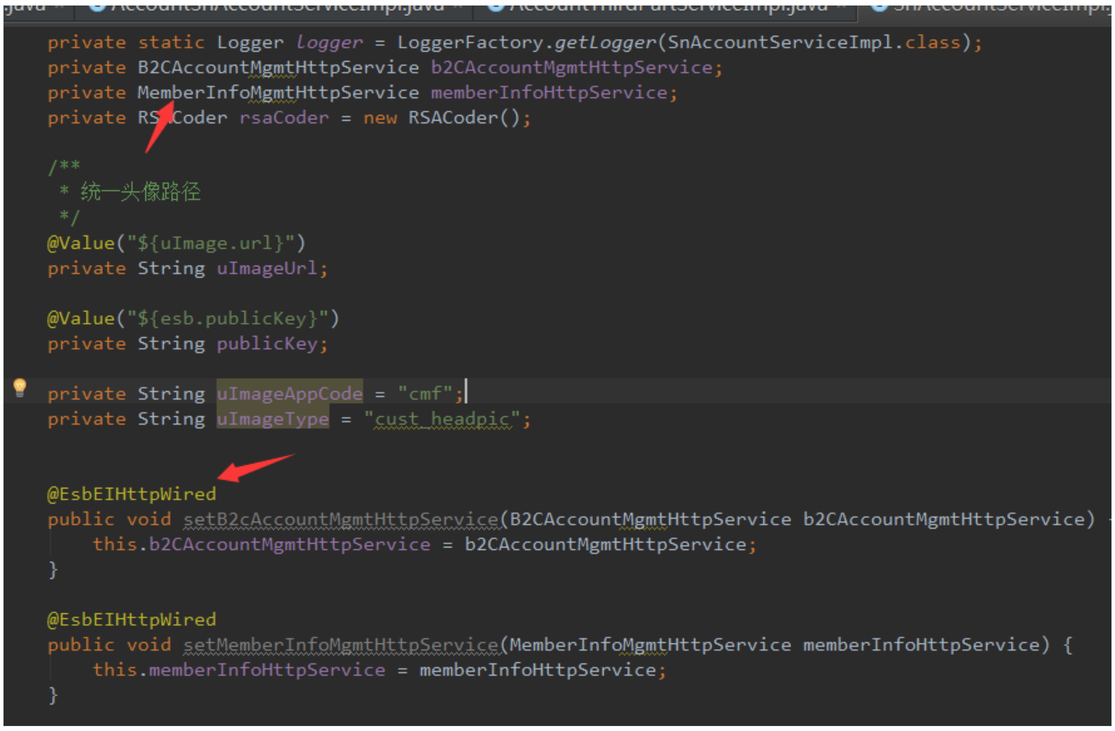
```java
/**
 * @author Johnny
 *
 */
@ContextConfiguration(locations={"classpath:conf/spring/spring-impl.xml","classpath:conf/spri
        "classpath:conf/spring/jms-test.xml","classpath:conf/spring/res-test.xml","classpath:
@TransactionConfiguration( defaultRollback=false)
public abstract class BaseTest extends AbstractTestNGSpringContextTests {

}
```

5.3.2 Mock EJB

参考

http://mockejb.sourceforge.net/

5.3.3 Mock ESB

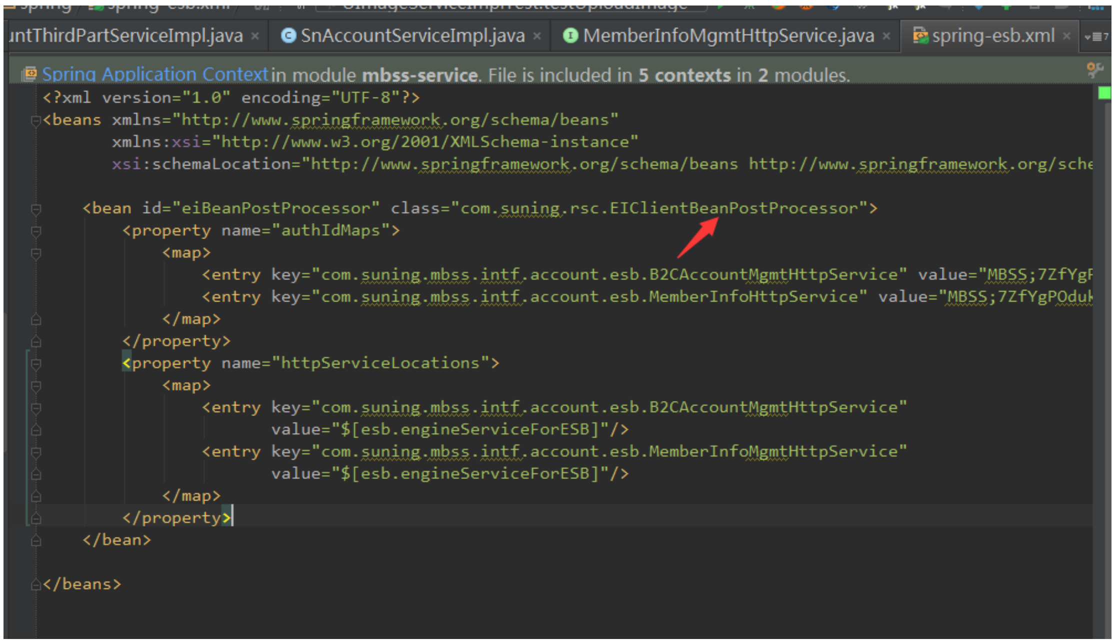


```java
    private static Logger logger = LoggerFactory.getLogger(SnAccountServiceImpl.class);
    private B2CAccountMgmtHttpService b2CAccountMgmtHttpService;
    private MemberInfoMgmtHttpService memberInfoHttpService;
    private RSACoder rsaCoder = new RSACoder();

    /**
     * 统一头像路径
     */
    @Value("${uImage.url}")
    private String uImageUrl;

    @Value("${esb.publicKey}")
    private String publicKey;

    private String uImageAppCode = "cmf";
    private String uImageType = "cust_headpic";


    @EsbEIHttpWired
    public void setB2cAccountMgmtHttpService(B2CAccountMgmtHttpService b2CAccountMgmtHttpService)
        this.b2CAccountMgmtHttpService = b2CAccountMgmtHttpService;
    }

    @EsbEIHttpWired
    public void setMemberInfoMgmtHttpService(MemberInfoMgmtHttpService memberInfoHttpService) {
        this.memberInfoHttpService = memberInfoHttpService;
    }
```

使用的注解，直接 mock 就可以了

service.setHttpService(Mockito.mock(B2CAccountMgmtHttpService.class));

配置文件单独配置

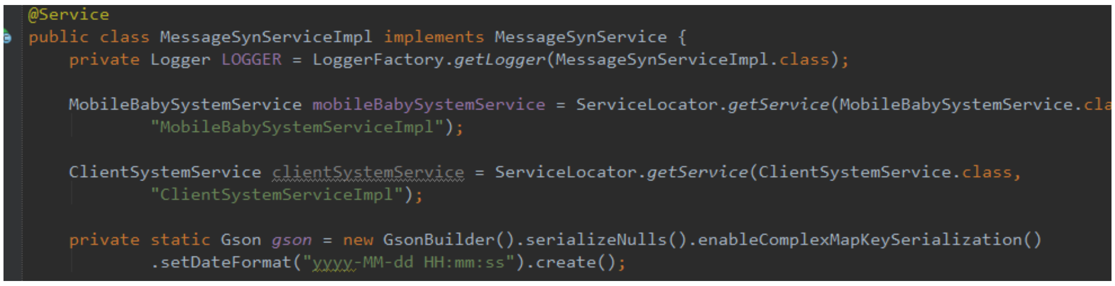


untThirdPartServiceImpl.java × SnAccountServiceImpl.java × MemberInfoMgmtHttpService.java × spring-esb.xml ×

Spring Application Context in module **mbss-service**. File is included in **5 contexts** in **2** modules.

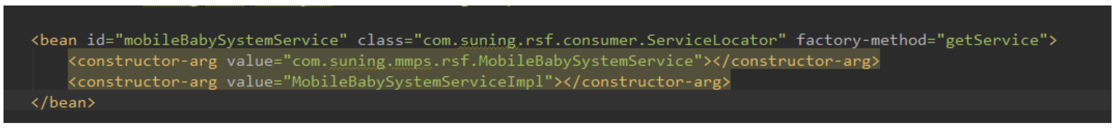
```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/sche

    <bean id="eiBeanPostProcessor" class="com.suning.rsc.EIClientBeanPostProcessor">
        <property name="authIdMaps">
            <map>
                <entry key="com.suning.mbss.intf.account.esb.B2CAccountMgmtHttpService" value="MBSS;7ZfYgP
                <entry key="com.suning.mbss.intf.account.esb.MemberInfoHttpService" value="MBSS;7ZfYgPOduk
            </map>
        </property>
        <property name="httpServiceLocations">
            <map>
                <entry key="com.suning.mbss.intf.account.esb.B2CAccountMgmtHttpService"
                       value="$[esb.engineServiceForESB]"/>
                <entry key="com.suning.mbss.intf.account.esb.MemberInfoMgmtHttpService"
                       value="$[esb.engineServiceForESB]"/>
            </map>
        </property>
    </bean>

</beans>
```

5.3.4 Mock RSF

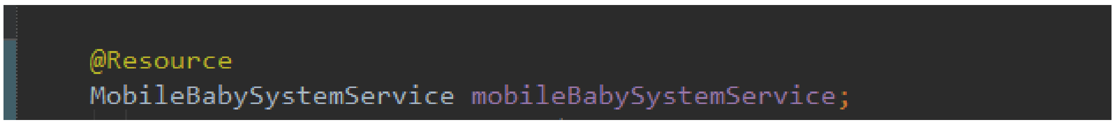


```java
@Service
public class MessageSynServiceImpl implements MessageSynService {
    private Logger LOGGER = LoggerFactory.getLogger(MessageSynServiceImpl.class);

    MobileBabySystemService mobileBabySystemService = ServiceLocator.getService(MobileBabySystemService.cla
            "MobileBabySystemServiceImpl");

    ClientSystemService clientSystemService = ServiceLocator.getService(ClientSystemService.class,
            "ClientSystemServiceImpl");

    private static Gson gson = new GsonBuilder().serializeNulls().enableComplexMapKeySerialization()
            .setDateFormat("yyyy-MM-dd HH:mm:ss").create();
```

改为



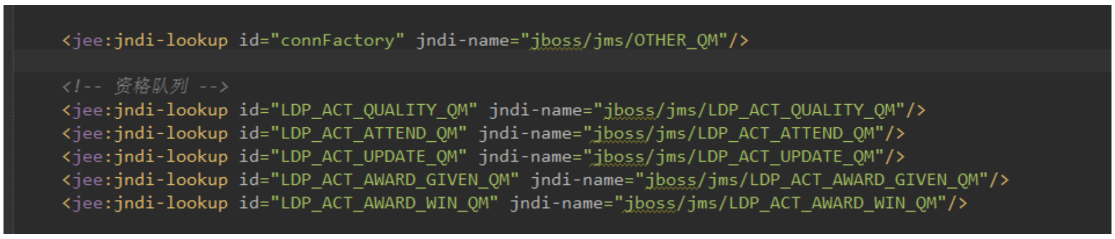
```xml
<bean id="mobileBabySystemService" class="com.suning.rsf.consumer.ServiceLocator" factory-method="getService">
    <constructor-arg value="com.suning.mmps.rsf.MobileBabySystemService"></constructor-arg>
    <constructor-arg value="MobileBabySystemServiceImpl"></constructor-arg>
</bean>
```



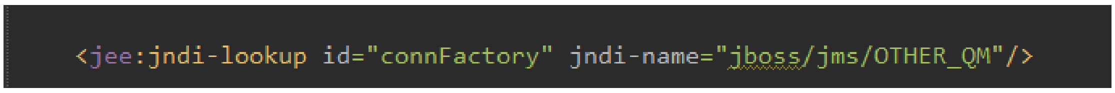
```java
@Resource
MobileBabySystemService mobileBabySystemService;
```

如果 Rsf 使用 @Reference 引用
使用的注解，直接 mock 就可以了
service.setRsfService(Mockito.mock(RsfService.class));

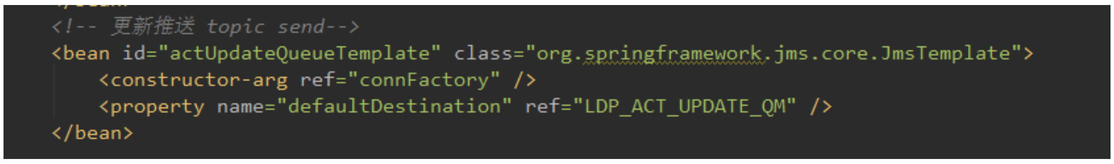
5.3.5 Mock MQ



```xml
<jee:jndi-lookup id="connFactory" jndi-name="jboss/jms/OTHER_QM"/>

<!-- 资格队列 -->
<jee:jndi-lookup id="LDP_ACT_QUALITY_QM" jndi-name="jboss/jms/LDP_ACT_QUALITY_QM"/>
<jee:jndi-lookup id="LDP_ACT_ATTEND_QM" jndi-name="jboss/jms/LDP_ACT_ATTEND_QM"/>
<jee:jndi-lookup id="LDP_ACT_UPDATE_QM" jndi-name="jboss/jms/LDP_ACT_UPDATE_QM"/>
<jee:jndi-lookup id="LDP_ACT_AWARD_GIVEN_QM" jndi-name="jboss/jms/LDP_ACT_AWARD_GIVEN_QM"/>
<jee:jndi-lookup id="LDP_ACT_AWARD_WIN_QM" jndi-name="jboss/jms/LDP_ACT_AWARD_WIN_QM"/>
```

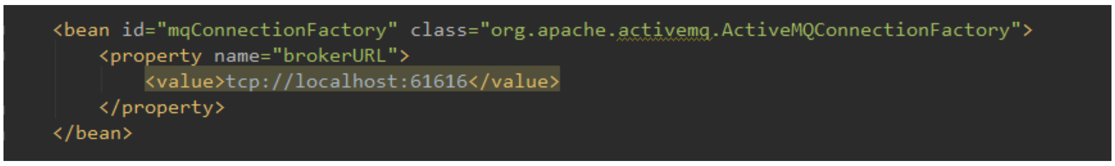


```xml
<jee:jndi-lookup id="connFactory" jndi-name="jboss/jms/OTHER_QM"/>
```

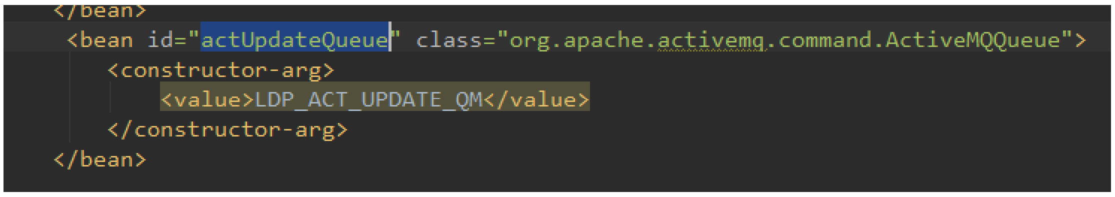


```xml
<!-- 更新推送 topic send-->
<bean id="actUpdateQueueTemplate" class="org.springframework.jms.core.JmsTemplate">
    <constructor-arg ref="connFactory" />
    <property name="defaultDestination" ref="LDP_ACT_UPDATE_QM" />
</bean>
```

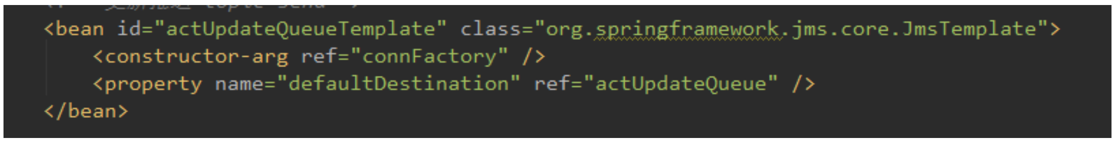
同样使用不同的资源文件



```xml
<bean id="mqConnectionFactory" class="org.apache.activemq.ActiveMQConnectionFactory">
    <property name="brokerURL">
        <value>tcp://localhost:61616</value>
    </property>
</bean>
```



```xml
</bean>
 <bean id="actUpdateQueue" class="org.apache.activemq.command.ActiveMQQueue">
    <constructor-arg>
        <value>LDP_ACT_UPDATE_QM</value>
    </constructor-arg>
</bean>
```



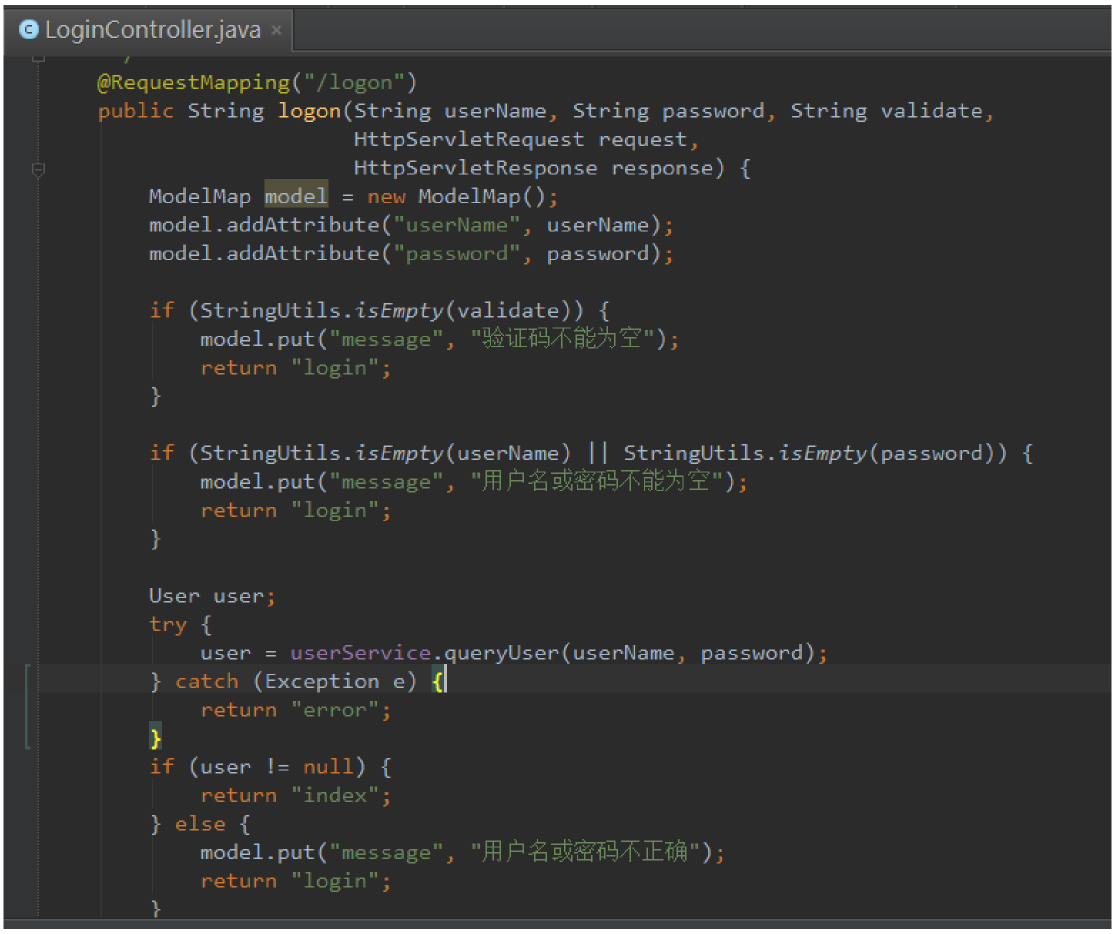
```xml
<bean id="actUpdateQueueTemplate" class="org.springframework.jms.core.JmsTemplate">
    <constructor-arg ref="connFactory" />
    <property name="defaultDestination" ref="actUpdateQueue" />
</bean>
```

对 jndi 资源进行包装

5.4 典型场景的测试方法

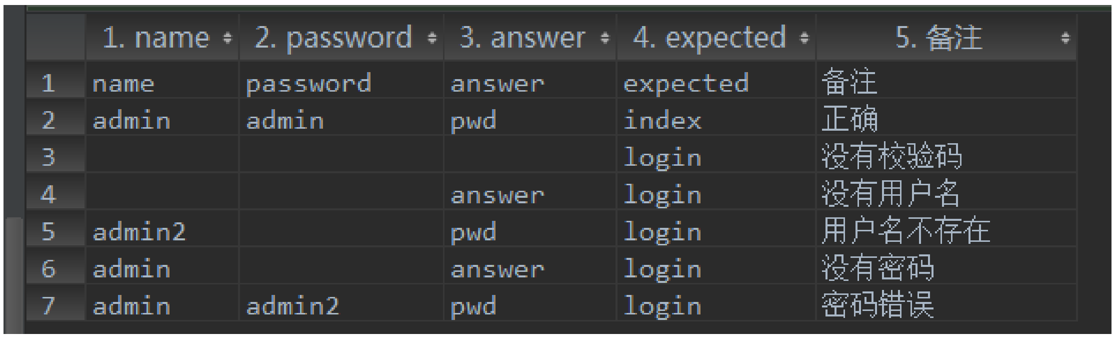
5.4.1 测试 controller

异常情况测试

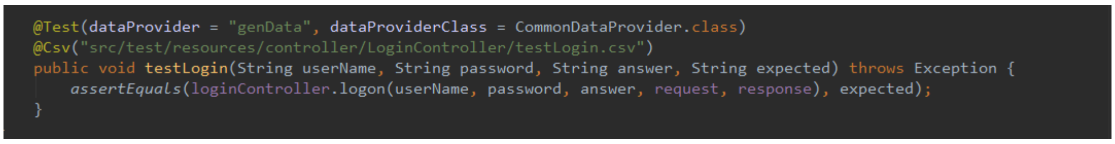


LoginController.java

```java
    @RequestMapping("/logon")
    public String logon(String userName, String password, String validate,
                        HttpServletRequest request,
                        HttpServletResponse response) {
        ModelMap model = new ModelMap();
        model.addAttribute("userName", userName);
        model.addAttribute("password", password);

        if (StringUtils.isEmpty(validate)) {
            model.put("message", "验证码不能为空");
            return "login";
        }

        if (StringUtils.isEmpty(userName) || StringUtils.isEmpty(password)) {
            model.put("message", "用户名或密码不能为空");
            return "login";
        }

        User user;
        try {
            user = userService.queryUser(userName, password);
        } catch (Exception e) {
            return "error";
        }
        if (user != null) {
            return "index";
        } else {
            model.put("message", "用户名或密码不正确");
            return "login";
        }
```

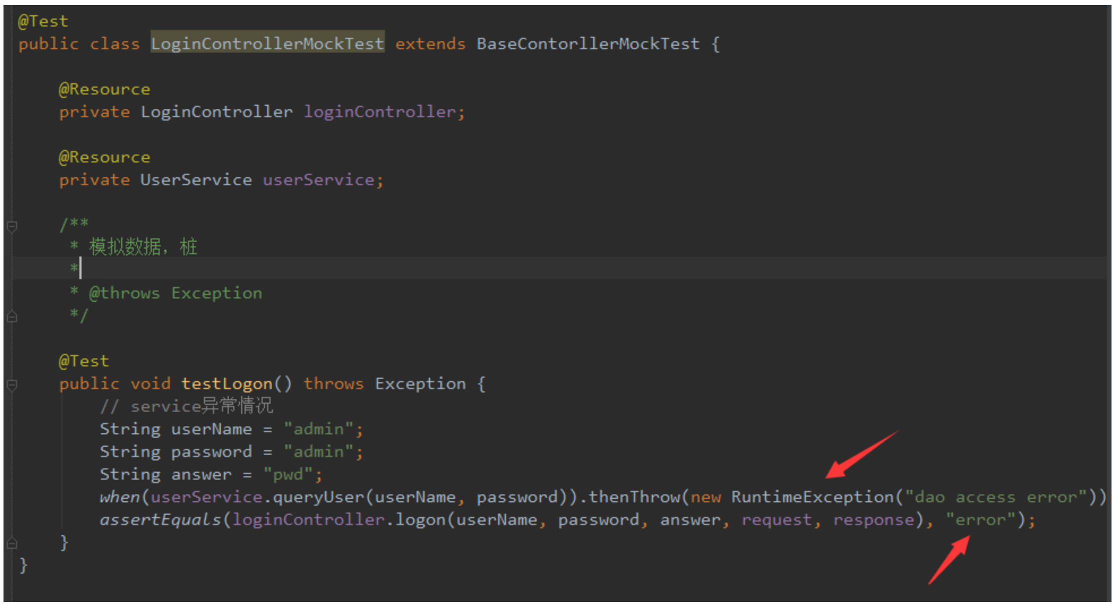


	1. name	2. password	3. answer	4. expected	5. 备注
1	name	password	answer	expected	备注
2	admin	admin	pwd	index	正确
3				login	没有校验码
4			answer	login	没有用户名
5	admin2		pwd	login	用户名不存在
6	admin		answer	login	没有密码
7	admin	admin2	pwd	login	密码错误



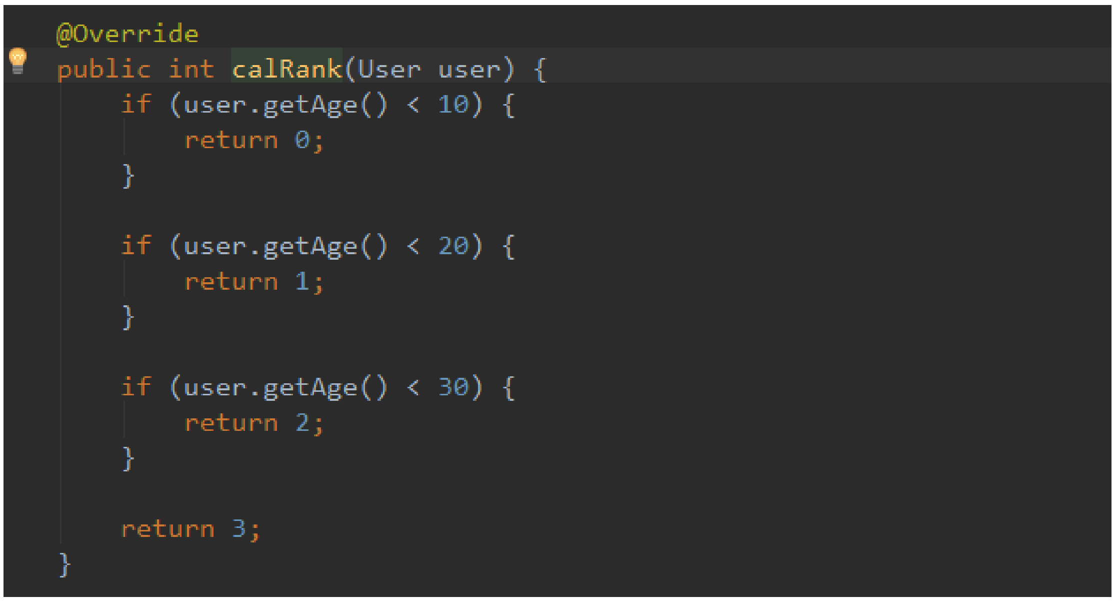
```java
@Test(dataProvider = "genData", dataProviderClass = CommonDataProvider.class)
@Csv("src/test/resources/controller/LoginController/testLogin.csv")
public void testLogin(String userName, String password, String answer, String expected) throws Exception {
    assertEquals(loginController.logon(userName, password, answer, request, response), expected);
}
```

异常情况测试

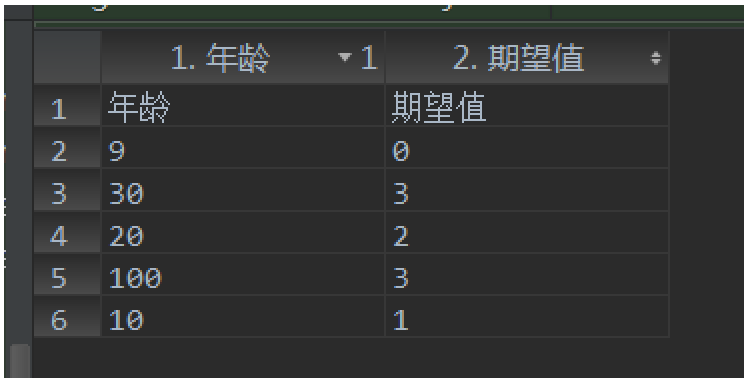


```java
@Test
public class LoginControllerMockTest extends BaseContorllerMockTest {

    @Resource
    private LoginController loginController;

    @Resource
    private UserService userService;

    /**
     * 模拟数据，桩
     *
     * @throws Exception
     */

    @Test
    public void testLogon() throws Exception {
        // service异常情况
        String userName = "admin";
        String password = "admin";
        String answer = "pwd";
        when(userService.queryUser(userName, password)).thenThrow(new RuntimeException("dao access error"))
        assertEquals(loginController.logon(userName, password, answer, request, response), "error");
    }
}
```

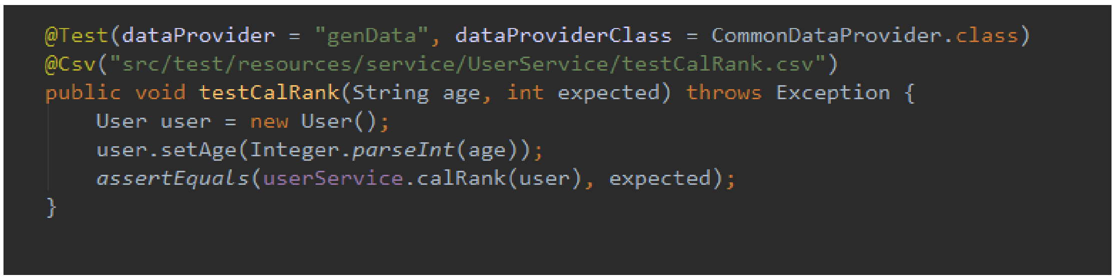
5.4.2 测试 service



```java
@Override
public int calRank(User user) {
    if (user.getAge() < 10) {
        return 0;
    }

    if (user.getAge() < 20) {
        return 1;
    }

    if (user.getAge() < 30) {
        return 2;
    }

    return 3;
}
```

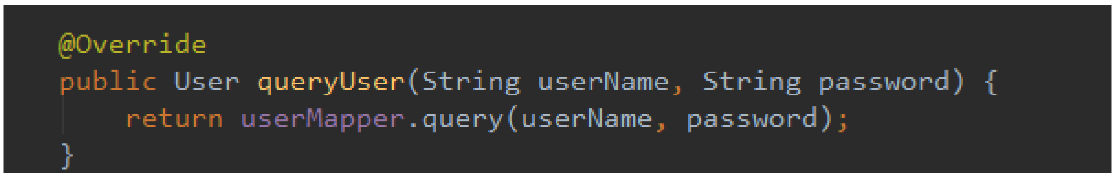


	1. 年龄 ▾1	2. 期望值
1	年龄	期望值
2	9	0
3	30	3
4	20	2
5	100	3
6	10	1

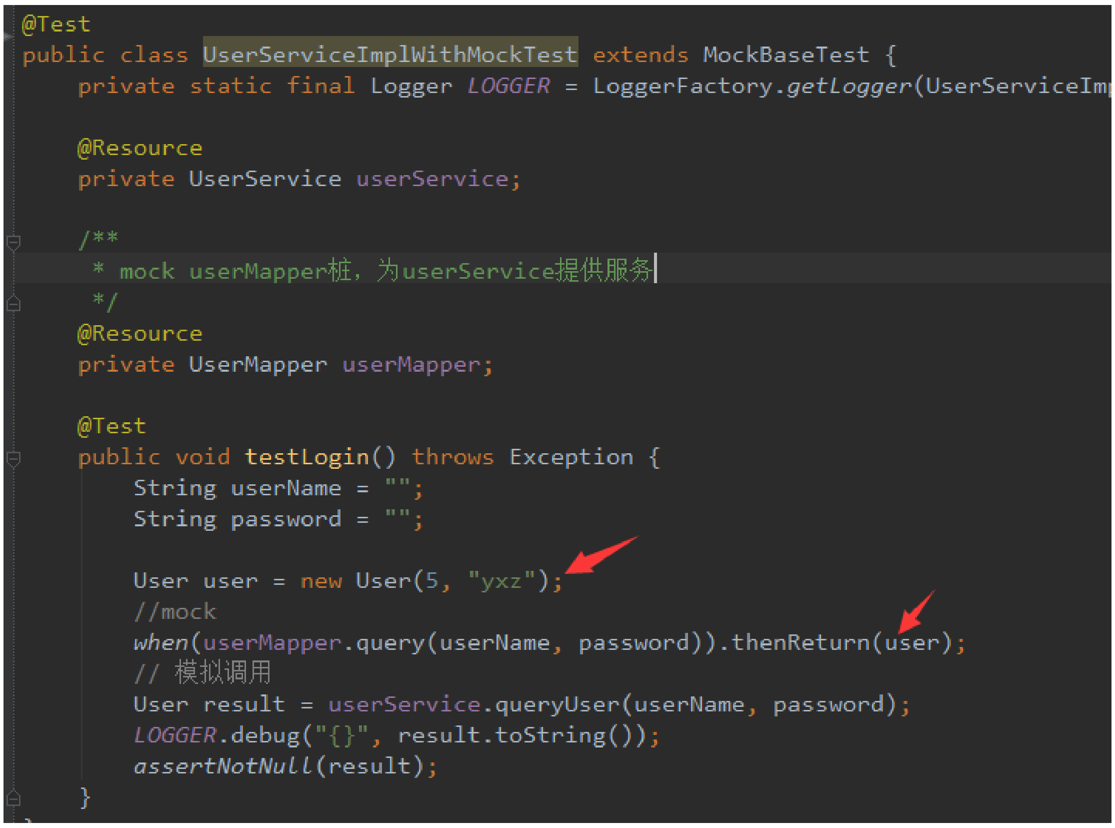


```java
@Test(dataProvider = "genData", dataProviderClass = CommonDataProvider.class)
@Csv("src/test/resources/service/UserService/testCalRank.csv")
public void testCalRank(String age, int expected) throws Exception {
    User user = new User();
    user.setAge(Integer.parseInt(age));
    assertEquals(userService.calRank(user), expected);
}
```

中间层 Mock 测试



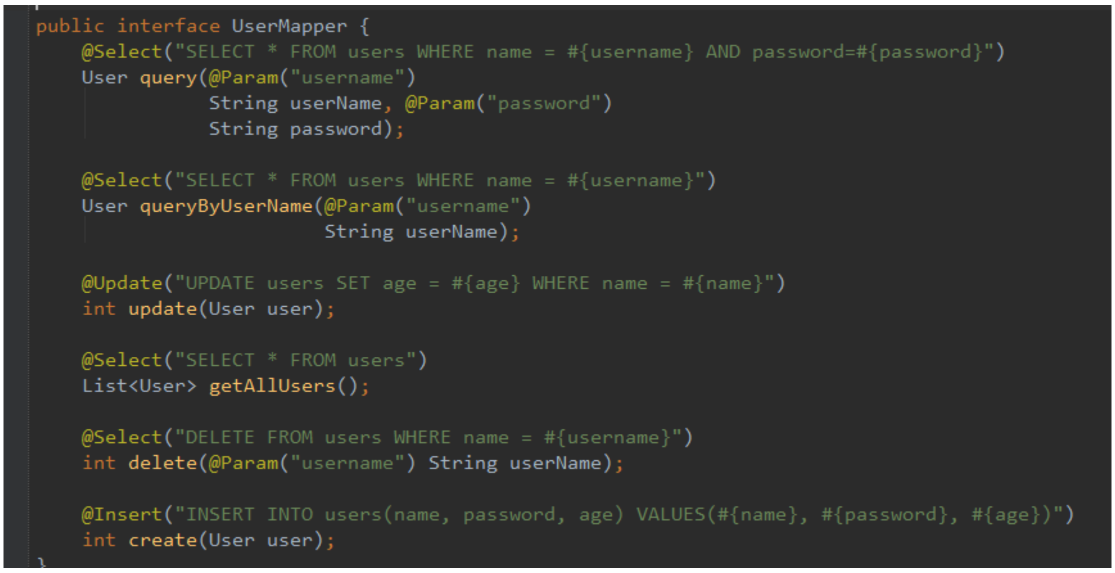
```java
@Override
public User queryUser(String userName, String password) {
    return userMapper.query(userName, password);
}
```



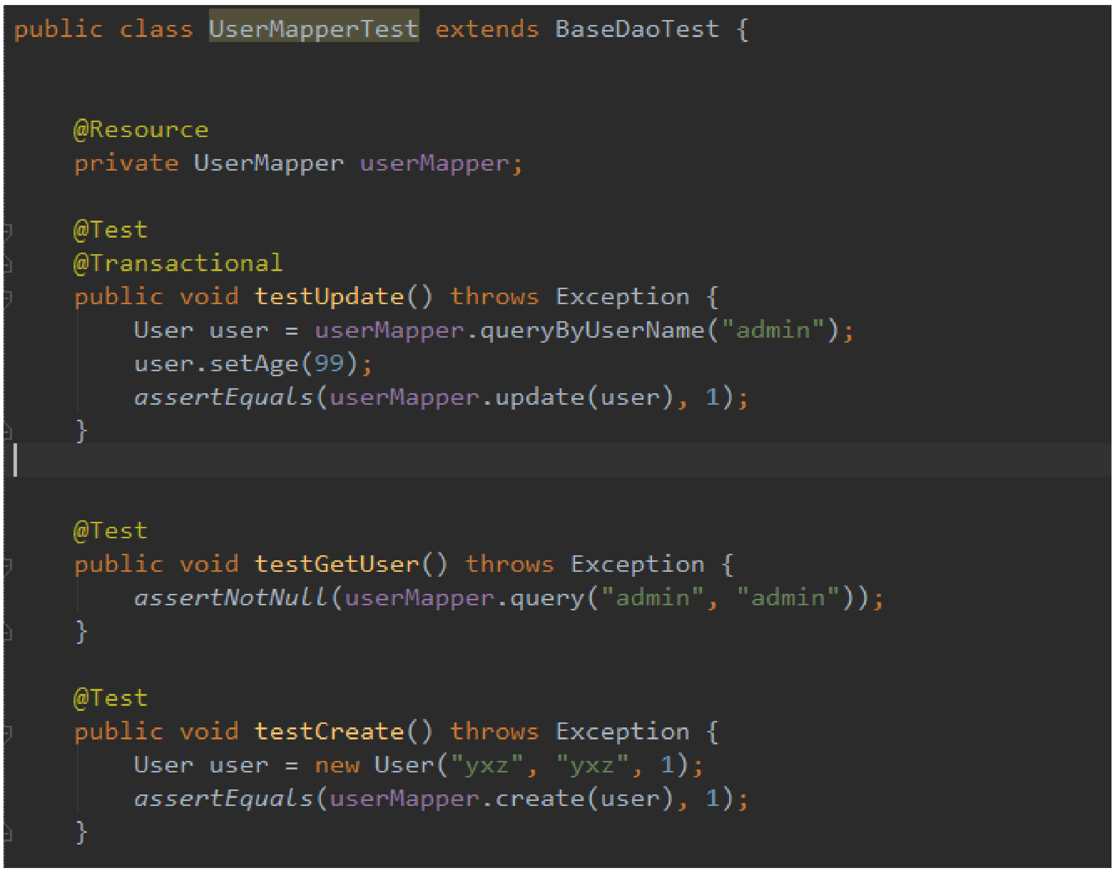
```java
@Test
public class UserServiceImplWithMockTest extends MockBaseTest {
    private static final Logger LOGGER = LoggerFactory.getLogger(UserServiceIm

    @Resource
    private UserService userService;

    /**
     * mock userMapper桩，为userService提供服务
     */
    @Resource
    private UserMapper userMapper;

    @Test
    public void testLogin() throws Exception {
        String userName = "";
        String password = "";

        User user = new User(5, "yxz");
        //mock
        when(userMapper.query(userName, password)).thenReturn(user);
        // 模拟调用
        User result = userService.queryUser(userName, password);
        LOGGER.debug("{}", result.toString());
        assertNotNull(result);
    }
```

5.4.3 测试 dao

5.4.3.1 直接操作数据库



```
public interface UserMapper {
    @Select("SELECT * FROM users WHERE name = #{username} AND password=#{password}")
    User query(@Param("username")
               String userName, @Param("password")
               String password);

    @Select("SELECT * FROM users WHERE name = #{username}")
    User queryByUserName(@Param("username")
                         String userName);

    @Update("UPDATE users SET age = #{age} WHERE name = #{name}")
    int update(User user);

    @Select("SELECT * FROM users")
    List<User> getAllUsers();

    @Select("DELETE FROM users WHERE name = #{username}")
    int delete(@Param("username") String userName);

    @Insert("INSERT INTO users(name, password, age) VALUES(#{name}, #{password}, #{age})")
    int create(User user);
}
```

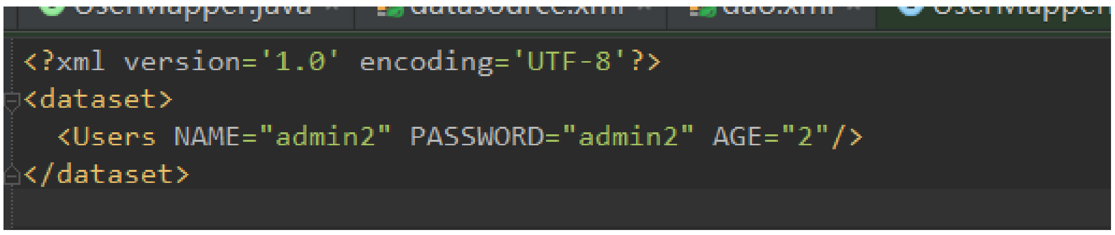


```
public class UserMapperTest extends BaseDaoTest {


    @Resource
    private UserMapper userMapper;

    @Test
    @Transactional
    public void testUpdate() throws Exception {
        User user = userMapper.queryByUserName("admin");
        user.setAge(99);
        assertEquals(userMapper.update(user), 1);
    }


    @Test
    public void testGetUser() throws Exception {
        assertNotNull(userMapper.query("admin", "admin"));
    }

    @Test
    public void testCreate() throws Exception {
        User user = new User("yxz", "yxz", 1);
        assertEquals(userMapper.create(user), 1);
    }
```

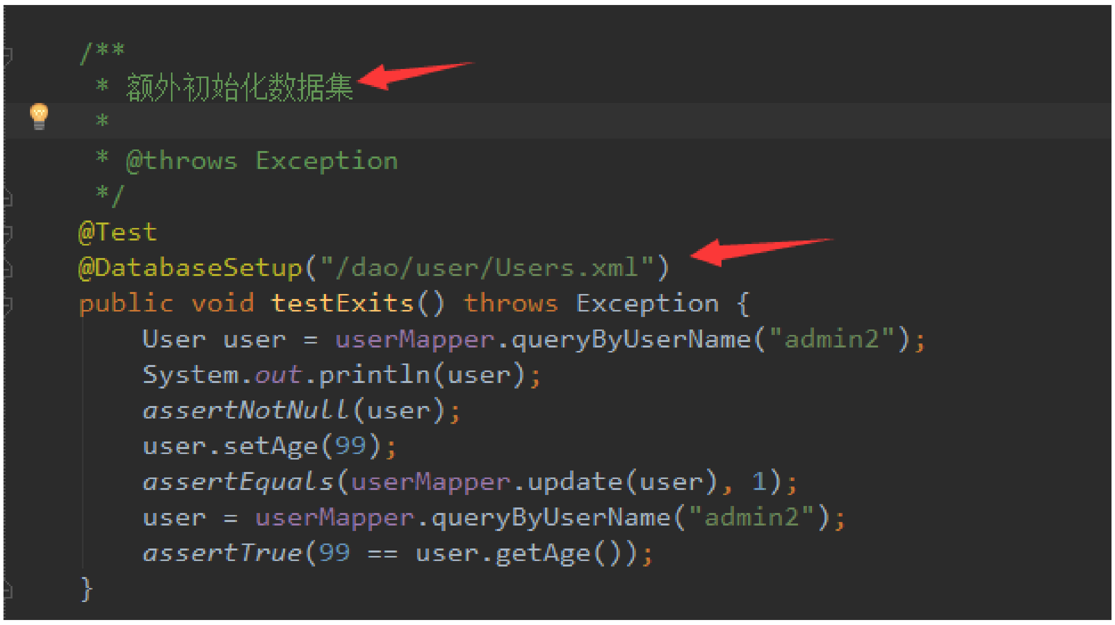
5.4.3.2 使用 dbunit+springtestdbunit 进行操作

1. 自动插入数据集

Users.xml 内容



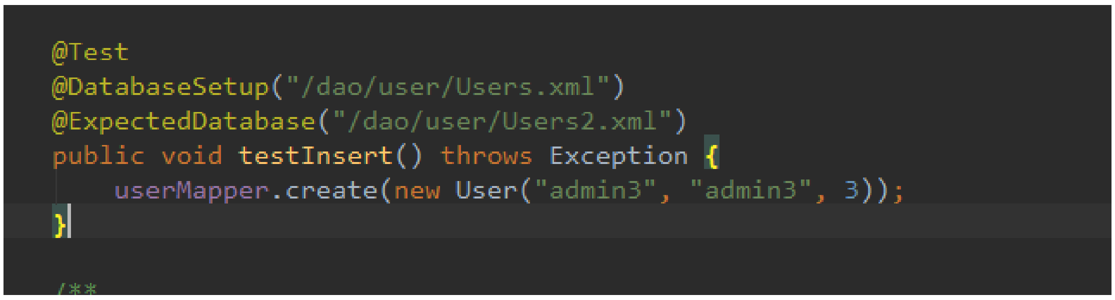
```
<?xml version='1.0' encoding='UTF-8'?>
<dataset>
  <Users NAME="admin2" PASSWORD="admin2" AGE="2"/>
</dataset>
```



```
    /**
     * 额外初始化数据集
     *
     * @throws Exception
     */
    @Test
    @DatabaseSetup("/dao/user/Users.xml")
    public void testExits() throws Exception {
        User user = userMapper.queryByUserName("admin2");
        System.out.println(user);
        assertNotNull(user);
        user.setAge(99);
        assertEquals(userMapper.update(user), 1);
        user = userMapper.queryByUserName("admin2");
        assertTrue(99 == user.getAge());
    }
```

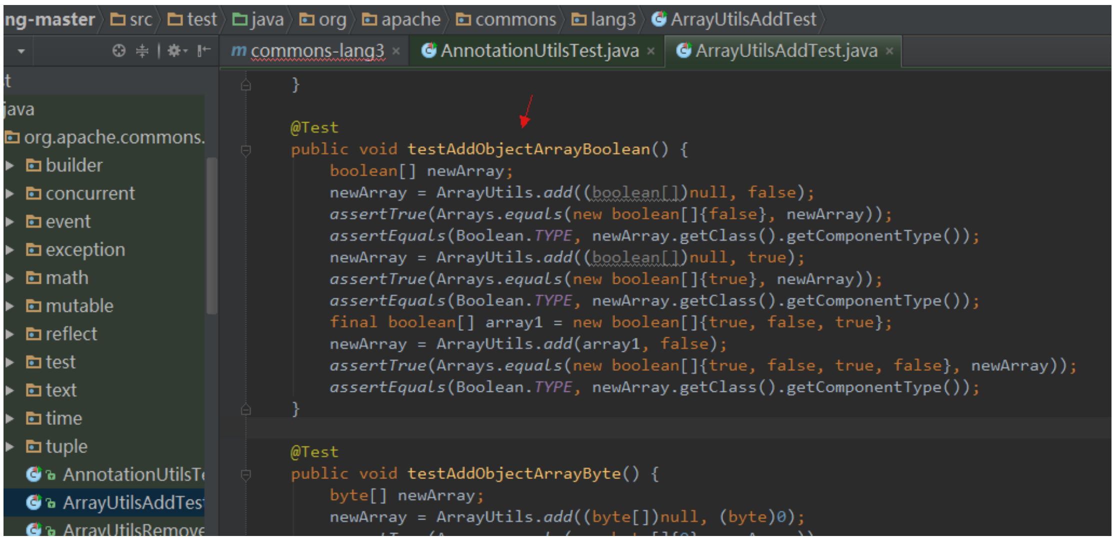
2. 数据集自动比较

初始化的数据集合，执行操作之后，期望的数据集合。

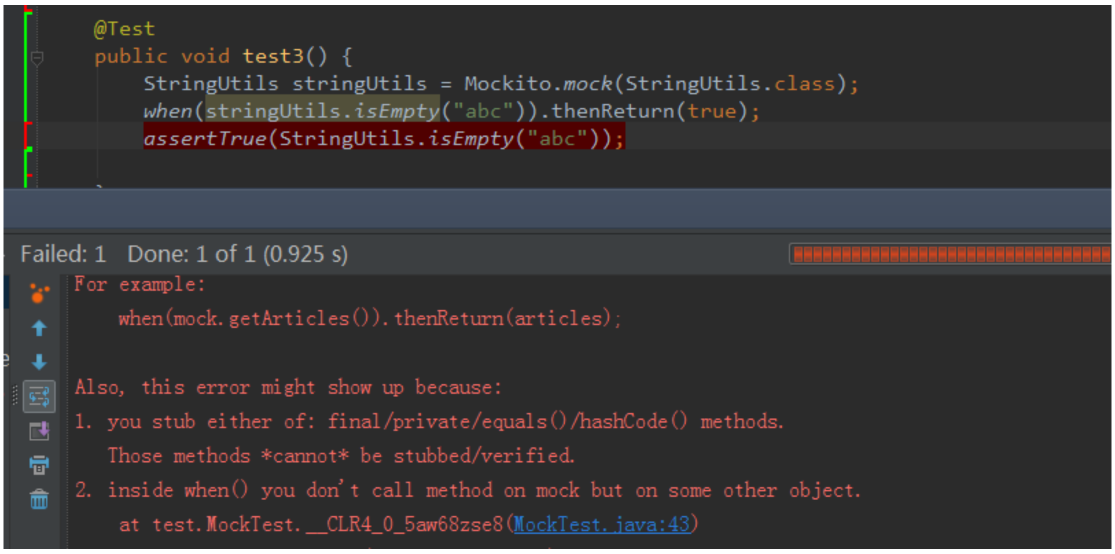


```
@Test
@DatabaseSetup("/dao/user/Users.xml")
@ExpectedDatabase("/dao/user/Users2.xml")
public void testInsert() throws Exception {
    userMapper.create(new User("admin3", "admin3", 3));
}

/**
```

5.4.4 测试工具类

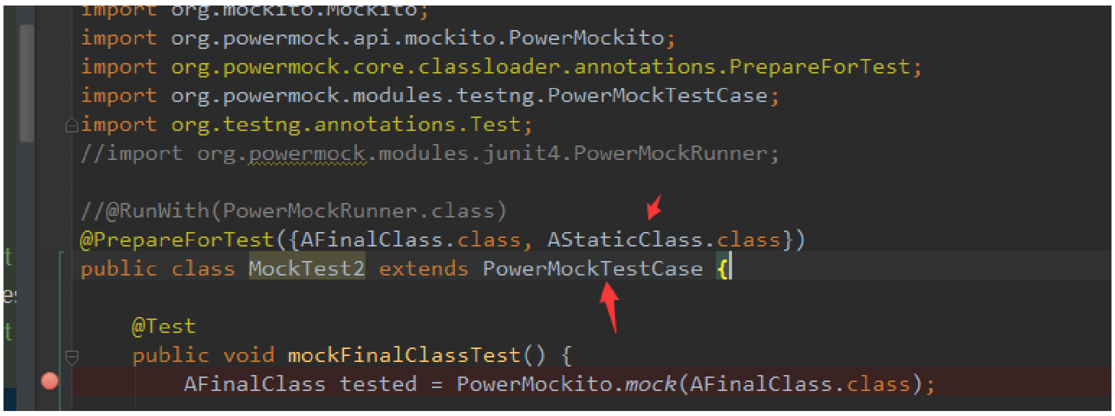


```java
    }

    @Test
    public void testAddObjectArrayBoolean() {
        boolean[] newArray;
        newArray = ArrayUtils.add((boolean[])null, false);
        assertTrue(Arrays.equals(new boolean[]{false}, newArray));
        assertEquals(Boolean.TYPE, newArray.getClass().getComponentType());
        newArray = ArrayUtils.add((boolean[])null, true);
        assertTrue(Arrays.equals(new boolean[]{true}, newArray));
        assertEquals(Boolean.TYPE, newArray.getClass().getComponentType());
        final boolean[] array1 = new boolean[]{true, false, true};
        newArray = ArrayUtils.add(array1, false);
        assertTrue(Arrays.equals(new boolean[]{true, false, true, false}, newArray));
        assertEquals(Boolean.TYPE, newArray.getClass().getComponentType());
    }

    @Test
    public void testAddObjectArrayByte() {
        byte[] newArray;
        newArray = ArrayUtils.add((byte[])null, (byte)0);
```

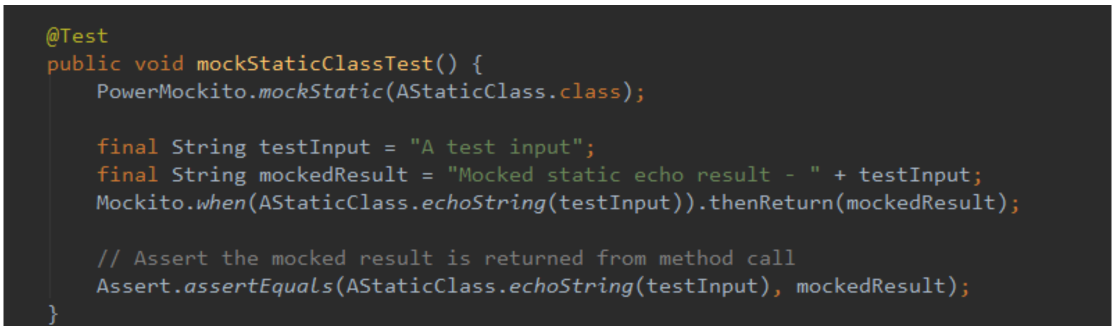


```java
    @Test
    public void test3() {
        StringUtils stringUtils = Mockito.mock(StringUtils.class);
        when(stringUtils.isEmpty("abc")).thenReturn(true);
        assertTrue(StringUtils.isEmpty("abc"));
```

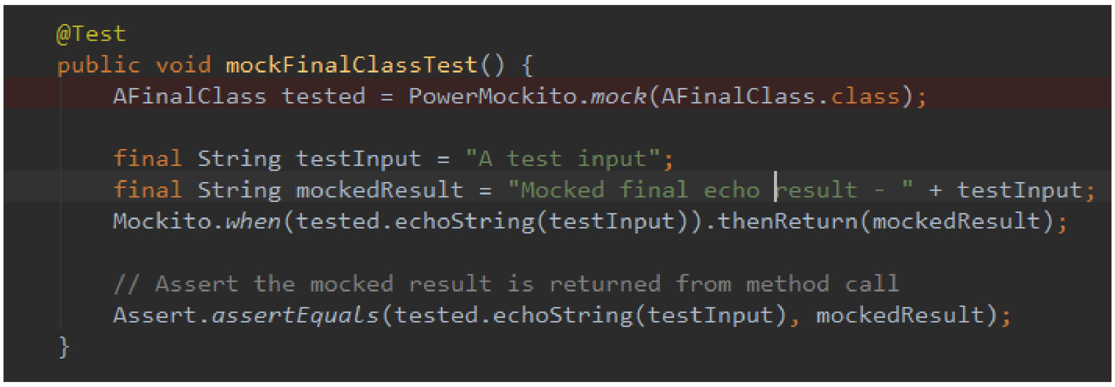
```
Failed: 1   Done: 1 of 1 (0.925 s)

For example:
    when(mock.getArticles()).thenReturn(articles);

Also, this error might show up because:
1. you stub either of: final/private/equals()/hashCode() methods.
   Those methods *cannot* be stubbed/verified.
2. inside when() you don't call method on mock but on some other object.
    at test.MockTest.__CLR4_0_5aw68zse8(MockTest.java:43)
```



```java
import org.mockito.Mockito;
import org.powermock.api.mockito.PowerMockito;
import org.powermock.core.classloader.annotations.PrepareForTest;
import org.powermock.modules.testng.PowerMockTestCase;
import org.testng.annotations.Test;
//import org.powermock.modules.junit4.PowerMockRunner;

//@RunWith(PowerMockRunner.class)
@PrepareForTest({AFinalClass.class, AStaticClass.class})
public class MockTest2 extends PowerMockTestCase {

    @Test
    public void mockFinalClassTest() {
        AFinalClass tested = PowerMockito.mock(AFinalClass.class);
```



```java
@Test
public void mockStaticClassTest() {
    PowerMockito.mockStatic(AStaticClass.class);

    final String testInput = "A test input";
    final String mockedResult = "Mocked static echo result - " + testInput;
    Mockito.when(AStaticClass.echoString(testInput)).thenReturn(mockedResult);

    // Assert the mocked result is returned from method call
    Assert.assertEquals(AStaticClass.echoString(testInput), mockedResult);
}
```



```java
@Test
public void mockFinalClassTest() {
    AFinalClass tested = PowerMockito.mock(AFinalClass.class);

    final String testInput = "A test input";
    final String mockedResult = "Mocked final echo result - " + testInput;
    Mockito.when(tested.echoString(testInput)).thenReturn(mockedResult);

    // Assert the mocked result is returned from method call
    Assert.assertEquals(tested.echoString(testInput), mockedResult);
}
```

慎用 powermock，当需要使用 powermock 来 mock 一个 private 函数，或者全局变量，或者静态函数
时候，要想着有更好的模式来支持 mockito 的功能，而不是来模拟。

示例代码路径

https://github.com/superproxy/sample-test

FINAL 测试

http://www.codeproject.com/Articles/806508/Using-PowerMockito-to-Mock-Final-and-Static-Method

5.5 性能测试

参考资料

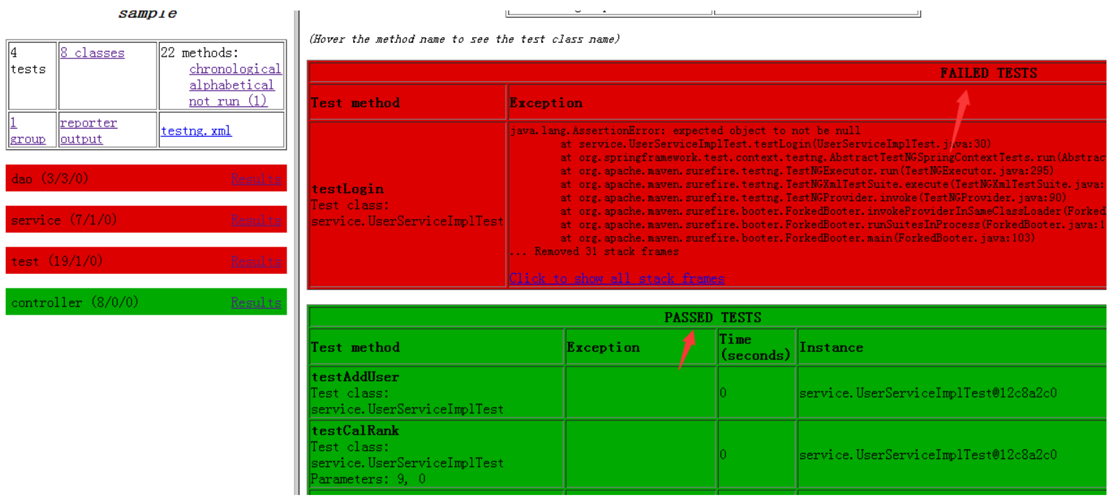
http://stamen.iteye.com/blog/1485837

http://www.ibm.com/developerworks/cn/java/j-cq11296.html
http://wenku.baidu.com/link?url=z0ISNJmBm_AQZmNPnSberLk7wu14vLYD4w4JbRa9gNRhk6RPwxOFttI6Ti5zGRo6kOblix2wy_vQFKXarP_PjBgU84RK64LoDGOjqXypr17

6 测试结果展示

6.1 本地测试

6.1.1 单元测试执行情况


sample

4 tests	8 classes	22 methods: chronological alphabetical not run (1)
1 group	reporter output	testng.xml

dao (3/3/0)	Results
service (7/1/0)	Results
test (19/1/0)	Results
controller (8/0/0)	Results

(Hover the method name to see the test class name)

FAILED TESTS

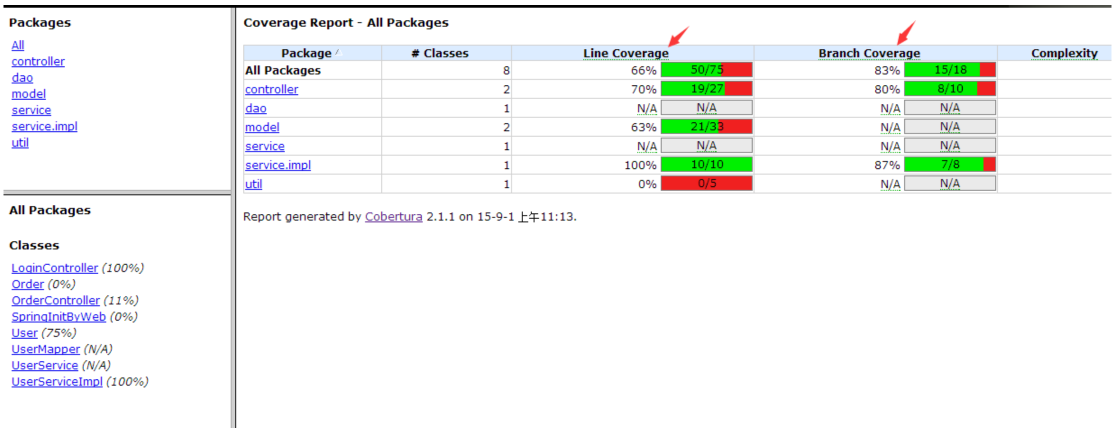
Test method	Exception
testLogin Test class: service.UserServiceImplTest	see below

```
java.lang.AssertionError: expected object to not be null
	at service.UserServiceImplTest.testLogin(UserServiceImplTest.java:30)
	at org.springframework.test.context.testng.AbstractTestNGSpringContextTests.run(Abstrac
	at org.apache.maven.surefire.testng.TestNGExecutor.run(TestNGExecutor.java:295)
	at org.apache.maven.surefire.testng.TestNGXmlTestSuite.execute(TestNGXmlTestSuite.java:
	at org.apache.maven.surefire.testng.TestNGProvider.invoke(TestNGProvider.java:90)
	at org.apache.maven.surefire.booter.ForkedBooter.invokeProviderInSameClassLoader(Forked
	at org.apache.maven.surefire.booter.ForkedBooter.runSuitesInProcess(ForkedBooter.java:1
	at org.apache.maven.surefire.booter.ForkedBooter.main(ForkedBooter.java:103)
... Removed 31 stack frames

Click to show all stack frames
```

PASSED TESTS

Test method	Exception	Time (seconds)	Instance
testAddUser Test class: service.UserServiceImplTest		0	service.UserServiceImplTest@12c8a2c0
testCalRank Test class: service.UserServiceImplTest Parameters: 9, 0		0	service.UserServiceImplTest@12c8a2c0

6.1.2 整体包测试覆盖率


Packages

All
controller
dao
model
service
service.impl
util

All Packages

Classes

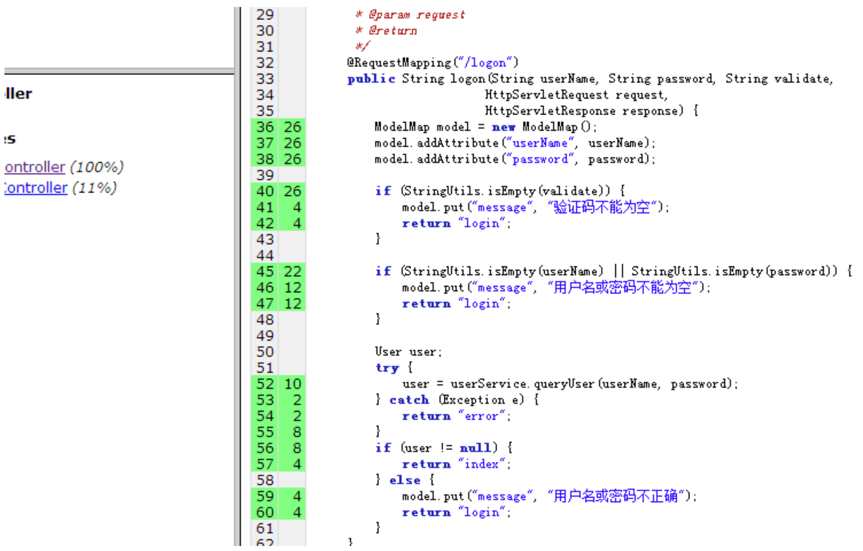
LoginController *(100%)*
Order *(0%)*
OrderController *(11%)*
SpringInitByWeb *(0%)*
User *(75%)*
UserMapper *(N/A)*
UserService *(N/A)*
UserServiceImpl *(100%)*

Coverage Report - All Packages

Package	# Classes	Line Coverage		Branch Coverage		Complexity
All Packages	8	66%	50/75	83%	15/18	
controller	2	70%	19/27	80%	8/10	
dao	1	N/A	N/A	N/A	N/A	
model	2	63%	21/33	N/A	N/A	
service	1	N/A	N/A	N/A	N/A	
service.impl	1	100%	10/10	87%	7/8	
util	1	0%	0/5	N/A	N/A	

Report generated by Cobertura 2.1.1 on 15-9-1 上午11:13.

6.1.3 行和分支包测试覆盖率

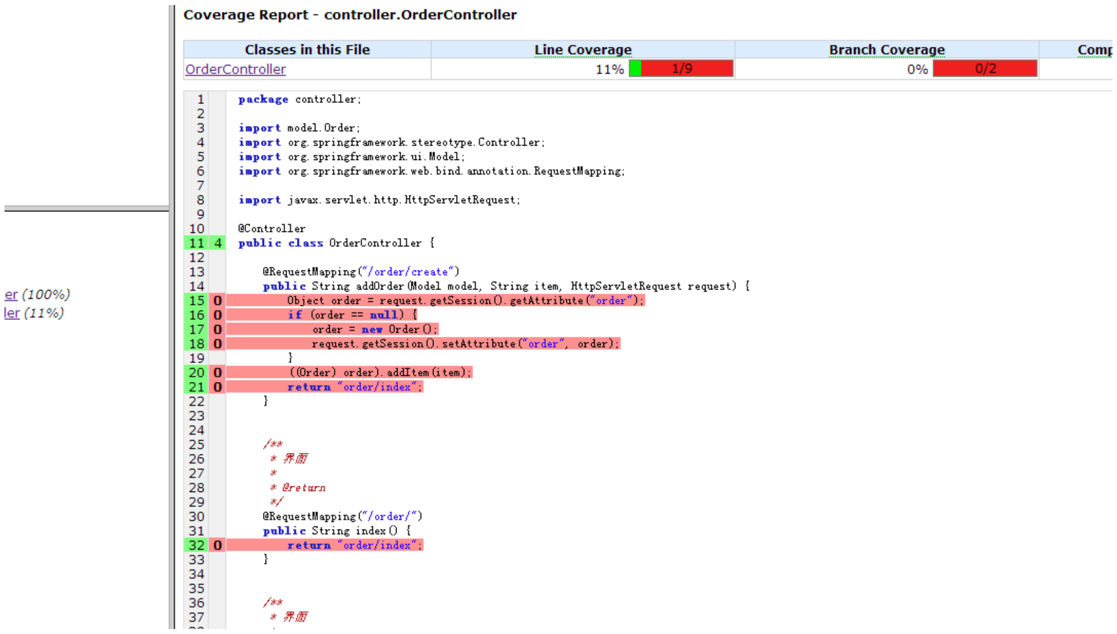


ller

s

ontroller (100%)
ontroller (11%)

```
         * @param request
         * @return
         */
        @RequestMapping("/logon")
        public String logon(String userName, String password, String validate,
                            HttpServletRequest request,
                            HttpServletResponse response) {
26          ModelMap model = new ModelMap();
26          model.addAttribute("userName", userName);
26          model.addAttribute("password", password);

26          if (StringUtils.isEmpty(validate)) {
 4              model.put("message", "验证码不能为空");
 4              return "login";
            }

22          if (StringUtils.isEmpty(userName) || StringUtils.isEmpty(password)) {
12              model.put("message", "用户名或密码不能为空");
12              return "login";
            }

            User user;
            try {
10              user = userService.queryUser(userName, password);
 2          } catch (Exception e) {
 2              return "error";
 8          }
 8          if (user != null) {
 4              return "index";
            } else {
 4              model.put("message", "用户名或密码不正确");
 4              return "login";
            }
        }
```



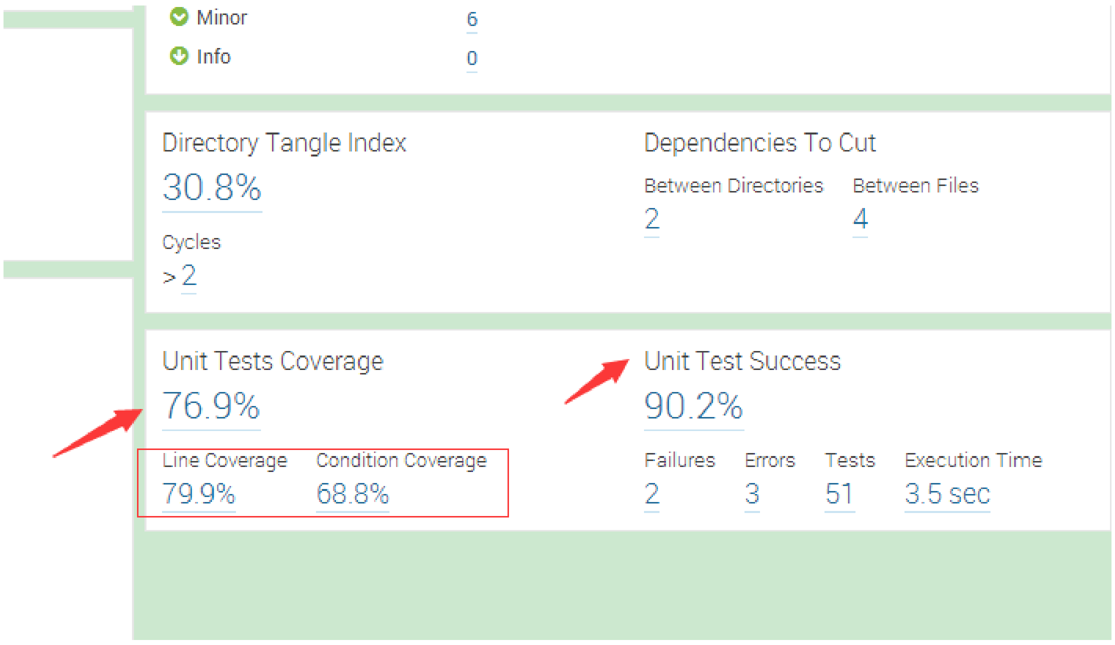
Coverage Report - controller.OrderController

Classes in this File	Line Coverage	Branch Coverage	Comp
OrderController	11% 1/9	0% 0/2	

er (100%)
ler (11%)

```
   package controller;

   import model.Order;
   import org.springframework.stereotype.Controller;
   import org.springframework.ui.Model;
   import org.springframework.web.bind.annotation.RequestMapping;

   import javax.servlet.http.HttpServletRequest;

   @Controller
4  public class OrderController {

       @RequestMapping("/order/create")
       public String addOrder(Model model, String item, HttpServletRequest request) {
0          Object order = request.getSession().getAttribute("order");
0          if (order == null) {
0              order = new Order();
0              request.getSession().setAttribute("order", order);
           }
0          ((Order) order).addItem(item);
0          return "order/index";
       }


       /**
        * 界面
        *
        * @return
        */
       @RequestMapping("/order/")
       public String index() {
0          return "order/index";
       }


       /**
        * 界面
```

6.2 Sonar 展示



Minor 6
Info 0

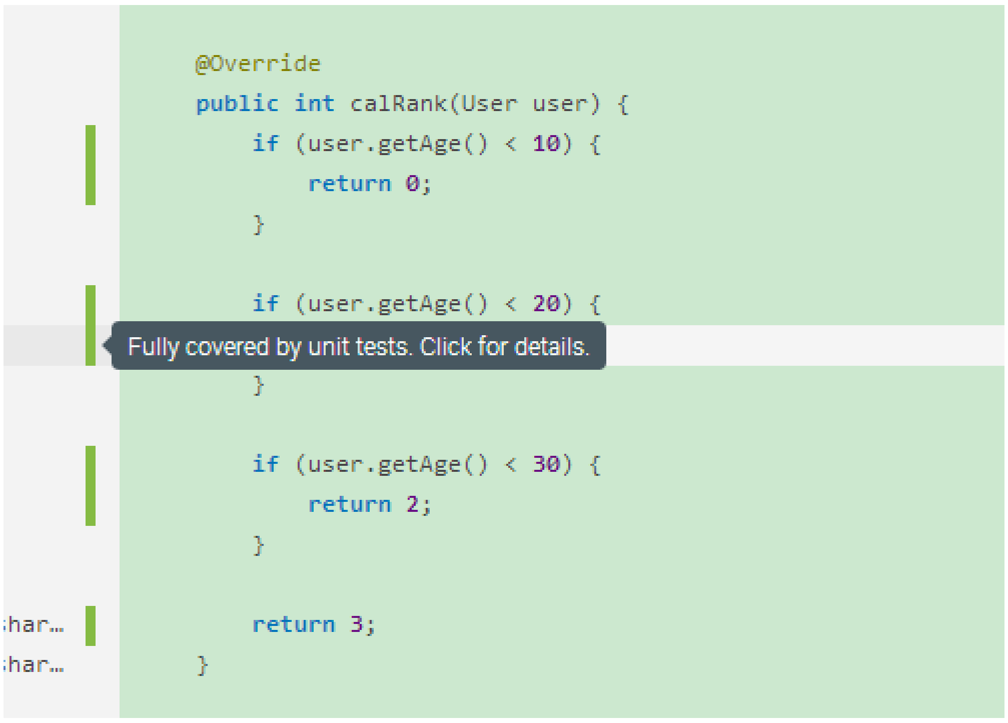
Directory Tangle Index
30.8%
Cycles
> 2

Dependencies To Cut
Between Directories 2
Between Files 4

Unit Tests Coverage
76.9%
Line Coverage 79.9%
Condition Coverage 68.8%

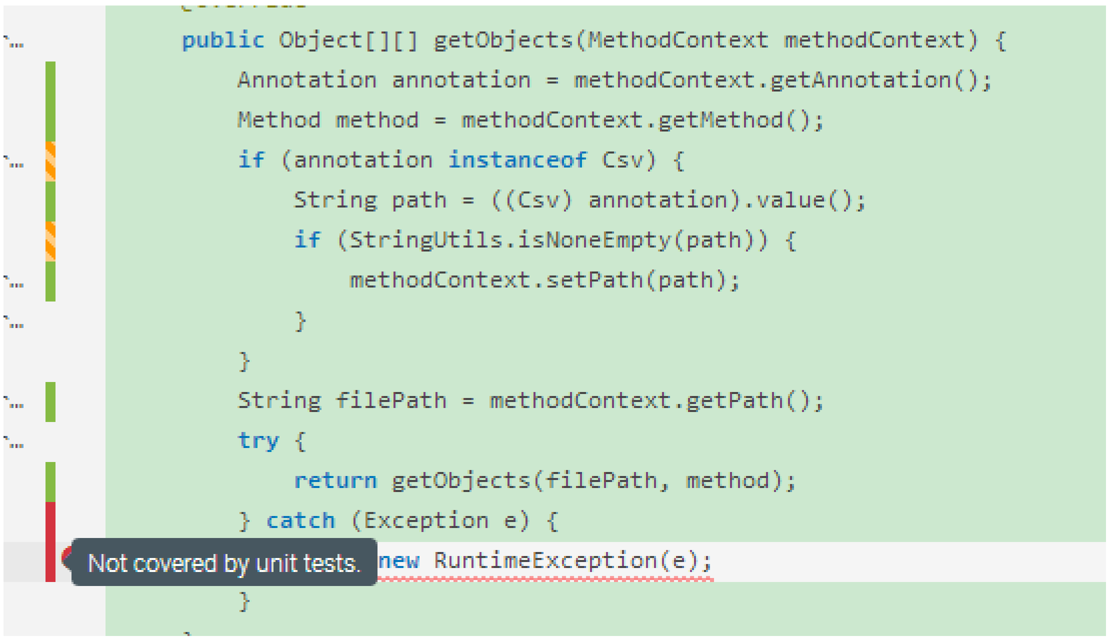
Unit Test Success
90.2%
Failures 2
Errors 3
Tests 51
Execution Time 3.5 sec

6.3 完整测试



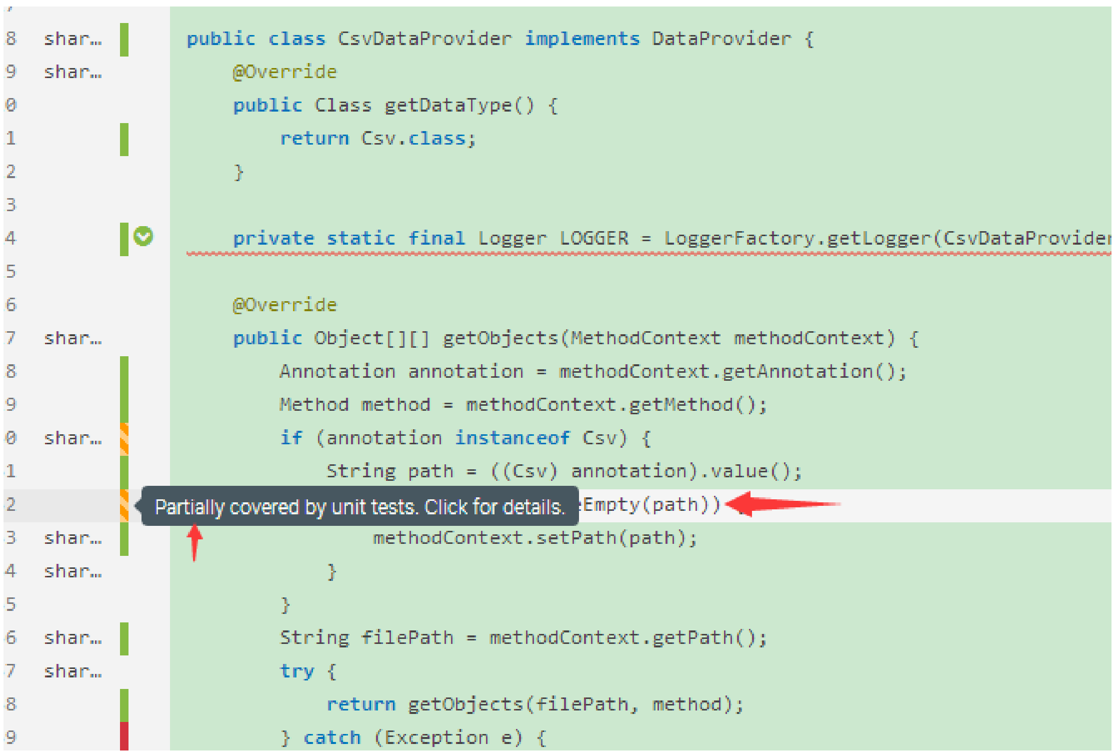
```
@Override
public int calRank(User user) {
    if (user.getAge() < 10) {
        return 0;
    }

    if (user.getAge() < 20) {
Fully covered by unit tests. Click for details.
    }

    if (user.getAge() < 30) {
        return 2;
    }

    return 3;
}
```

6.4 没有测试



```
    public Object[][] getObjects(MethodContext methodContext) {
        Annotation annotation = methodContext.getAnnotation();
        Method method = methodContext.getMethod();
        if (annotation instanceof Csv) {
            String path = ((Csv) annotation).value();
            if (StringUtils.isNoneEmpty(path)) {
                methodContext.setPath(path);
            }
        }
        String filePath = methodContext.getPath();
        try {
            return getObjects(filePath, method);
        } catch (Exception e) {
 Not covered by unit tests. new RuntimeException(e);
        }
```

6.5 分支部分测试



```
public class CsvDataProvider implements DataProvider {
    @Override
    public Class getDataType() {
        return Csv.class;
    }

    private static final Logger LOGGER = LoggerFactory.getLogger(CsvDataProvide

    @Override
    public Object[][] getObjects(MethodContext methodContext) {
        Annotation annotation = methodContext.getAnnotation();
        Method method = methodContext.getMethod();
        if (annotation instanceof Csv) {
            String path = ((Csv) annotation).value();
 Partially covered by unit tests. Click for details. Empty(path))
                methodContext.setPath(path);
            }
        }
        String filePath = methodContext.getPath();
        try {
            return getObjects(filePath, method);
        } catch (Exception e) {
```

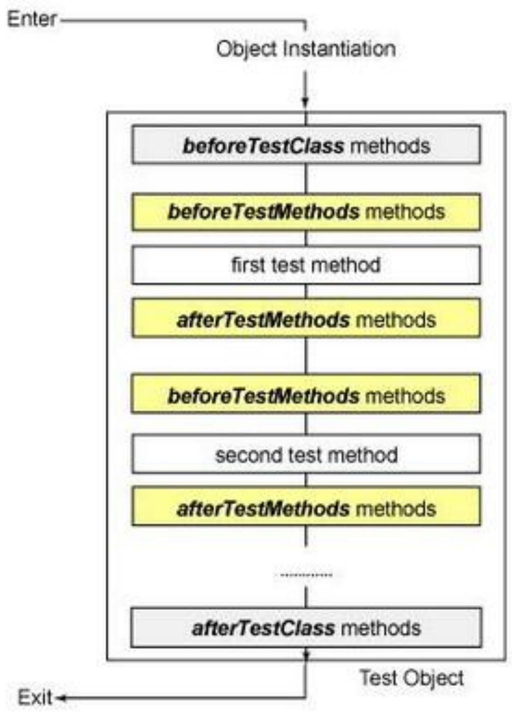
7 测试工具

7.1 TestNG

先看一下Junit的执行顺序

Setup( )　test1( )　tearDown( )　　Setup( )　test2( )　　tearDown( )............

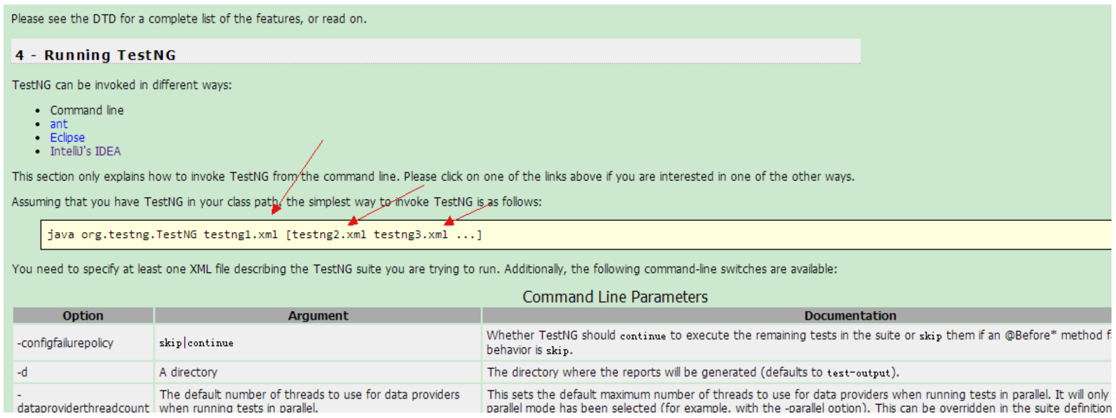
下面是TestNG的



TSTNG 优点：　其实发展到现在相差不大
从 junit 到 testng
https://developers.opengamma.com/blog/2011/04/04/converting-opengamma-junit-testng
TesgNG 为什么更好
http://www.ibm.com/developerworks/cn/java/j-cq08296/
http://beust.com/weblog2/archives/000369.html

7.1.1 启动 testng

7.1.1.1命令行



Please see the DTD for a complete list of the features, or read on.

4 - Running TestNG

TestNG can be invoked in different ways:

- Command line
- ant
- Eclipse
- IntelliJ's IDEA

This section only explains how to invoke TestNG from the command line. Please click on one of the links above if you are interested in one of the other ways.

Assuming that you have TestNG in your class path, the simplest way to invoke TestNG is as follows:

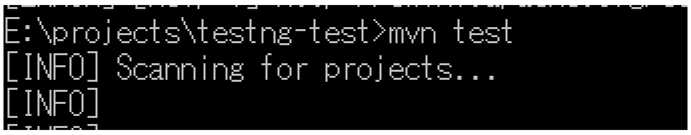
```
java org.testng.TestNG testng1.xml [testng2.xml testng3.xml ...]
```

You need to specify at least one XML file describing the TestNG suite you are trying to run. Additionally, the following command-line switches are available:

Command Line Parameters

Option	Argument	Documentation
-configfailurepolicy	skip|continue	Whether TestNG should continue to execute the remaining tests in the suite or skip them if an @Before* method f behavior is skip.
-d	A directory	The directory where the reports will be generated (defaults to test-output).
- dataproviderthreadcount	The default number of threads to use for data providers when running tests in parallel.	This sets the default maximum number of threads to use for data providers when running tests in parallel. It will only parallel mode has been selected (for example, with the -parallel option). This can be overridden in the suite definition

7.1.1.2Mvn 集成

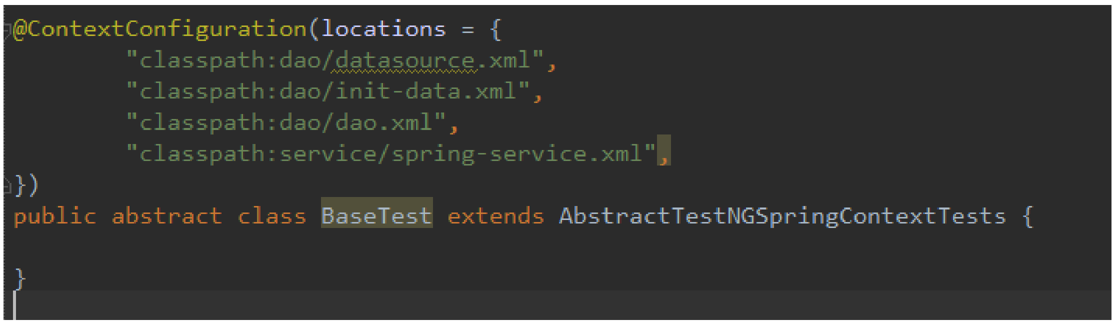


```
E:\projects\testng-test>mvn test
[INFO] Scanning for projects...
[INFO]
```

http://maven.apache.org/surefire/maven-surefire-plugin/examples/testng.html

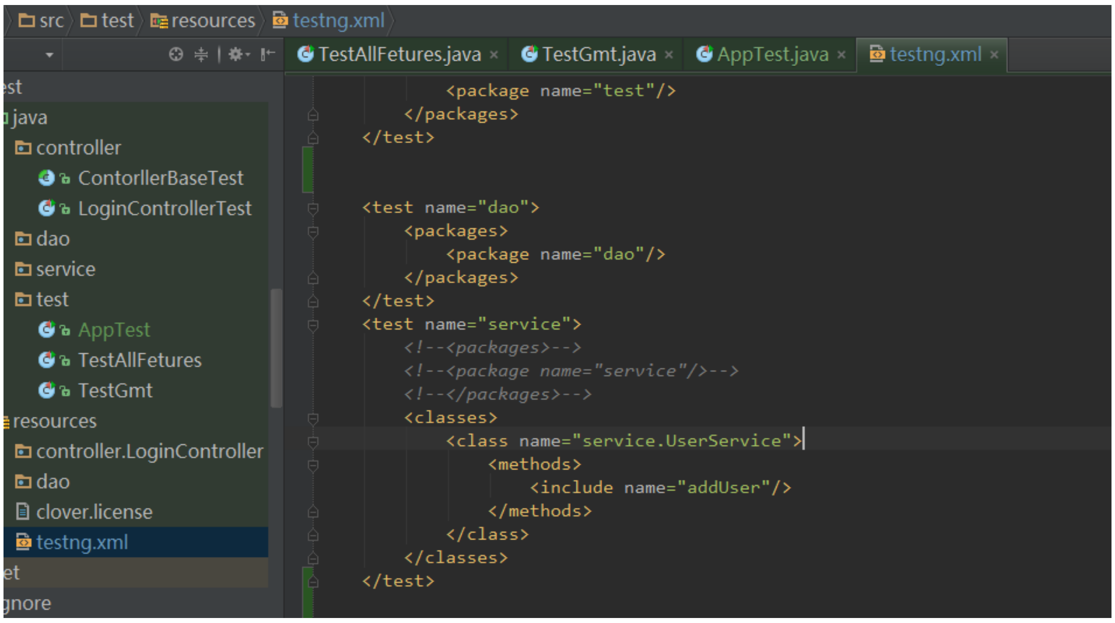
7.1.2 Spring 集成

Spring 2.5 以后，就开始支持 TestNG 了，支持的方法包括：
将您的 TestNG 测试类继承 Spring 的测试父类：AbstractTransactionalTestNGSpringContextTests 或者 AbstractTestNGSpringContextTests，这样您的 TestNG 测试类内部就可以访问 applicationContext 成员变量了
不继承 Spring 父类，在测试类上使用 @TestExecutionListeners 注释标签，可以引入的监听器包括
DependencyInjectionTestExecutionListener：使得测试类拥有依赖注入特性
DirtiesContextTestExecutionListener：使得测试类拥有更新 applicationContext 能力
TransactionalTestExecutionListener：使得测试类拥有自动的事务管理能力
这里我们演示一下如何使用 Spring 提供的 TestNG 父类来进行测试。

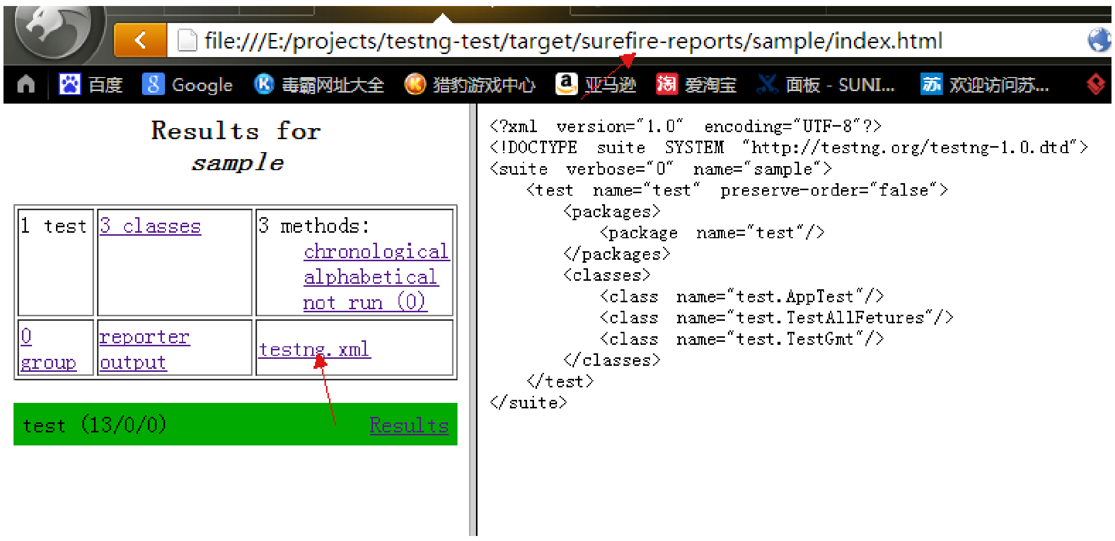


```java
@ContextConfiguration(locations = {
        "classpath:dao/datasource.xml",
        "classpath:dao/init-data.xml",
        "classpath:dao/dao.xml",
        "classpath:service/spring-service.xml",
})
public abstract class BaseTest extends AbstractTestNGSpringContextTests {

}
```

7.1.3 Testng 和 maven 集成



```xml
            <package name="test"/>
        </packages>
    </test>


    <test name="dao">
        <packages>
            <package name="dao"/>
        </packages>
    </test>
    <test name="service">
        <!--<packages>-->
        <!--<package name="service"/>-->
        <!--</packages>-->
        <classes>
            <class name="service.UserService">
                <methods>
                    <include name="addUser"/>
                </methods>
            </class>
        </classes>
    </test>
```


file:///E:/projects/testng-test/target/surefire-reports/sample/index.html
百度
Google
毒霸网址大全
猎豹游戏中心
亚马逊
爱淘宝
面板 - SUNI...
欢迎访问苏...

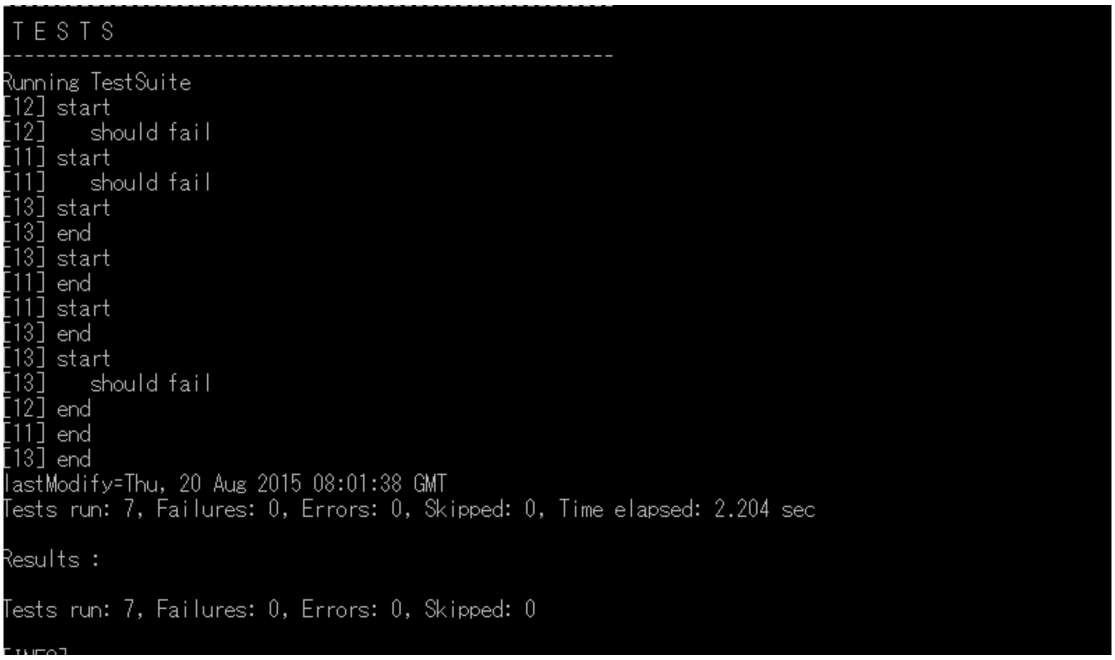
Results for
sample

1 test	3 classes	3 methods: chronological alphabetical not run (0)
0 group	reporter output	testng.xml

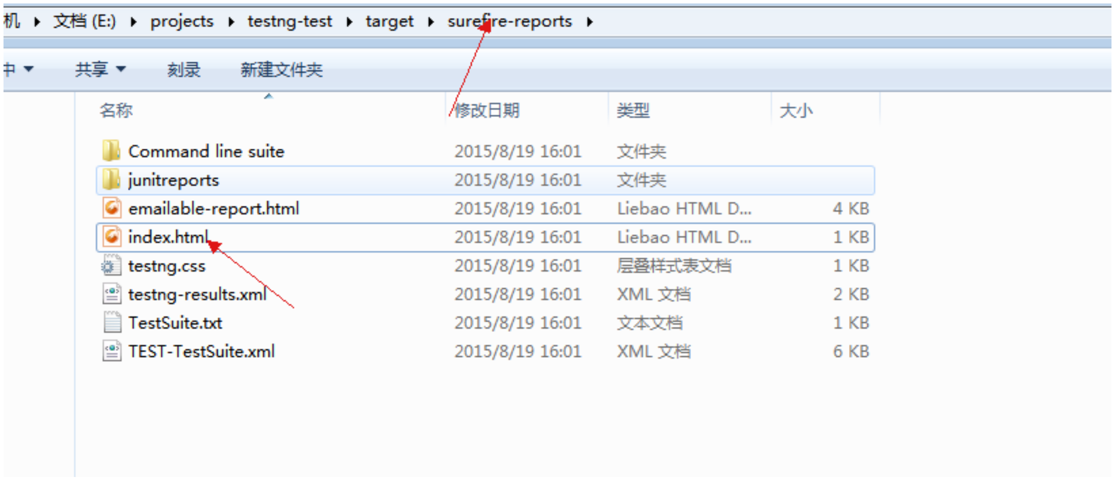
test (13/0/0) Results

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite verbose="0" name="sample">
    <test name="test" preserve-order="false">
        <packages>
            <package name="test"/>
        </packages>
        <classes>
            <class name="test.AppTest"/>
            <class name="test.TestAllFetures"/>
            <class name="test.TestGmt"/>
        </classes>
    </test>
</suite>
```

测试报告

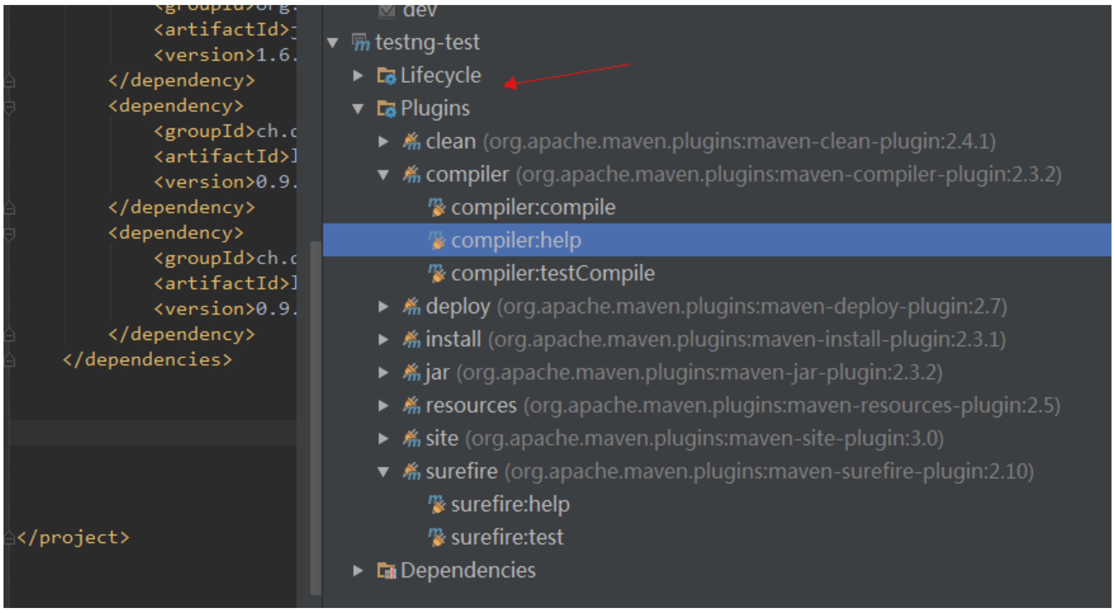


```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running TestSuite
[12] start
[12]    should fail
[11] start
[11]    should fail
[13] start
[13] end
[13] start
[11] end
[11] start
[13] end
[13] start
[13]    should fail
[12] end
[11] end
[13] end
lastModify=Thu, 20 Aug 2015 08:01:38 GMT
Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.204 sec

Results :

Tests run: 7, Failures: 0, Errors: 0, Skipped: 0
```



机 ▸ 文档 (E:) ▸ projects ▸ testng-test ▸ target ▸ surefire-reports ▸

名称	修改日期	类型	大小
Command line suite	2015/8/19 16:01	文件夹	
junitreports	2015/8/19 16:01	文件夹	
emailable-report.html	2015/8/19 16:01	Liebao HTML D...	4 KB
index.html	2015/8/19 16:01	Liebao HTML D...	1 KB
testng.css	2015/8/19 16:01	层叠样式表文档	1 KB
testng-results.xml	2015/8/19 16:01	XML 文档	2 KB
TestSuite.txt	2015/8/19 16:01	文本文档	1 KB
TEST-TestSuite.xml	2015/8/19 16:01	XML 文档	6 KB



```
            <artifactId>
            <version>1.6.
        </dependency>
        <dependency>
            <groupId>ch.q
            <artifactId>l
            <version>0.9.
        </dependency>
        <dependency>
            <groupId>ch.q
            <artifactId>l
            <version>0.9.
        </dependency>
    </dependencies>


</project>
```

```
▼ testng-test
  ▶ Lifecycle
  ▼ Plugins
    ▶ clean (org.apache.maven.plugins:maven-clean-plugin:2.4.1)
    ▼ compiler (org.apache.maven.plugins:maven-compiler-plugin:2.3.2)
        compiler:compile
        compiler:help
        compiler:testCompile
    ▶ deploy (org.apache.maven.plugins:maven-deploy-plugin:2.7)
    ▶ install (org.apache.maven.plugins:maven-install-plugin:2.3.1)
    ▶ jar (org.apache.maven.plugins:maven-jar-plugin:2.3.2)
    ▶ resources (org.apache.maven.plugins:maven-resources-plugin:2.5)
    ▶ site (org.apache.maven.plugins:maven-site-plugin:3.0)
    ▼ surefire (org.apache.maven.plugins:maven-surefire-plugin:2.10)
        surefire:help
        surefire:test
  ▶ Dependencies
```

http://maven.apache.org/surefire/maven-surefire-plugin/
surefire 插件
http://maven.apache.org/surefire/maven-surefire-plugin/

7.1.4 Maven Surefire Plugin

The Surefire Plugin is used during the `test` phase of the build lifecycle to execute the unit tests of an application. It generates reports in two different file formats:

- Plain text files (`*.txt`)
- XML files (`*.xml`)

By default, these files are generated at `${basedir}/target/surefire-reports`.

For an HTML format of the report, please see the Maven Surefire Report Plugin.

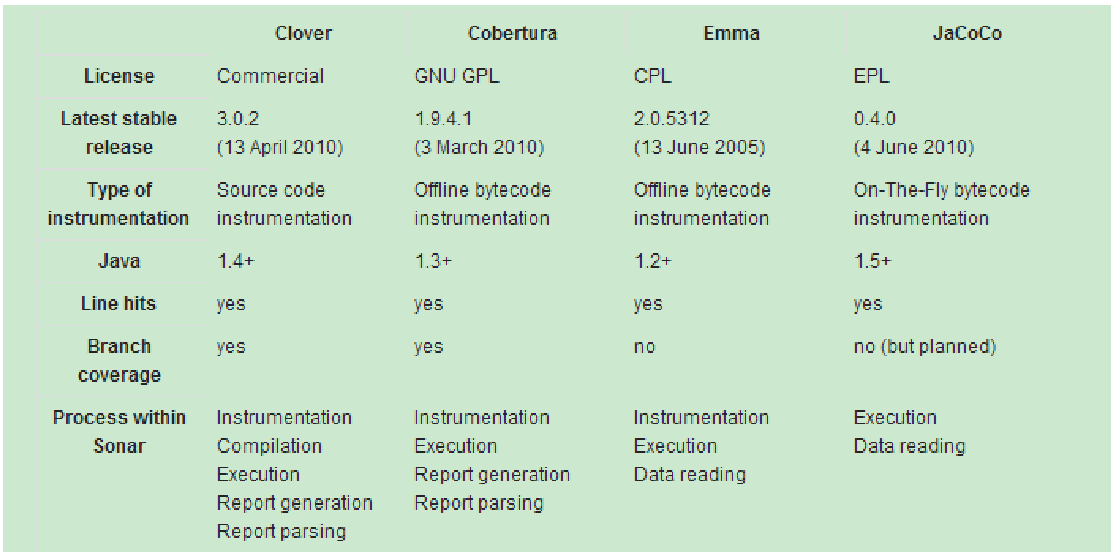
http://maven.apache.org/surefire/maven-surefire-report-plugin/

7.1.5 Mvn surefire 和 testng 集成原理

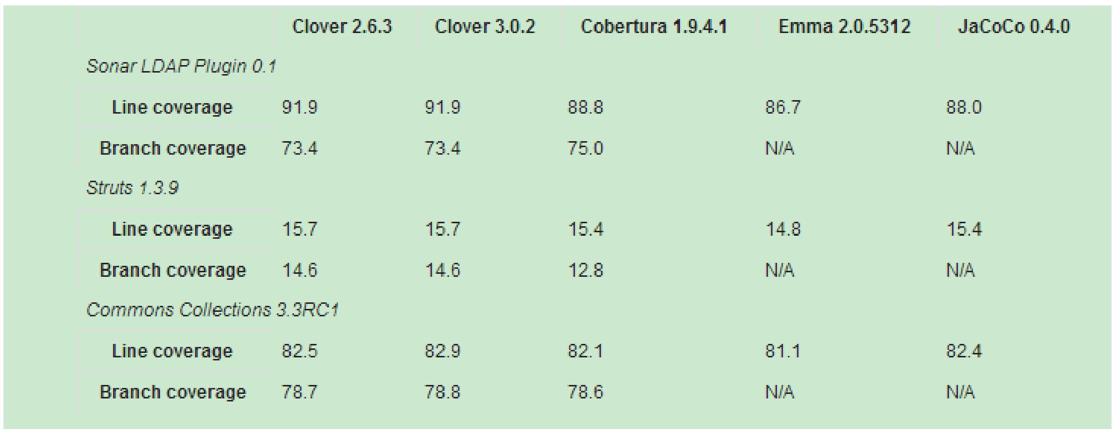
Surefire 在 maven test 过程中触发
查找测试包是否包含 testngjar
自动执行 src/test/java 下面 *Test.java 文件
http://maven.apache.org/surefire/maven-surefire-plugin/examples/inclusion-exclusion.html

8 测试覆盖率工具

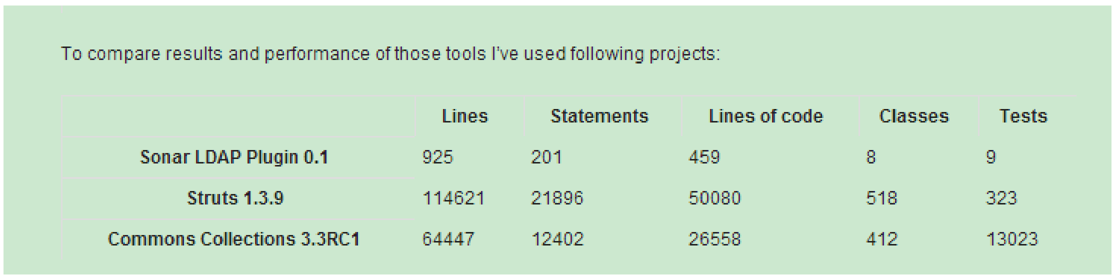
代码覆盖率测试的工具，比较常用的如下：
（1）开源：
JaCoCo http://www.eclemma.org/jacoco/
Cobetura http://cobertura.sourceforge.net/
Emma http://emma.sourceforge.net/
（2）商用：
Clover http://www.atlassian.com/software/clover/overview



	Clover	Cobertura	Emma	JaCoCo
License	Commercial	GNU GPL	CPL	EPL
Latest stable release	3.0.2 (13 April 2010)	1.9.4.1 (3 March 2010)	2.0.5312 (13 June 2005)	0.4.0 (4 June 2010)
Type of instrumentation	Source code instrumentation	Offline bytecode instrumentation	Offline bytecode instrumentation	On-The-Fly bytecode instrumentation
Java	1.4+	1.3+	1.2+	1.5+
Line hits	yes	yes	yes	yes
Branch coverage	yes	yes	no	no (but planned)
Process within Sonar	Instrumentation Compilation Execution Report generation Report parsing	Instrumentation Execution Report generation Report parsing	Instrumentation Execution Data reading	Execution Data reading



	Clover 2.6.3	Clover 3.0.2	Cobertura 1.9.4.1	Emma 2.0.5312	JaCoCo 0.4.0
Sonar LDAP Plugin 0.1					
Line coverage	91.9	91.9	88.8	86.7	88.0
Branch coverage	73.4	73.4	75.0	N/A	N/A
Struts 1.3.9					
Line coverage	15.7	15.7	15.4	14.8	15.4
Branch coverage	14.6	14.6	12.8	N/A	N/A
Commons Collections 3.3RC1					
Line coverage	82.5	82.9	82.1	81.1	82.4
Branch coverage	78.7	78.8	78.6	N/A	N/A



To compare results and performance of those tools I've used following projects:

	Lines	Statements	Lines of code	Classes	Tests
Sonar LDAP Plugin 0.1	925	201	459	8	9
Struts 1.3.9	114621	21896	50080	518	323
Commons Collections 3.3RC1	64447	12402	26558	412	13023

http://www.sonarqube.org/pick-your-code-coverage-tool-in-sonar-2-2/

8.1 Jacobo

Since version 2.0 EclEmma is based on the JaCoCo code coverage library. The Eclipse integration has its focus on supporting the individual developer in an highly interactive way. For automated builds please refer to JaCoCo documentation for integrations with other tools.
Originally EclEmma was inspired by and technically based on the great EMMA library developed

by Vlad Roubtsov.

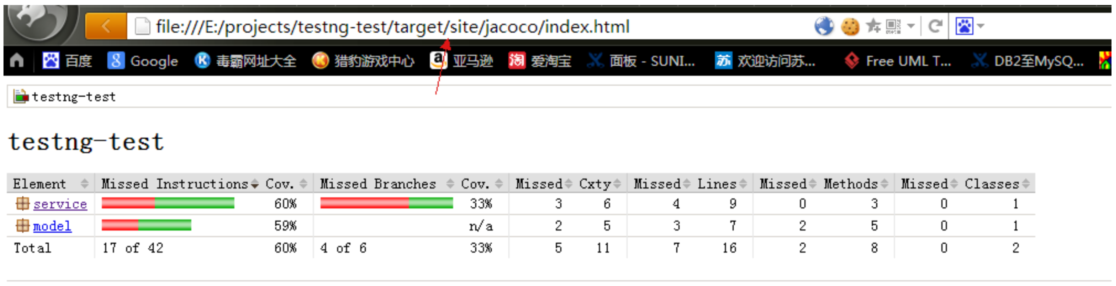
mvn clean org.jacoco:jacoco-maven-plugin:prepare-agent install -Dmaven.test.failure.ignore=true

mvn sonar:sonar

这步必须

org.jacoco:jacoco-maven-plugin:prepare-agent

mvn clean jacoco:prepare-agent install jacoco:report


file:///E:/projects/testng-test/target/site/jacoco/index.html

testng-test

testng-test

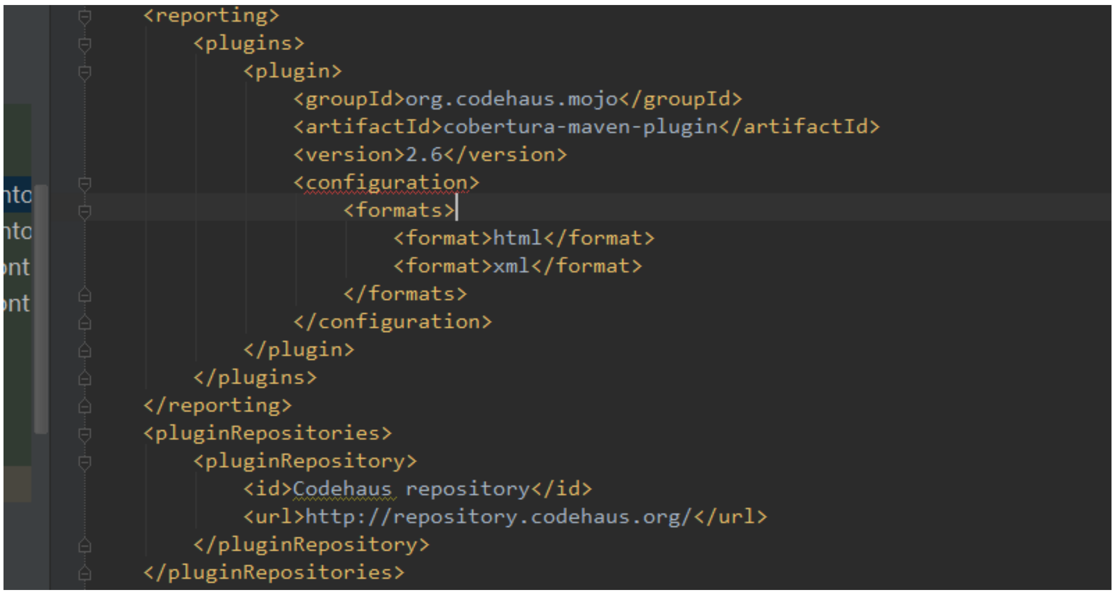
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
service		60%		33%	3	6	4	9	0	3	0	1
model		59%		n/a	2	5	3	7	2	5	0	1
Total	17 of 42	60%	4 of 6	33%	5	11	7	16	2	8	0	2

与 IDE 集成

EclEmma

8.2 Cobertura

8.2.1 与 maven 集成


```
<reporting>
    <plugins>
        <plugin>
            <groupId>org.codehaus.mojo</groupId>
            <artifactId>cobertura-maven-plugin</artifactId>
            <version>2.6</version>
            <configuration>
                <formats>
                    <format>html</format>
                    <format>xml</format>
                </formats>
            </configuration>
        </plugin>
    </plugins>
</reporting>
<pluginRepositories>
    <pluginRepository>
        <id>Codehaus repository</id>
        <url>http://repository.codehaus.org/</url>
    </pluginRepository>
</pluginRepositories>
```

8.2.2 支持的命令

cobertura:check Check the coverage percentages for unit tests from the last instrumentation, and optionally fail the build if the targets are not met.

cobertura:check-integration-test Check the coverage percentages for unit tests and integration tests from the last instrumentation, and optionally fail the build if the targets are not met.

cobertura:clean Clean up the files that Cobertura Maven Plugin has created during instrumentation.

cobertura:dump-datafile Output the contents of Cobertura's data file to the command line.

cobertura:instrument Instrument the compiled classes.

cobertura:cobertura Instrument the compiled classes, run the unit tests and generate a Cobertura report.

cobertura:cobertura-integration-test Instrument the compiled classes, run the unit tests and integration tests and generate a Cobertura report.

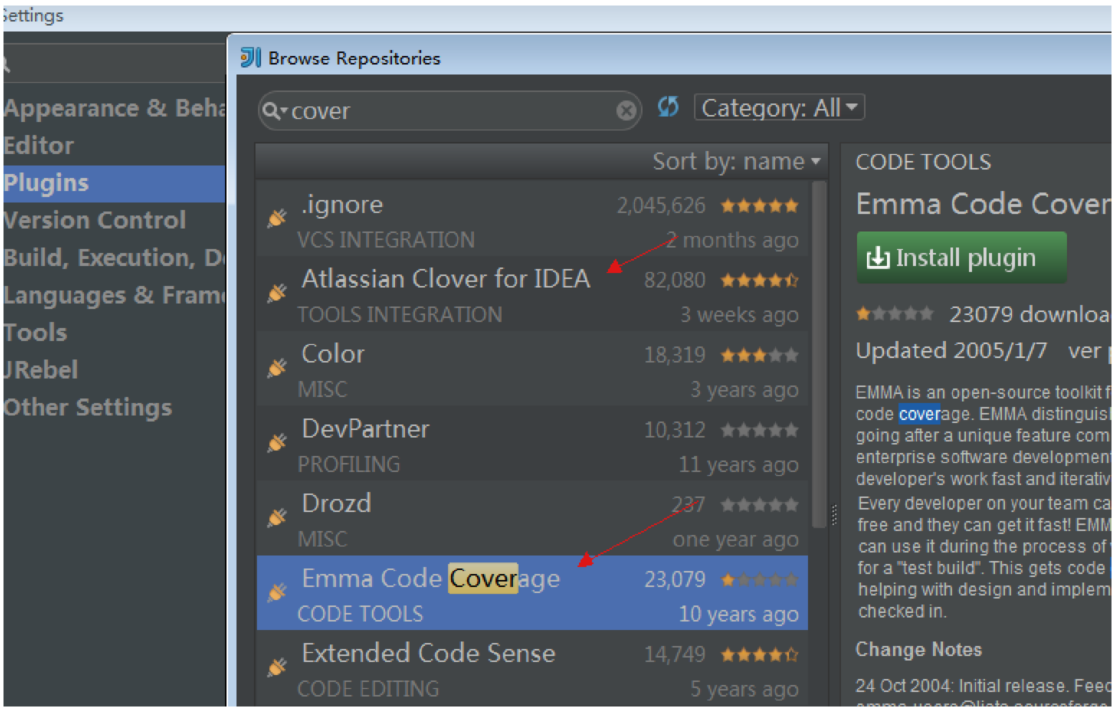
参考文档

http://cobertura.github.io/cobertura/

http://www.mojohaus.org/cobertura-maven-plugin/

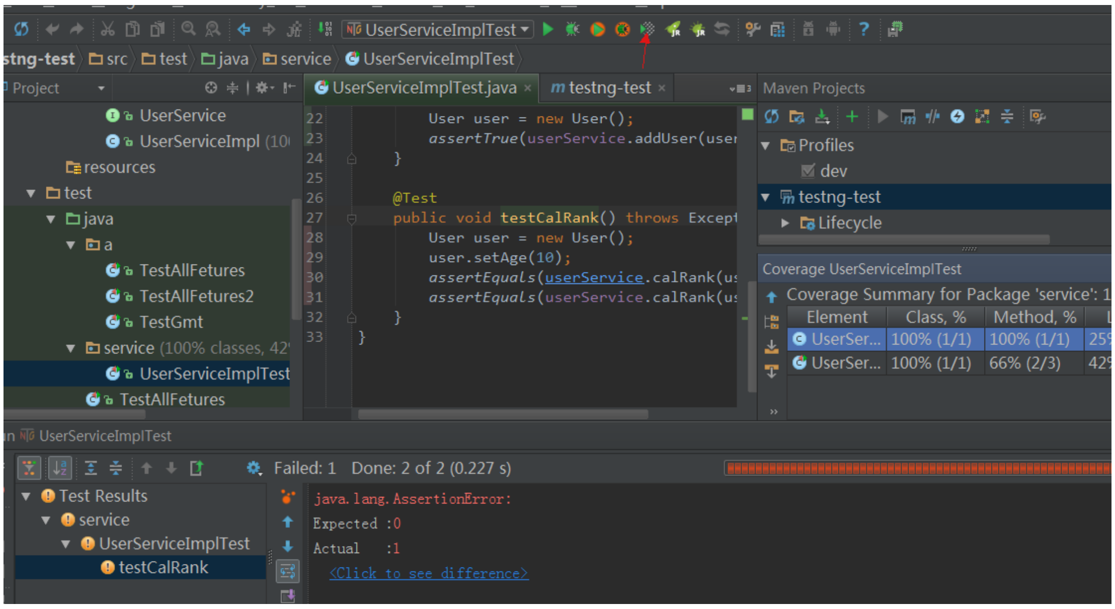
http://www.mojohaus.org/cobertura-maven-plugin/usage.html

8.2.3 和 IDE 集成


Settings
Appearance & Beha
Editor
Plugins
Version Control
Build, Execution, D
Languages & Frame
Tools
JRebel
Other Settings
Browse Repositories
cover
Category: All
Sort by: name
.ignore 2,045,626 VCS INTEGRATION 2 months ago
Atlassian Clover for IDEA 82,080 TOOLS INTEGRATION 3 weeks ago
Color 18,319 MISC 3 years ago
DevPartner 10,312 PROFILING 11 years ago
Drozd 237 MISC one year ago
Emma Code Coverage 23,079 CODE TOOLS 10 years ago
Extended Code Sense 14,749 CODE EDITING 5 years ago
CODE TOOLS
Emma Code Cover
Install plugin
23079 downloa
Updated 2005/1/7 ver
Change Notes

EMMA is an open-source toolkit f
code coverage. EMMA distinguis
going after a unique feature com
enterprise software developmen
developer's work fast and iterativ
Every developer on your team ca
free and they can get it fast! EMM
can use it during the process of
for a "test build". This gets code
helping with design and implem
checked in.

24 Oct 2004: Initial release. Feed



```
        User user = new User();
        assertTrue(userService.addUser(user
    }

    @Test
    public void testCalRank() throws Except
        User user = new User();
        user.setAge(10);
        assertEquals(userService.calRank(us
        assertEquals(userService.calRank(us
    }
}
```

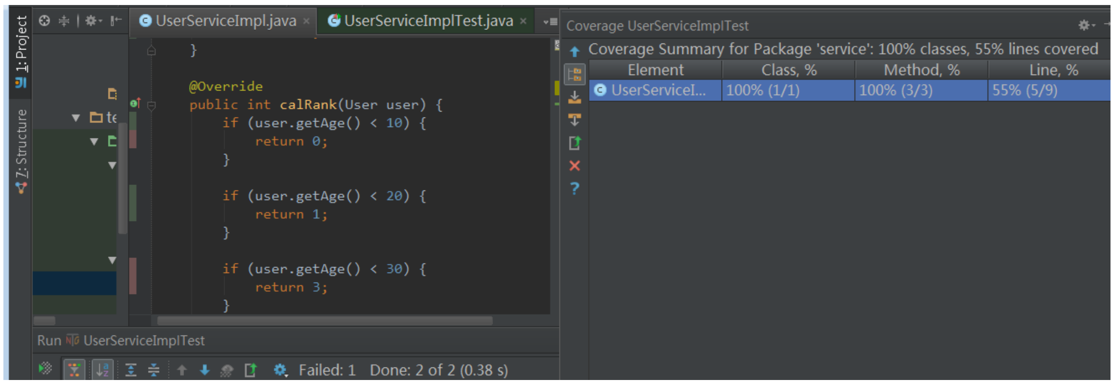
Coverage UserServiceImplTest

Coverage Summary for Package 'service': 1

Element	Class, %	Method, %	L
UserSer...	100% (1/1)	100% (1/1)	25
UserSer...	100% (1/1)	66% (2/3)	42

Failed: 1   Done: 2 of 2 (0.227 s)

```
java.lang.AssertionError:
Expected :0
Actual   :1
 <Click to see difference>
```



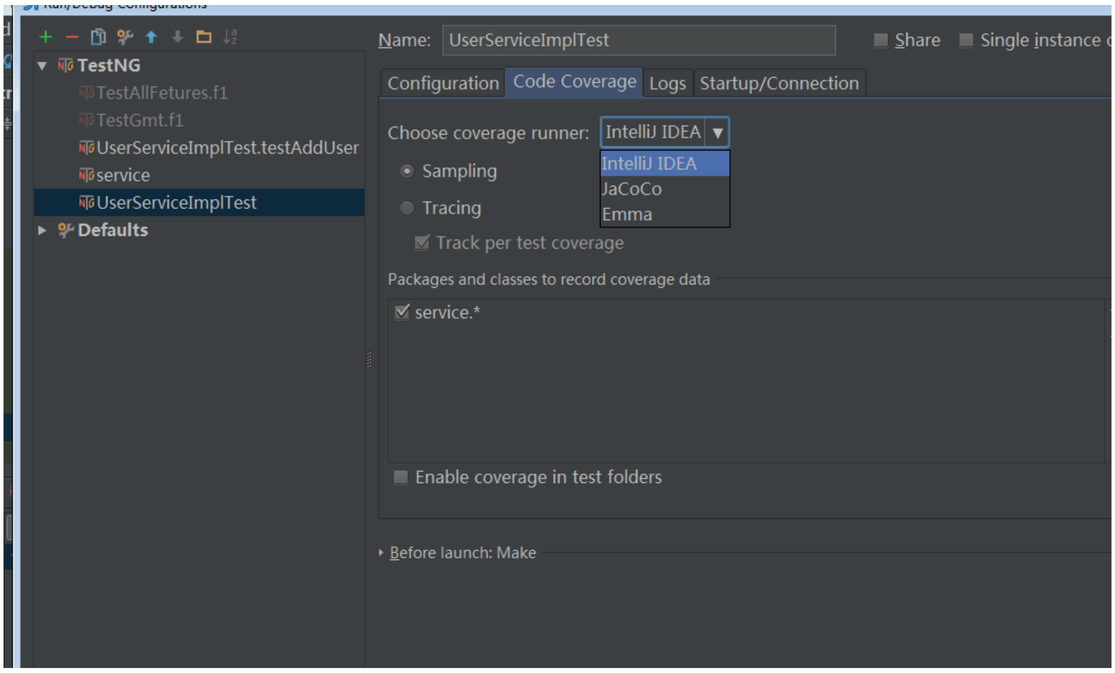
```
    }

    @Override
    public int calRank(User user) {
        if (user.getAge() < 10) {
            return 0;
        }

        if (user.getAge() < 20) {
            return 1;
        }

        if (user.getAge() < 30) {
            return 3;
        }
```

Coverage UserServiceImplTest

Coverage Summary for Package 'service': 100% classes, 55% lines covered

Element	Class, %	Method, %	Line, %
UserServiceI...	100% (1/1)	100% (3/3)	55% (5/9)

Failed: 1   Done: 2 of 2 (0.38 s)



Name: UserServiceImplTest   Share   Single instance

Configuration | Code Coverage | Logs | Startup/Connection

Choose coverage runner: IntelliJ IDEA ▾ (IntelliJ IDEA / JaCoCo / Emma)

[✓ Sampling]

Tracing

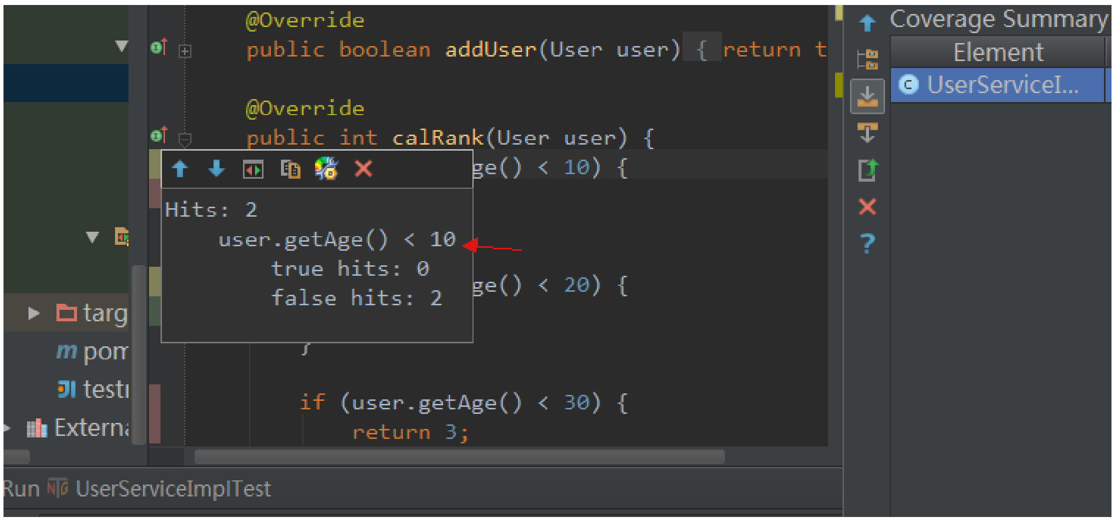
Track per test coverage

Packages and classes to record coverage data

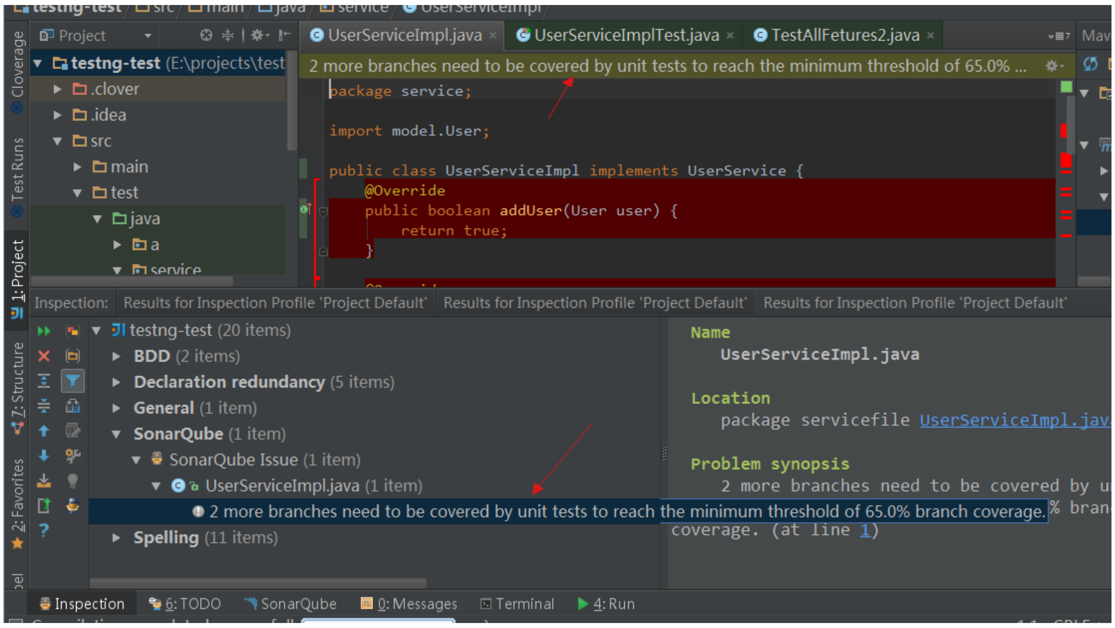
☑ service.*

☐ Enable coverage in test folders

Before launch: Make



```
@Override
public boolean addUser(User user) { return t

@Override
public int calRank(User user) {
                              ge() < 10) {
Hits: 2
    user.getAge() < 10
        true hits: 0
        false hits: 2
                              ge() < 20) {

    if (user.getAge() < 30) {
        return 3;
```



```
2 more branches need to be covered by unit tests to reach the minimum threshold of 65.0% ...
package service;

import model.User;

public class UserServiceImpl implements UserService {
    @Override
    public boolean addUser(User user) {
        return true;
    }
```

```
testng-test (20 items)
    BDD (2 items)
    Declaration redundancy (5 items)
    General (1 item)
    SonarQube (1 item)
        SonarQube Issue (1 item)
            UserServiceImpl.java (1 item)
                2 more branches need to be covered by unit tests to reach the minimum threshold of 65.0% branch coverage.
    Spelling (11 items)
```

```
Name
    UserServiceImpl.java

Location
    package servicefile UserServiceImpl.jav

Problem synopsis
    2 more branches need to be covered by u...% bran
coverage. (at line 1)
```

8.2.4 与 jenkins 集成

https://wiki.jenkins-ci.org/display/JENKINS/Cobertura+Plugin
https://wiki.jenkins-ci.org/display/JENKINS/Clover+Plugin

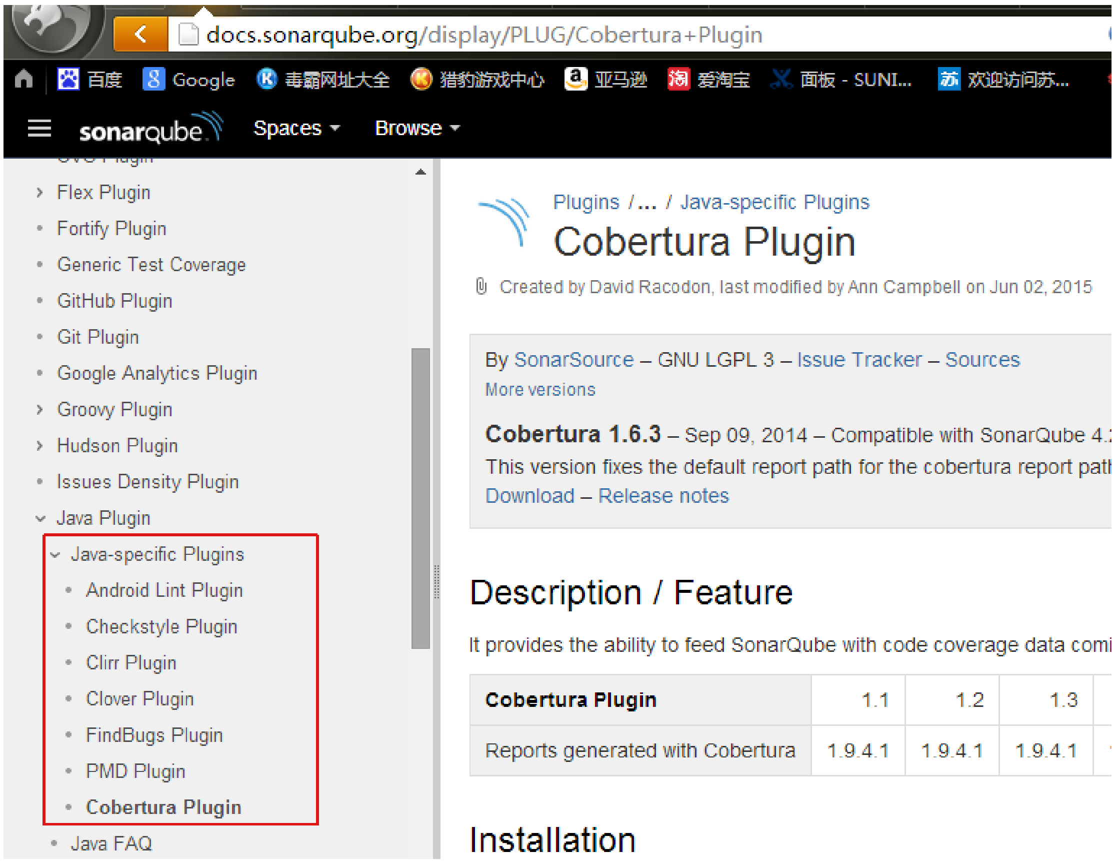
8.2.1 与 sonar 集成

https://github.com/SonarSource/sonar-examples/tree/master/projects/languages/java/code-coverage/ut/ut-maven-cobertura
插件
http://docs.sonarqube.org/display/PLUG/Cobertura+Plugin
http://docs.sonarqube.org/display/PLUG/Clover+Plugin

http://downloads.sonarsource.com/plugins/org/codehaus/sonar-plugins/sonar-clover-plugin/3.0/sonar-clover-plugin-3.0.jar

8.2.2 Installation

1. Install the plugin through the Update Center or download it into the *SONARQUBE_HOME/extensions/plugins* directory



docs.sonarqube.org/display/PLUG/Cobertura+Plugin

Flex Plugin
Fortify Plugin
Generic Test Coverage
GitHub Plugin
Git Plugin
Google Analytics Plugin
Groovy Plugin
Hudson Plugin
Issues Density Plugin
Java Plugin
Java-specific Plugins
Android Lint Plugin
Checkstyle Plugin
Clirr Plugin
Clover Plugin
FindBugs Plugin
PMD Plugin
Cobertura Plugin
Java FAQ

Plugins / ... / Java-specific Plugins

Cobertura Plugin

Created by David Racodon, last modified by Ann Campbell on Jun 02, 2015

By SonarSource – GNU LGPL 3 – Issue Tracker – Sources
More versions

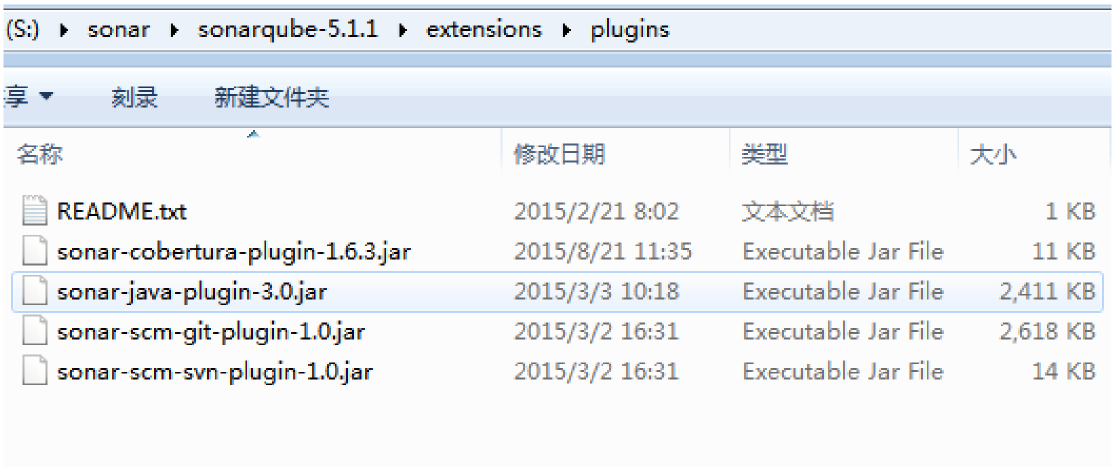
Cobertura 1.6.3 – Sep 09, 2014 – Compatible with SonarQube 4.
This version fixes the default report path for the cobertura report pat
Download – Release notes

Description / Feature

It provides the ability to feed SonarQube with code coverage data comi

Cobertura Plugin	1.1	1.2	1.3
Reports generated with Cobertura	1.9.4.1	1.9.4.1	1.9.4.1

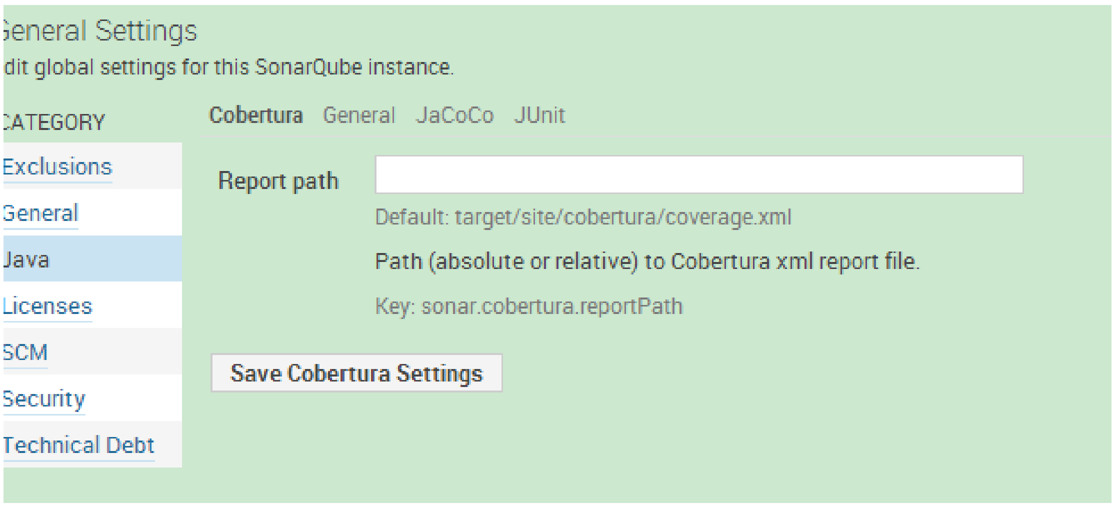
Installation



(S:) ▸ sonar ▸ sonarqube-5.1.1 ▸ extensions ▸ plugins

名称	修改日期	类型	大小
README.txt	2015/2/21 8:02	文本文档	1 KB
sonar-cobertura-plugin-1.6.3.jar	2015/8/21 11:35	Executable Jar File	11 KB
sonar-java-plugin-3.0.jar	2015/3/3 10:18	Executable Jar File	2,411 KB
sonar-scm-git-plugin-1.0.jar	2015/3/2 16:31	Executable Jar File	2,618 KB
sonar-scm-svn-plugin-1.0.jar	2015/3/2 16:31	Executable Jar File	14 KB

2. The default location of the XML Cobertura report is : target/site/cobertura/coverage.xml . You can change it in Configure in the Settings > General Settings > Java > Cobertura page
3. Restart the SonarQube server


ieneral Settings

dit global settings for this SonarQube instance.

Report path

Default: target/site/cobertura/coverage.xml

Path (absolute or relative) to Cobertura xml report file.

Key: sonar.cobertura.reportPath

http://docs.sonarqube.org/display/PLUG/Code+Coverage+by+Unit+Tests+for+Java+Project

8.2.3 Usage

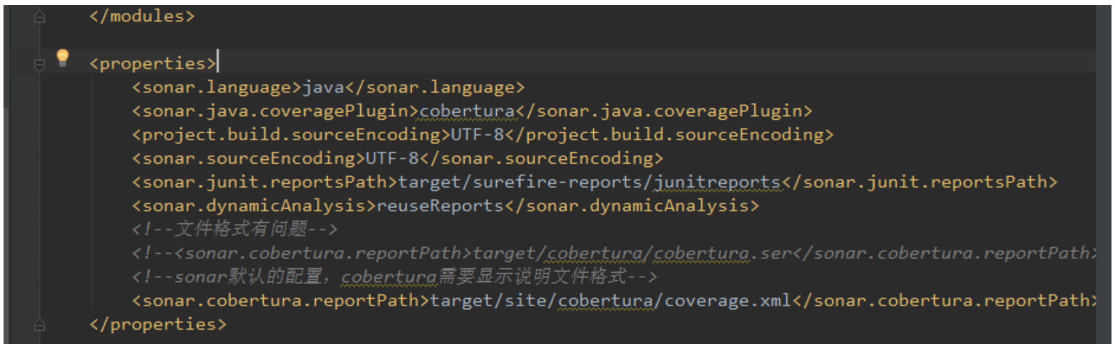
- Build the project and execute the unit tests:

```
• mvn clean compile
• mvn cobertura:cobertura  注意
• mvn cobertura:cobertura -Dcobertura.report.format=xml   默认的
  不知 sar 方式
```

- Analyze the project with SonarQube using Maven:

```
mvn sonar:sonar
```

8.2.4 mvn 配置支持 sonar



```xml
    </modules>

    <properties>
        <sonar.language>java</sonar.language>
        <sonar.java.coveragePlugin>cobertura</sonar.java.coveragePlugin>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <sonar.sourceEncoding>UTF-8</sonar.sourceEncoding>
        <sonar.junit.reportsPath>target/surefire-reports/junitreports</sonar.junit.reportsPath>
        <sonar.dynamicAnalysis>reuseReports</sonar.dynamicAnalysis>
        <!--文件格式有问题-->
        <!--<sonar.cobertura.reportPath>target/cobertura/cobertura.ser</sonar.cobertura.reportPath
        <!--sonar默认的配置，cobertura需要显示说明文件格式-->
        <sonar.cobertura.reportPath>target/site/cobertura/coverage.xml</sonar.cobertura.reportPath
    </properties>
```

参考：
https://github.com/SonarSource/sonar-examples/blob/master/projects/languages/java/code-coverage/ut/ut-maven-cobertura/pom.xml

8.3 Clover

Why does Clover use source code instrumentation?

Possible feature	JVMDI/PI	Bytecode instrumentation	Source code instrumentation
Gathers method coverage	yes	yes	yes
Gathers statement coverage	line only	indirectly	yes
Gathers branch coverage	indirectly	indirectly	yes
Can work without source	yes	yes	no
Requires separate build	no	no	yes

Possible feature	JVMDI/PI	Bytecode instrumentation	Source code instrumentation
Requires specialised runtime	yes	yes	no
Gathers source metrics	no	no	yes
View coverage data inline with source	not accurate	not accurate	yes
Source level directives to control coverage gathering	no	no	yes
Control which entities are reported on	limited	limited	yes
Compilation time	no impact	variable	variable
Runtime performace	high impact	variable	variable
Container friendly	no	no	yes

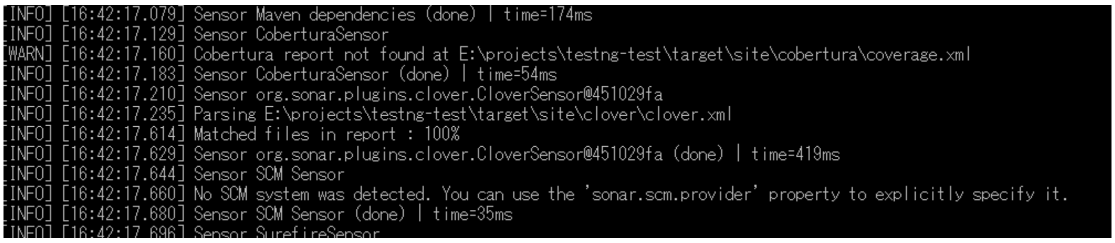
8.3.1 clover 和 maven 集成

```
mvn clean clover2:setup test clover2:aggregate clover2:clover
```

There are four basic parts executed when recording code coverage with Clover.

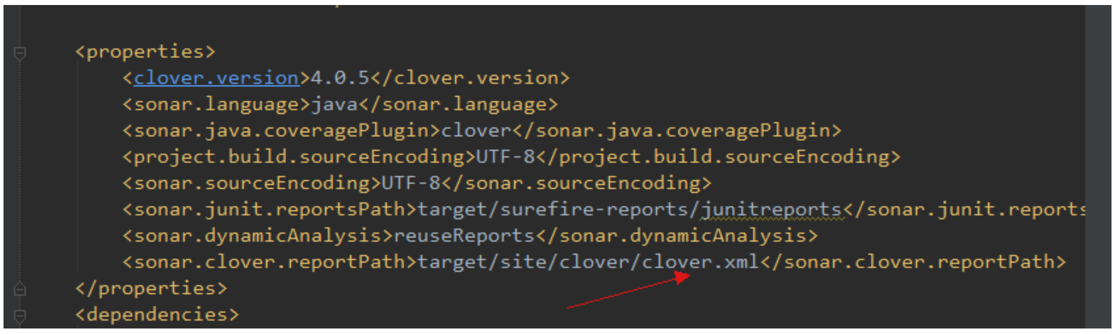
1. The **clover2:setup** goal will instrument your Java source files.

2. The **test** phase is Maven 2 and 3's standard command for running a unit test phase.
3. The **clover2:aggregate** goal is used for merging coverage data generated by multi-module projects.
4. The **clover2:clover** goal generates an HTML, XML, PDF or JSON report.



```
[INFO] [16:42:17.079] Sensor Maven dependencies (done) | time=174ms
[INFO] [16:42:17.129] Sensor CoberturaSensor
[WARN] [16:42:17.160] Cobertura report not found at E:\projects\testng-test\target\site\cobertura\coverage.xml
[INFO] [16:42:17.183] Sensor CoberturaSensor (done) | time=54ms
[INFO] [16:42:17.210] Sensor org.sonar.plugins.clover.CloverSensor@451029fa
[INFO] [16:42:17.235] Parsing E:\projects\testng-test\target\site\clover\clover.xml
[INFO] [16:42:17.614] Matched files in report : 100%
[INFO] [16:42:17.629] Sensor org.sonar.plugins.clover.CloverSensor@451029fa (done) | time=419ms
[INFO] [16:42:17.644] Sensor SCM Sensor
[INFO] [16:42:17.660] No SCM system was detected. You can use the 'sonar.scm.provider' property to explicitly specify it.
[INFO] [16:42:17.680] Sensor SCM Sensor (done) | time=35ms
[INFO] [16:42:17.696] Sensor SurefireSensor
```

8.3.2 和 sonar 集成

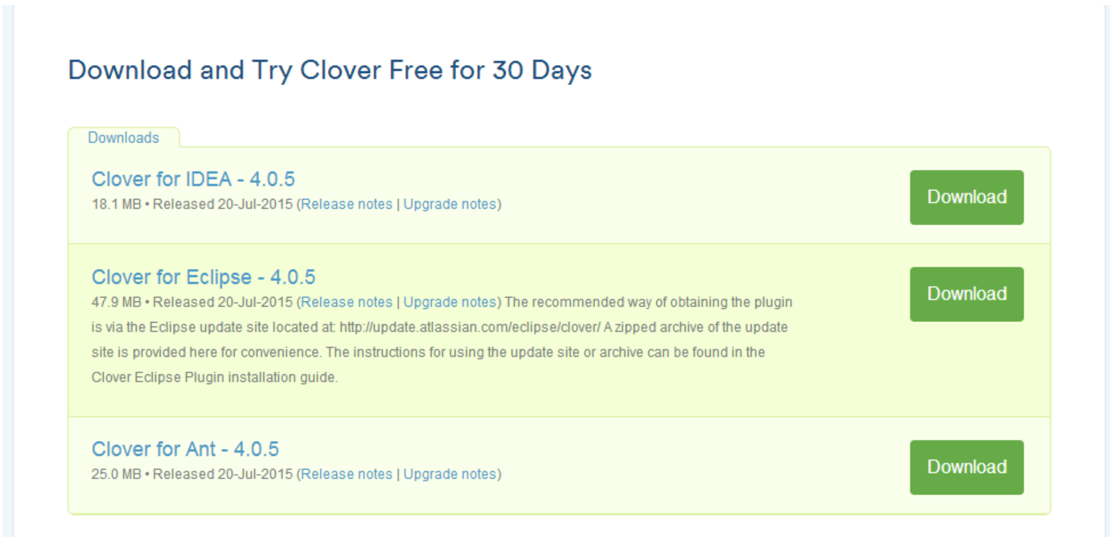


```
<properties>
    <clover.version>4.0.5</clover.version>
    <sonar.language>java</sonar.language>
    <sonar.java.coveragePlugin>clover</sonar.java.coveragePlugin>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <sonar.sourceEncoding>UTF-8</sonar.sourceEncoding>
    <sonar.junit.reportsPath>target/surefire-reports/junitreports</sonar.junit.reports
    <sonar.dynamicAnalysis>reuseReports</sonar.dynamicAnalysis>
    <sonar.clover.reportPath>target/site/clover/clover.xml</sonar.clover.reportPath>
</properties>
<dependencies>
```

8.3.3 Clover 和 IDE 集成

8.3.3.1下载

https://www.atlassian.com/software/clover/download



Download and Try Clover Free for 30 Days

Downloads

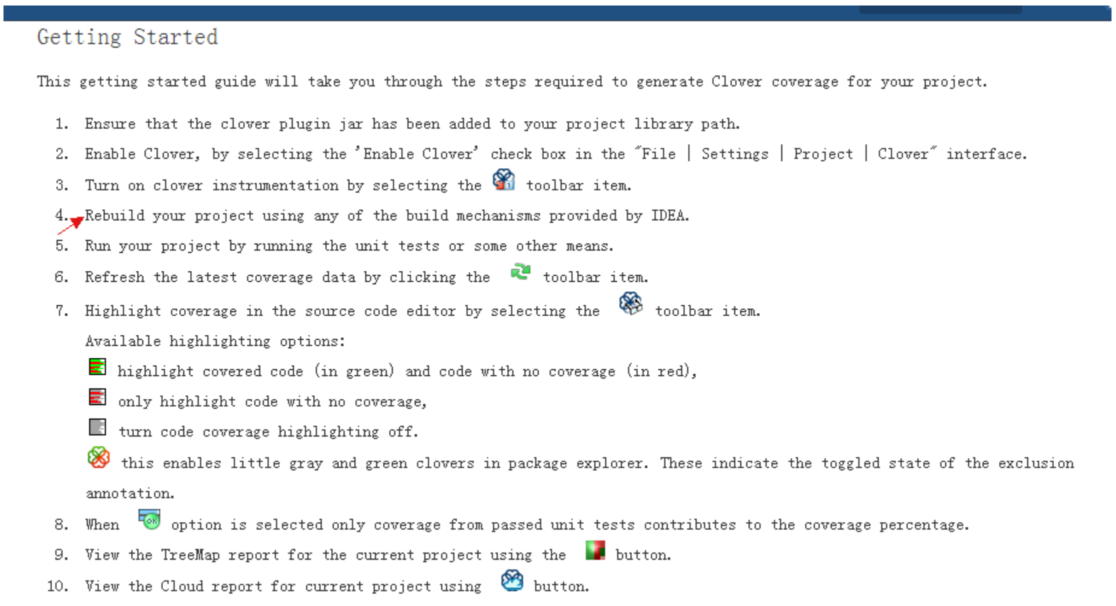
Clover for IDEA - 4.0.5
18.1 MB • Released 20-Jul-2015 (Release notes | Upgrade notes)

Clover for Eclipse - 4.0.5
47.9 MB • Released 20-Jul-2015 (Release notes | Upgrade notes) The recommended way of obtaining the plugin is via the Eclipse update site located at: http://update.atlassian.com/eclipse/clover/ A zipped archive of the update site is provided here for convenience. The instructions for using the update site or archive can be found in the Clover Eclipse Plugin installation guide.

Clover for Ant - 4.0.5
25.0 MB • Released 20-Jul-2015 (Release notes | Upgrade notes)

https://www.atlassian.com/software/clover/downloads/binary/clover-idea-4.0.5.jar
https://downloads.atlassian.com/software/clover/downloads/clover-idea-4.0.5.jar

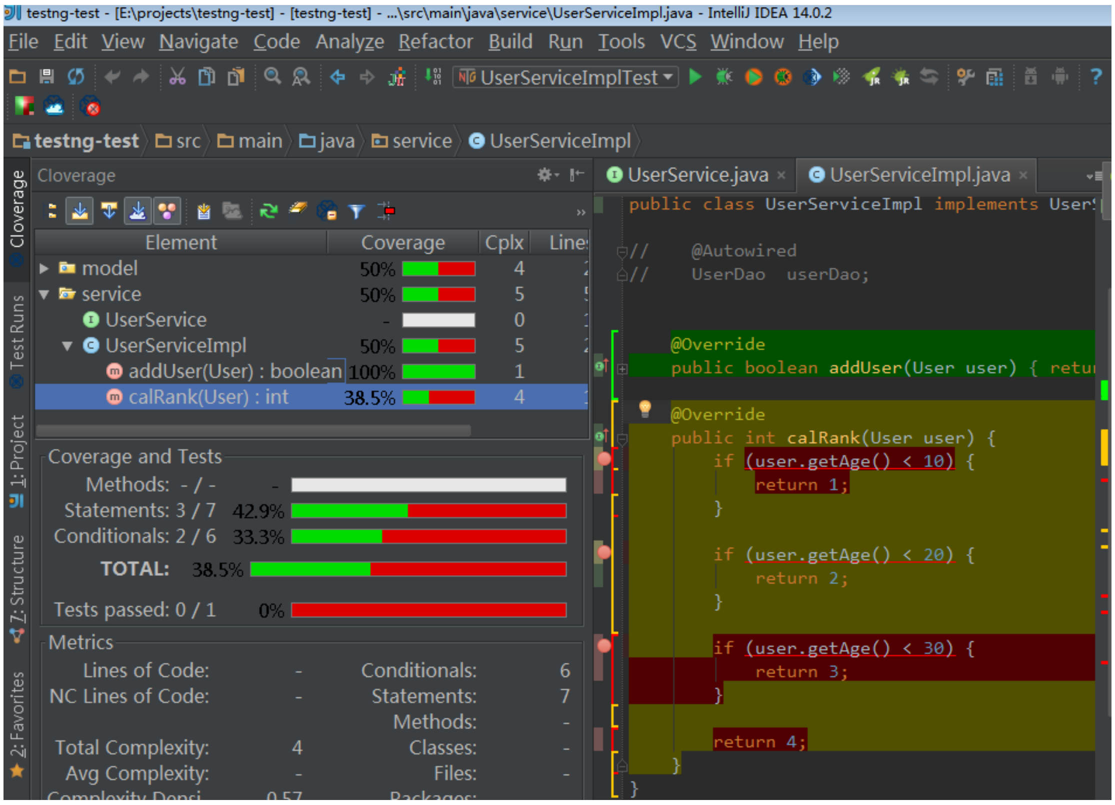
8.3.3.2使用



Getting Started

This getting started guide will take you through the steps required to generate Clover coverage for your project.

1. Ensure that the clover plugin jar has been added to your project library path.
2. Enable Clover, by selecting the 'Enable Clover' check box in the "File | Settings | Project | Clover" interface.
3. Turn on clover instrumentation by selecting the toolbar item.
4. Rebuild your project using any of the build mechanisms provided by IDEA.
5. Run your project by running the unit tests or some other means.
6. Refresh the latest coverage data by clicking the toolbar item.
7. Highlight coverage in the source code editor by selecting the toolbar item.
   Available highlighting options:
   highlight covered code (in green) and code with no coverage (in red),
   only highlight code with no coverage,
   turn code coverage highlighting off.
   this enables little gray and green clovers in package explorer. These indicate the toggled state of the exclusion annotation.
8. When option is selected only coverage from passed unit tests contributes to the coverage percentage.
9. View the TreeMap report for the current project using the button.
10. View the Cloud report for current project using button.


testng-test - [E:\projects\testng-test] - [testng-test] - ...\src\main\java\service\UserServiceImpl.java - IntelliJ IDEA 14.0.2
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
UserServiceImplTest
testng-test src main java service UserServiceImpl
Cloverage
Cloverage
Test Runs
1: Project
7: Structure
2: Favorites
UserService.java
UserServiceImpl.java

Element	Coverage		Cplx
model	50%	🟩🟥	4
service	50%	🟩🟥	5
UserService	-	⬜	0
UserServiceImpl	50%	🟩🟥	5
addUser(User) : boolean	100%	🟩	1
calRank(User) : int	38.5%	🟩🟥	4

```
Coverage and Tests
   Methods:  - / -       -
Statements: 3 / 7    42.9%
Conditionals: 2 / 6  33.3%
      TOTAL:        38.5%

Tests passed: 0 / 1     0%

Metrics
    Lines of Code:     -      Conditionals:   6
 NC Lines of Code:     -        Statements:   7
                                  Methods:    -
 Total Complexity:     4          Classes:    -
   Avg Complexity:     -            Files:    -
```

```java
public class UserServiceImpl implements User

//    @Autowired
//    UserDao  userDao;


    @Override
    public boolean addUser(User user) { retu

    @Override
    public int calRank(User user) {
        if (user.getAge() < 10) {
            return 1;
        }

        if (user.getAge() < 20) {
            return 2;
        }

        if (user.getAge() < 30) {
            return 3;
        }

        return 4;
    }
}
```

参考：
https://confluence.atlassian.com/display/CLOVER/1.+Clover+for+IDEA+in+10+minutes
http://blog.csdn.net/yanmingming1989/article/details/8557981
http://www.taobaotest.com/blogs/qa?bid=6425
https://docs.atlassian.com/maven-clover2-plugin/latest/architecture/architecture.html#clover-check.html
https://confluence.atlassian.com/pages/viewpage.action?pageId=79986998
https://confluence.atlassian.com/display/CLOVER/Clover-for-IDEA+Installation+Guide
https://docs.atlassian.com/maven-clover2-plugin/latest/#instrument-mojo.html

8.3.4 与 jenkins 集成

https://wiki.jenkins-ci.org/display/JENKINS/Clover+Plugin

9 其他测试工具

9.1 JTest

Jtest 一体化的软件，支持 findbug、junit 、catus 等功能
such as static analysis, peer review, unit testing, coverage analysis, and runtime error detection

参考资料
http://www.ibm.com/developerworks/cn/java/j-lo-jtest/
https://www.parasoft.com/product/jtest/

9.2 Unitils

整合了几个测试框架，但是整合的粒度和支持情况并不理想，支持的框架有限，需要单独学习。成本比较高。
没有对 mockito 整合
没有对 testng 整合
Db 测试只是整合了 dbunit，不如直接使用方便。

参考资料
http://www.unitils.org/summary.html

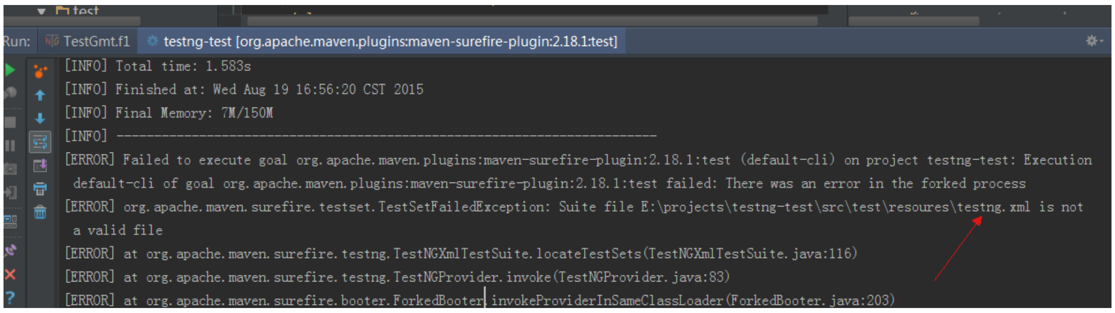
9.3 DbUnit

http://dbunit.sourceforge.net/

DbUnit is a JUnit extension
DbUnit has the ability to export and import your database data to and from XML datasets. Since version 2.0, DbUnit can also work with very large datasets when used in streaming mode. DbUnit can also help you to verify that your database data match an expected set of values.

10 常见错误

10.1 没有配置文件


Run: TestGmt.f1
testng-test [org.apache.maven.plugins:maven-surefire-plugin:2.18.1:test]

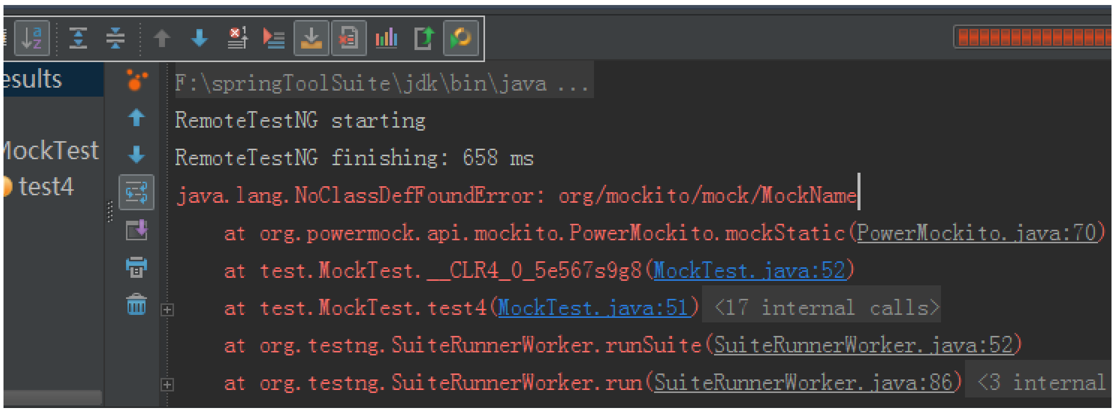
```
[INFO] Total time: 1.583s
[INFO] Finished at: Wed Aug 19 16:56:20 CST 2015
[INFO] Final Memory: 7M/150M
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-surefire-plugin:2.18.1:test (default-cli) on project testng-test: Execution
 default-cli of goal org.apache.maven.plugins:maven-surefire-plugin:2.18.1:test failed: There was an error in the forked process
[ERROR] org.apache.maven.surefire.testset.TestSetFailedException: Suite file E:\projects\testng-test\src\test\resoures\testng.xml is not
 a valid file
[ERROR] at org.apache.maven.surefire.testng.TestNGXmlTestSuite.locateTestSets(TestNGXmlTestSuite.java:116)
[ERROR] at org.apache.maven.surefire.testng.TestNGProvider.invoke(TestNGProvider.java:83)
[ERROR] at org.apache.maven.surefire.booter.ForkedBooter.invokeProviderInSameClassLoader(ForkedBooter.java:203)
```

10.2 没有测试结果

是否提交到 sonar:sonar 上，另外检查测试报告是否正确。

10.3 版本兼容问题

Mockito 和 powermock



```
F:\springToolSuite\jdk\bin\java ...
RemoteTestNG starting
RemoteTestNG finishing: 658 ms
java.lang.NoClassDefFoundError: org/mockito/mock/MockName
    at org.powermock.api.mockito.PowerMockito.mockStatic(PowerMockito.java:70)
    at test.MockTest.__CLR4_0_5e567s9g8(MockTest.java:52)
    at test.MockTest.test4(MockTest.java:51) <17 internal calls>
    at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:52)
    at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:86) <3 internal
```

11 参考文献

11.1 文档

软件测试的艺术

有效的单元测试
.NET 单元测试艺术
JUnit 实战
TestNG　http://testng.org/doc/index.html
代码覆盖率 http://www.cnblogs.com/coderzh/archive/2009/03/29/1424344.html
Mock 对比 http://www.ibm.com/developerworks/cn/java/j-lo-powermock/
http://blog.csdn.net/ht99582/article/details/43152921
http://blog.csdn.net/ht99582/article/details/43152921
http://jinnianshilongnian.iteye.com/blog/2108400
http://jinnianshilongnian.iteye.com/blog/2108400
http://jinnianshilongnian.iteye.com/blog/2106184
http://rensanning.iteye.com/blog/2002371
http://www.blogjava.net/gentoo1439/archive/2007/07/29/133055.html
http://book.51cto.com/art/201203/321054.htm
http://www.eclemma.org/jacoco/index.html
http://liangruijun.blog.51cto.com/3061169/803473/
http://www.eclemma.org/
http://www.sonarqube.org/unit-test-execution-in-sonarqube/

11.2 代码

Spring 测试
Common util 测试
Testing junit 测试
https://github.com/cbeust/testng/tree/master/src/test/java
Petshop 测试
http://www.open-open.com/lib/view/open1439793373083.html