

O2O业务质量体系建设

-----持续集成实践

孙斌

目录

- 持续集成概述
- 测试环境的演变
- 代码静态扫描
- 代码覆盖率收集
- 自动化测试

● 持续集成概念

持续集成是一种软件开发实践，即团队开发成员经常集成他们的工作，通常每个成员每天至少集成一次，也就意味着每天可能会发生多次集成。每次集成都通过自动化的构建（包括编译，发布，自动化测试）来验证，从而尽快地发现集成错误。许多团队发现这个过程可以大大减少集成的问题，让团队能够更快的开发内聚的软件。

● 持续集成的价值

1. 减少风险

越早发现的错误，修正的费用越低。

及时检查软件测试的健康状况，减少发布时的风险。

2. 减少重复性的工作

通过自动化的持续集成可以将编译，部署，测试等动作都变成自动化的，无需太多人工干预

3. 任何时间，任何地点都可以部署软件

利用持续集成，可以经常对源代码进行一些小改动，并将这些改动和其他的代码进行集成。如果出现问题，项目成员马上就会被通知到。

● 持续集成对到家的意义

京东到家是基于地理位置的o2o生活平台。虽然我们的业务和京东商城不太一样，但是我们的系统复杂度不低于商城，从成立到现在有200+个服务。系统之间依赖度较高，但是由于我们业务的特殊性，迭代速度比较快。如何使各个服务能够快速部署快速测试上线，对测试和研发工程师形成了很大的挑战。我们引入了持续集成的概念，并开始逐步实施

持续集成解决了京东到家的的问题：

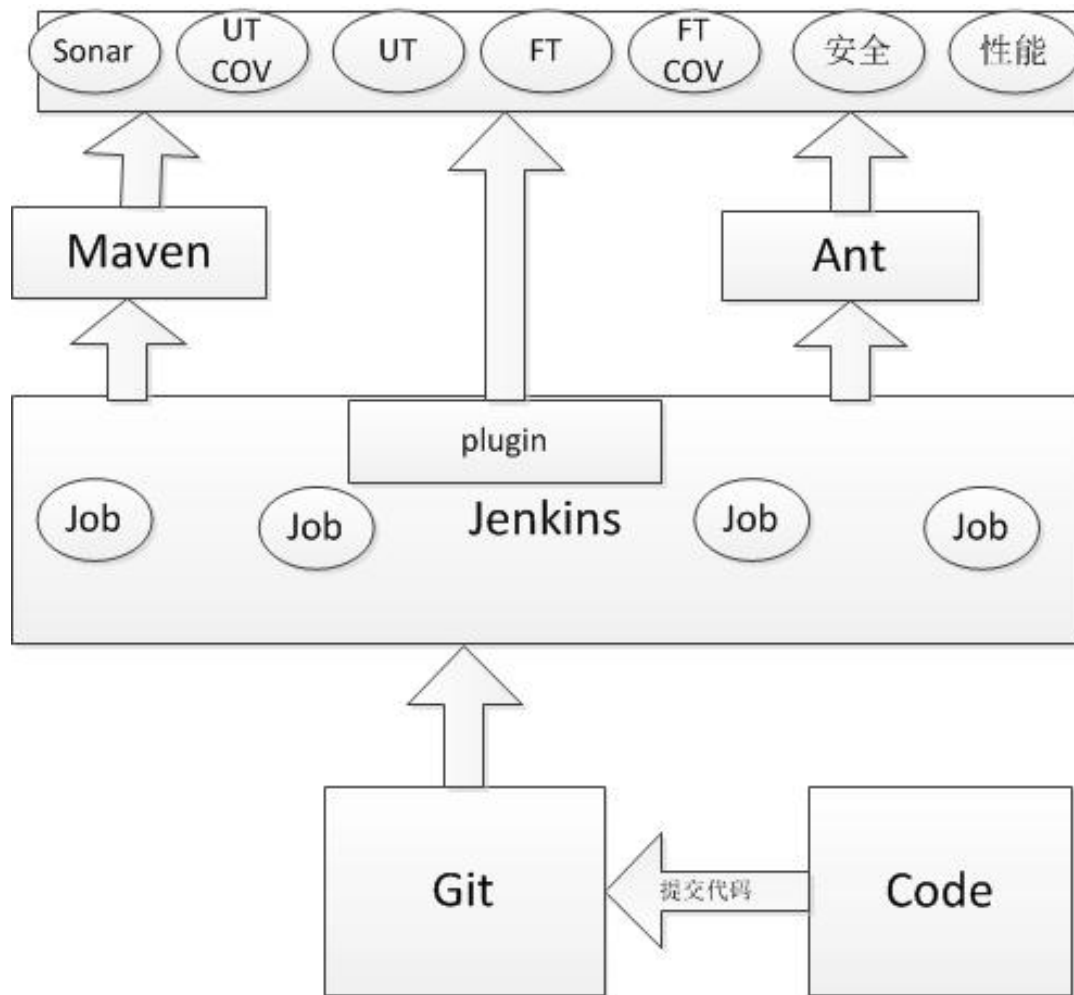
1. 提测质量差
2. 测试周期长
3. 重复性的工作多

● 如何推进持续集成

持续集成需要真个team都达成共识，按统一好的规则去做同一件事

1. 易做的影响力较大的先行
2. 先用一个项目做试点
3. 用数据去说服领导和同事

● 持续集成基本架构



● 京东到家持续集成的实现

提交代码到
git

代码静态扫描
和单元测试&
覆盖率

测试环境自动
部署测试分支

UI+jsf+App
自动化测试

修复和分析失
败case

功能测试

功能测试代码
覆盖率

项目名称

代码静态扫描

单元测试&代码覆盖率

部署

自动化测试

自动化功能测试代码覆盖率

要求

修复阻塞和重要级别的bug

新增代码必须达到一定的覆盖率

所有系统都现在自动编译，打包，部署

覆盖主流程，不断添加测试用例

查看覆盖率报告，对于新增代码业务逻辑必须全覆盖

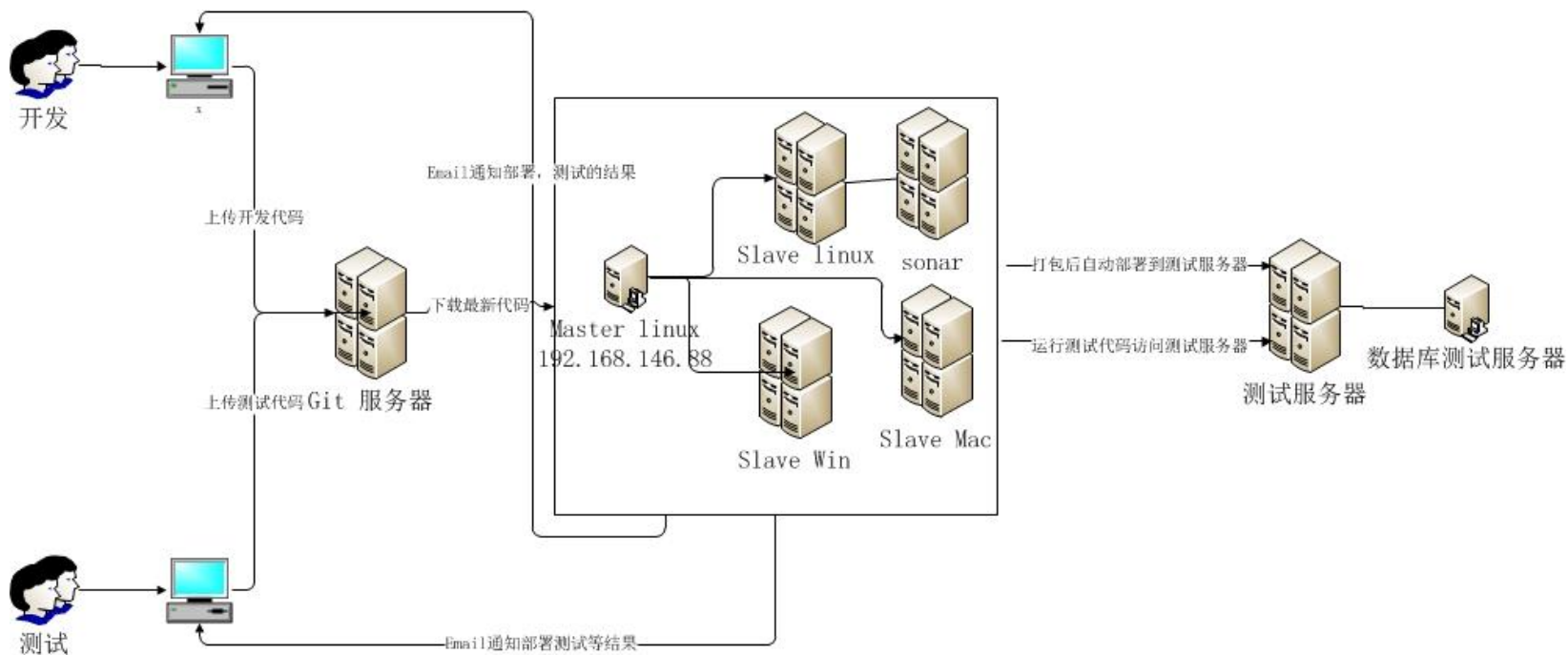
后续添加功能

自动扫描安全漏洞

性能测试自动化

code review

●持续集成服务器架构



- 持续集成概述
- **测试环境的演变**
- 代码静态扫描
- 代码覆盖率收集
- 自动化测试

● 京东到家测试环境演变

第一阶段 手动编译，打包，部署。

测试：帮我打个包吧

开发：等一下

测试：（有那么麻烦吗？）

开发：（刚想好的代码有没思路了，又的保存现在的代码，切换到对应分支，编译打包。）

开发：发你包部署一下吧

半个小时过去了。。。。。。

测试：咋不对，是版本不对吗？帮我打上一个吗？（打个包都打不对）

开发：等我一下，我这块需要紧急处理一个线上的问题（不早说，我还的回复到以前代码）

2个小时过去了。。。。。。

测试：帮我打包了吗？

开发：刚才忙忘了，马上打

半天过去了。。。。。。

服务终于部署好了

● 京东到家测试环境演变

方案：引入jenkins自动化编译打包部署

第二阶段主要问题：服务为啥这么慢？服务为啥挂了？

原因：

1. 服务器硬盘打满了
2. 一台机器上部署的服务太多了
3. 服务停止服务

方案：

1. 定时清理硬盘空间
2. 监控服务状态，自动重启服务

第三阶段主要问题：分布式服务，服务之间调不通。(jsf服务，类似于淘宝dubbo服务)

原因：

1. 迭代速度快，例如：前端服务还在1.3，后端1.4的服务已经开始测试，修改接口调用字段
2. 开发在本地调试，随意修改本地配置文件。组名不对，导致调用不到服务。

方案：

1. 两套测试环境。一套对外提供服务，供他人测试调用。一套自己测试使用。
2. 保存本地配置，与线上配置比对，防止误改
3. 使用自动化测试，检查整个系统的健康状况，及时修复

- 持续集成概述
- 测试环境的演变
- **代码静态扫描**
- 代码覆盖率收集
- 自动化测试

● Sonar 代码质量管理体系-代码静态扫描

1. 配置sonar环境

2. 配置jenkins环境, 下载sonarqube插件, 并进行配置

SonarQube
SonarQube installations

Name	Sonar4.5.4
Disable	<input type="checkbox"/> Check to quickly disable SonarQube on all jobs.
高级...	
Delete SonarQube	
Name	Sonar4.5.4.165
Disable	<input type="checkbox"/> Check to quickly disable SonarQube on all jobs.
Server URL	<input type="text" value="http://192.168.195.165:9000"/> Default is http://localhost:9000
SonarQube account login	<input type="text" value="admin"/> SonarQube account used to perform analysis. Mandatory when anonymous access is disabled.
SonarQube account password	<input type="password" value="*****"/> SonarQube account used to perform analysis. Mandatory when anonymous access is disabled.
Database URL	<input type="text" value="jdbc:mysql://192.168.195.165:3306/sonar?useUnicode=true&characterEncoding=utf8&rewriteBatchedStatements=true"/> Do not set if default embedded database.
Database login	<input type="text" value="root"/> Default is sonar.
Database password	<input type="password" value="*****"/> Default is sonar.
Version of sonar-maven-plugin	<input type="text"/> If not specified, then sonar:sonar will be used.
Additional properties	<input type="text"/> Additional properties to be passed to the mvn executable (example : -Dsome.property=some.value)
Trigger Exclusions (only for jobs with SonarQube as a post build action)	
Skip if triggered by SCM Changes	<input type="checkbox"/>
Skip if triggered by the build of a dependency	<input type="checkbox"/>
Skip if environment variable is defined and set to true	<input type="checkbox"/>
Delete SonarQube	

3. 在jenkins代码静态扫描的job后, 配置运行sonar

构建后操作

SonarQube

SonarQube Installation

● Sonar 代码静态扫描实例分析

- Home
- 公共服务
- 订单中心
- 网关配置
- 运营平台
- 商家后台
- 权限系统
- 商家主数据
- 基础服务
- 促销系统
- 创新业务
- 订单交易
- 工具
- 依赖
- 比较

O2O-WORKER	
技术债务	问题
6小时 ▼	4 ▼
❗ 阻断	0
❗ 严重	0
❗ 主要	3
✅ 次要	0
ℹ️ 信息	1

O2O-CUSTOMER-TRADE-ASSISTANT	
技术债务	问题
2小时 ▼	6
❗ 阻断	0
❗ 严重	0
❗ 主要	6
✅ 次要	0
ℹ️ 信息	0

O2O-LOCATION	
技术债务	问题
3小时 ▼	5 ▼
❗ 阻断	0
❗ 严重	0
❗ 主要	5
✅ 次要	0
ℹ️ 信息	0

O2O-OCS-PROMISE-HELPER:POM:1.0.0-SNAPSHOT

O2O-WORKER

2016年10月27日

1.0.0.0-SNAPSHOT

● 阻断违规: 0 ● 严重违规: 0
● 主要违规: 3

O2O-CUSTOMER-TRADE-ASSISTANT

2016年11月1日

● 阻断违规: 0 ● 严重违规: 2
● 主要违规: 4

O2O-LOCATION

2016年10月27日

3.0.0-SNAPSHOT

● 阻断违规: 0 ● 严重违规: 0
● 主要违规: 5

● Sonar 代码静态扫描实例分析

时间修改

过滤

未解决问题

打开/重开问题

已解决问题

误判问题

严重性

- 严重
- 主要
-
-

规则

- Boxed value is unboxed and t... 2 批量修改
- Correctness - Call to equals()... 1

```
109     openPlatAddStationRequest.setActivityStationList(new ArrayList<OpenPlatActivityStation>());
110   }
111   if(openPlatAddStationRequest.getOutActivityStationList() != null) {
112     for (OpenPlatActivityStation openPlatActivityStation : openPlatAddStationRequest.getOutActivityStationList()) {
113       if (openPlatActivityStation.getStationNo() == null || "".equals(openPlatActivityStation.getStationNo())) {
```

Call to String.equals(Long) in com.jd.o2o.orderdiscount.facade.api.openplatform.impl.OpenPlatConvertOutToInner.convertOutStationNoToInnerNo(PublicResponse, OpenPlatAddStationR

评论 | 打开 | 确认 | 解决 | 误判 | 分配 [分配给我] | 计划 | 修改严重性 | 债务: 1h

规则 | 修改日志

Correctness - Call to equals() comparing different types

This method calls equals(Object) on two references of different class types with no common subclasses. Therefore, the objects being compared are unlikely to be members of the same class classes were not analyzed, or dynamic class loading can occur at runtime). According to the contract of equals(), objects of different classes should always compare as unequal; therefore, ac java.lang.Object.equals(Object), the result of this comparison will always be false at runtime.

findbugs:EC_UNRELATED_TYPES | Reliability > Architecture

●Sonar 代码静态扫描实例分析

Unnamed - com.jd.o2o.o2o-vender-settle-api-pom:1.0.0-SNAPSHOT / Unnamed - com.jd.o2o.o2o-vender-settle-api-war:1.0.0-SNAPSHOT

src/main/java/com/jd/o2o/api/vender/settle/service/impl/VenderBaseProfileServiceImpl.java

317 代码行数 | 1小时10分钟 2 技术债务 违规 | 0.0% 覆盖率 4.4% 重复行(%) | SCM

时间修改

过滤	严重性	规则
未解决问题	严重	1 Boxed value is unboxe... 1 批量修改
打开/重开问题	主要	1 Correctness - Possible ... 1
已解决问题		
误判问题		

```
386         if (CollectionUtils.isNotEmpty(listEmps)) {
387             VenderShopAuto venderSettleShop = list.get(0);
388             maps.put("shopname", venderSettleShop.getShopName().trim());
389         }
390         VenderSettleContract venderSettleContract = venderContractMapper.findCompanyFinInfoByBaseId(baseId.intValue());
```

Possible null pointer dereference of baseId in com.jd.o2o.api.vender.settle.service.impl.VenderBaseProfileServiceImpl.venderJump(String)

评论 | 打开 确认 解决 误判 | 分配 [分配给我] | 计划 | 修改严重性 | 债务: 1h

规则 修改日志

Correctness - Possible null pointer dereference

There is a branch of statement that, if executed, guarantees that a null value will be dereferenced, which would generate a `NullPointerException` when the code is executed. Of course, the problem might be that the branch or statement is infeasible and that the null pointer exception can't ever be executed, deciding that is beyond the ability of FindBugs.

findbugs:NP_NULL_ON_SOME_PATH | Reliability > Logic

- 持续集成概述
- 测试环境的演变
- 代码静态扫描
- **代码覆盖率收集**
- 自动化测试

●代码覆盖率收集

单元测试代码覆盖率环境搭建

```
sed -i 's%</plugins>%<plugin><groupId>org.codehaus.mojo</groupId>  
<artifactId>sonar-maven-plugin</artifactId><version>1.0</version>  
</plugin><plugin><groupId>org.apache.maven.plugins</groupId>  
<artifactId>maven-surefire-plugin</artifactId>  
<version>2.9</version><configuration><skip>>false</skip>  
</configuration></plugin><plugin><groupId>org.jacoco</groupId>  
<artifactId>jacoco-maven-plugin</artifactId>  
<version>0.7.4.201502262128</version><executions><execution><id>  
default-prepare-agent</id><goals><goal>prepare-agent</goal></goals>  
</execution><execution><id>default-report</id><phase>prepare-  
package</phase><goals><goal>report</goal></goals></execution>  
<execution><id>default-check</id><goals><goal>check</goal></goals>  
<configuration><rules><!-- implementation is needed only for Maven  
2 --><rule implementation="org.jacoco.maven.RuleConfiguration">  
<element>BUNDLE</element><limits><!-- implementation is needed  
only for Maven 2 --><limit  
implementation="org.jacoco.report.check.Limit"><counter>  
COMPLEXITY</counter><value>COVEREDRATIO</value>  
<minimum>0.00</minimum></limit></limits></rule></rules>  
</configuration></execution></executions></plugin></plugins>%'  
pom.xml
```

●代码覆盖率收集

单元测试代码覆盖率

2. jenkins运行mvn clean install命令,生成代码覆盖率报告

仪表盘
时间机器
问题
工具
组件
问题明细
CSV Export
设计
依赖
比较
sonarqube

覆盖率
23.8%

o2o-platform-promotion-ruleengine-domain	0.0%	src/main/java/com/jd/o2o/pre/service/orderdiscount/impl	42.5%	PushGrabCacheThread.java	84.2%
o2o-platform-promotion-ruleengine-api	0.0%	src/main/java/com/jd/o2o/pre/service/publish	49.2%		
o2o-platform-promotion-ruleengine-worker	0.0%	src/main/java/com/jd/o2o/pre/service/promoterule/impl	51.3%		
o2o-platform-promotion-ruleengine-web	0.0%	src/main/java/com/jd/o2o/pre/service/promoterule/factory	62.5%		
o2o-platform-promotion-ruleengine-dao	0.0%	src/main/java/com/jd/o2o/pre/service/mark/impl	76.1%		
o2o-platform-promotion-ruleengine-rpc	0.0%	src/main/java/com/jd/o2o/pre/service/thread	84.2%		

o2o-platform-promotion-ruleengine / o2o-platform-promotion-ruleengine-service
src/main/java/com/jd/o2o/pre/service/thread/PushGrabCacheThread.java

31 代码行数 0 技术债务 0 违规 84.2% 覆盖率 SCM

时间修改

单测			
覆盖率	84.2%		
代码覆盖率	93.3%	分支覆盖率	50.0%
代码行	15	分支覆盖率	4
未覆盖的代码	1	未覆盖分支	2

```
21  
22 public PushGrabCacheThread(Long gridId,GrabCacheKeyEnum keyEnum,String value,RedisService redisService,DynamicArgs dynamicArgs){  
23     this.gridId = gridId;  
24     this.keyEnum = keyEnum;  
25     this.value = value;  
26     this.redisService = redisService;  
27     this.dynamicArgs = dynamicArgs;  
28 }  
29  
30  
31  
32  
33 public void run() {  
34     String key= keyEnum.getGrabCacheKey(gridId.toString());//拼装缓存key  
35     boolean isUnLock = True;  
36     //是否是锁  
37     if (dynamicArgs.isDownLevelRewriteGrabLock() == false){  
38         String lockKey = keyEnum.getGrabCacheLockKey(gridId.toString());  
39         isUnLock = redisService.set(lockKey, "lock",dynamicArgs.grabLockExistTime(),TimeUnit.SECONDS, false);  
40     }  
41     if (isUnLock){  
42         redisService.setEX(key,value,30L, TimeUnit.MINUTES);  
43     }  
44 }  
45 }
```

● 代码覆盖率收集

功能测试代码覆盖率环境搭建

1. 启动脚本中插入监听参数

```

| JAVA_OPTS="$JAVA_OPTS -
| javaagent:/export/servers/jacoco/jacocoagent.jar=includes=*,output=tcpserver,port=
| 8893,address=192.168.200.92"

```

2. dump出监听文件

```

<jacoco:dump address="10.199.145.117" reset="true" destfile="${basedir}/jacoco_10.199.145.117_1.exec"
port="10001" append="true"/>

```

3. 使用Ant脚本生成最终的html

[Back to maintest_13_report_functional_coverage](#) index

JaCoCo Report > o2o function coverage

o2o function coverage

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
*com.jd.o2o.promotion.planning.service.impl	19%	13%	604	714	1,829	2,296	104	184	4	27		
*com.jd.o2o.promotion.planning.web.controller	0%	0%	469	469	1,883	1,883	128	128	14	14		
*com.jd.o2o.promotion.planning.facade.yunying	15%	10%	238	284	789	927	46	85	0	11		
*com.jd.o2o.promotion.planning.domain.common.utils	5%	3%	271	287	662	700	104	118	9	13		
*com.jd.o2o.promotion.planning.domain.common.enums	33%	26%	262	329	490	703	233	290	16	31		
*com.jd.o2o.promotion.planning.rpc.impl	12%	11%	123	159	499	567	42	77	0	14		
*com.jd.o2o.promotion.planning.domain.dto	1%	0%	532	535	791	797	531	534	11	13		
*com.jd.o2o.promotion.planning.domain.entity	31%	n/a	505	744	780	1,123	505	744	11	21		
*com.jd.o2o.promotion.planning.innerapi.yunying.dto	27%	n/a	468	680	795	1,028	468	680	10	29		
*com.jd.o2o.promotion.planning.facade.orderdiscount	7%	0%	83	94	361	391	26	37	2	6		
*com.jd.o2o.promotion.planning.innerapi.orderdiscount.dto	3%	n/a	451	462	676	689	451	462	5	13		

●代码覆盖率收集

功能测试代码覆盖率

4. 配置到Jenkins的持续集成环境

构建

Execute shell

Command

```
cp -rf /export/jenkins/functionalCoverage/* .  
dir=/export/.jenkins/jobs/workspace/maintest_13_report_service  
  
sh dump.sh 192.168.104.13 8893 jacoco1.exec  
sh dump.sh 192.168.104.13 8894 jacoco2.exec  
sh dump.sh 192.168.104.13 8895 jacoco3.exec  
  
sh merge.sh  
  
sh -x coverage.sh merge.exec $dir
```

See [the list of available environment variables](#)

删除

- 持续集成概述
- 测试环境的演变
- 代码静态扫描
- 代码覆盖率收集
- **自动化测试**

● 自动化测试

UI 自动化测试： Selenium+TestNG+Jenkins

1. 配置 maven pom.xml 文件

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>2.16</version>
  <configuration>
    <forkMode>once</forkMode>
    <!-- argLine>-Dfile.encoding=UTF-8</argLine -->
    <suiteXmlFiles>
      <suiteXmlFile>${suiteXmlFile}</suiteXmlFile>
    </suiteXmlFiles>
  </configuration>
</plugin>
```

2. 配置 Jenkins job

Build

Maven Version	maven3.1.1
Root POM	pom.xml
Goals and options	clean test -DsuiteXmlFile=src/test/java/AutoTest_SmokeTest.xml

●自动化测试

接口自动化测试：jmeter + ant+jenkins

```
<?xml version="1.0" encoding="UTF-8"?>

<project name="ant-jmeter-test" default="all" basedir=".">
  <tstamp>
    <format property="time" pattern="yyyyMMddhhmmss" />
  </tstamp>
  <!-- 需要改成自己本地的 jmeter 目录-->
  <property name="jmeter.home" value="D:\program\apache-jmeter-2.13" />
  <!-- jmeter生成jtl格式的结果报告的路径-->
  <property name="jmeter.result.jtl.dir" value="results" />
  <!-- jmeter生成html格式的结果报告的路径-->
  <property name="jmeter.result.html.dir" value="results" />
  <!-- 生成的报告的前缀-->
  <property name="ReportName" value="maincase" />
  <property name="jmeter.result.jtlName" value="${jmeter.result.jtl.dir}/${time}-${ReportName}.jtl" />
  <property name="jmeter.result.htmlName" value="${jmeter.result.html.dir}/${time}-${ReportName}.html" />
  <property name="jmeter.result.index" value="${jmeter.result.html.dir}/index.html" />

  <target name="all">
    <antcall target="clean" />
    <antcall target="test" />
    <antcall target="report" />
  </target>

  <target name="clean">
    <delete>
      <fileset dir="${jmeter.result.jtl.dir}" includes="*.jtl"/>
    </delete>
  </target>
</project>
```

● 自动化测试

接口自动化测试：jmeter + ant+jenkins

```
<target name="test">
  <taskdef name="jmeter" classname="org.programmerplanet.ant.taskdefs.jmeter.JMeterTask" />
  <jmeter jmeterhome="${jmeter.home}" resultlog="${jmeter.result.jtlName}">
    <!-- 声明要运行的脚本， "*.jmx"指包含此目录下的所有jmeter脚本-->
    <testplans dir="jmeter" includes="*.jmx" />
  </jmeter>
</target>

<target name="report">
  <xslt in="${jmeter.result.jtlName}"
    out="${jmeter.result.htmlName}"
    style="${jmeter.home}/extras/jmeter-results-detail-report_21.xsl" />

  <delete file="${jmeter.result.index}"/>
  <copy tofile="${jmeter.result.index}" file="${jmeter.result.htmlName}"/>

  <!-- 因为上面生成报告的时候，不会将相关的图片也一起拷贝至目标目录，所以，需要手动拷贝 -->
  <copy todir="${jmeter.result.html.dir}">
    <fileset dir="${jmeter.home}/extras">
      <include name="collapse.png" />
      <include name="expand.png" />
    </fileset>
  </copy>
</target>
</project>
```



● 自动化测试

APP自动化测试

1. 为什么选择appium

支持android/ios

支持现有框架

支持多种语言

跨应用

不依赖于源代码

2. app自动化测试在持续集成中遇到的问题

1. 网络不稳定失败率高

引入重试监听，重试3次

2. 不容易定位case失败的问题

加入截图和log日志

3. 运行速度慢

android启动多个服务，分布运行

●自动化测试

同一台电脑可以同时启动多个appium service，在多台Android手机上运行不同的case

```
command.addArgument("/k");
//command1.addArgument("/c");
command.addArgument("start");
command.addArgument("appium");
//command.addArgument("dir");
//command.addArgument("//program//Appium//node.exe");
//command.addArgument("//program//Appium//node_modules//appium//bin/");
command.addArgument("--address");
command.addArgument("127.0.0.1");
command.addArgument("--port");
command.addArgument("4723");
command.addArgument("--bootstrap-port");
command.addArgument("4726");
command.addArgument("--udid");
command.addArgument("ZX1C622688");

DesiredCapabilities dc = new DesiredCapabilities();
dc.setCapability("deviceName", "NX511J");//02ba37a5091ac7c1 MX4Pro 7608BL623325
dc.setCapability("platformName", "Android");
dc.setCapability("appPackage", "com.jingdong.pdj");
dc.setCapability("appActivity", "pdj.start.StartActivity");
dc.setCapability("unicodeKeyboard", "True");
dc.setCapability("resetKeyboard", "True");
dc.setCapability("noReset", "True");
try {
    driver = new AndroidDriver<AndroidElement>(new URL("http://127.0.0.1:4723/wd/hub"),dc);
} catch (MalformedURLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

Q&A