

Jenkins+ANT+jmeter 接口自动化测试

一、环境准备：

1 Jdk1.6 或 以上 :

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

命令行输入：java -version，出现如下提示说明安装成功

```
F:\apache-jmeter-2.13\work>java -version
java version "1.8.0_31"
Java(TM) SE Runtime Environment (build 1.8.0_31-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.31-b07, mixed mode)
F:\apache-jmeter-2.13\work>
```

2、ANT 下载：<http://ant.apache.org/bindownload.cgi>

ant 环境变量：

ANT_HOME: F:\apache-ant-1.9.4

Path: %ANT_HOME%\bin;

CLASSPATH: %ANT_HOME%\lib

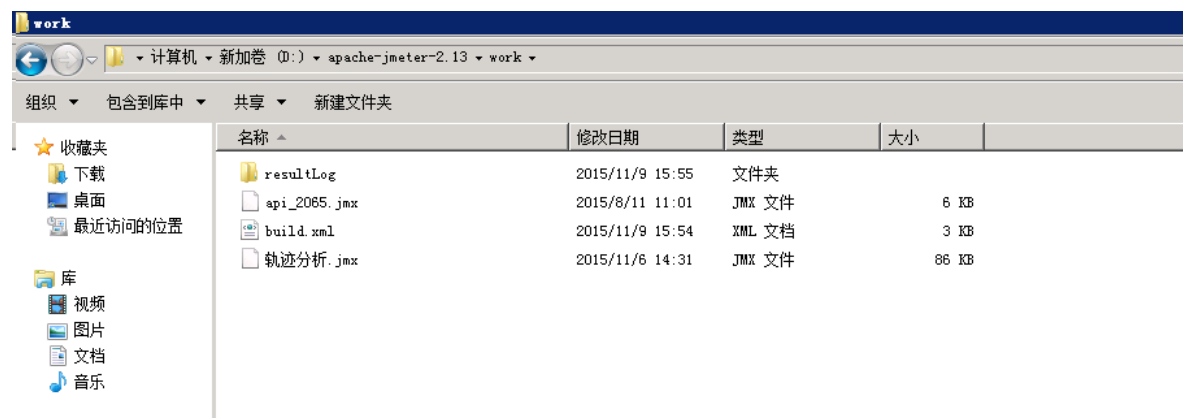
命令行输入：ant -v，出现如下提示说明安装成功

```
C:\Users\Administrator>ant -v
Apache Ant(TM) version 1.9.4 compiled on April 29 2014
Trying the default build file: build.xml
Buildfile: build.xml does not exist!
Build failed
```

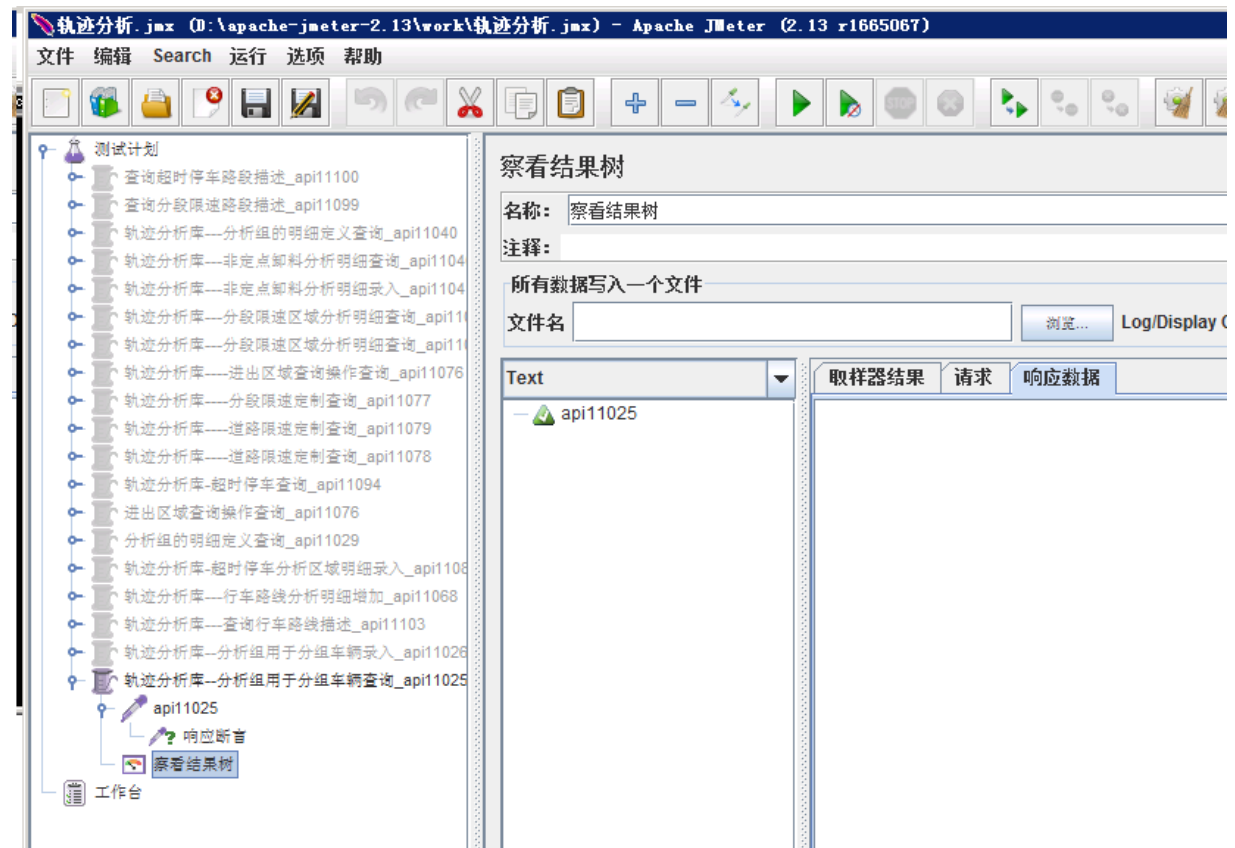
3、Jmeter 下载: http://jmeter.apache.org/download_jmeter.cgi

二、Jmeter 脚本:

1、脚本目录: D:\apache-jmeter-2.13\work



2、脚本内容



三、build.xml 代码:

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

```
<project name="ant-jmeter-test" default="all" basedir=".">
```

```
<tstamp>
```

```
<format property="time" pattern="yyyyMMddhhmm" />
```

```
</tstamp>
```

```
<property name="encoding" value="UTF-8" />
```

```
<!-- 需要改成自己本地的 Jmeter 目录-->

<property name="jmeter.home" value="D:\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="D:\apache-jmeter-2.13\work\resultLog\jtl" />

<!-- jmeter 生成 html 格式的结果报告的路径-->

<property name="jmeter.result.html.dir" value="D:\apache-jmeter-2.13\work\resultLog\html" />

<!-- 生成的报告的前缀-->

<property name="ReportName" value="TestReport" />

<property name="jmeter.result.jtlName" value="${jmeter.result.jtl.dir}/${ReportName}${time}.jtl" />

<property name="jmeter.result.htmlName" value="${jmeter.result.html.dir}/${ReportName}${time}.html" />


<target name="all">

    <antcall target="test" />

    <antcall target="report" />

    <antcall target="mail" />

</target>


<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="D:\apache-jmeter-2.13\work" includes="*.jmx" />
```

```
<property name="jmeter.save.saveservice.output_format" value="xml"/>
```

```
</jmeter>
```

```
</target>
```

```
<target name="report">
```

```
<xslt in="${jmeter.result.jtlName}"
```

```
out="${jmeter.result.htmlName}"
```

```
style="${jmeter.home}/extras/jmeter-results-detail-report_21.xsl" />
```

```
<!-- 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>

<property name="mail_from" value="邮箱地址" />

<property name="password" value="邮箱登录密码" />

<!--mail_to: 发 送 列 表 , 多 个 之 间 逗 号 间 隔

<property name="mailhost" value="smtp.exmail.qq.com" /> -->

<property name="mail_to" value="接收邮件的邮箱地址" />

<property name="mailport" value="25" />

<property name="mailhost" value="smtp.qq.com" />

<target name="mail">

    <loadfile property="report" srcFile="${jmeter.result.htmlName}" encoding="
${encoding}" />

    <mail from="${mail_from}" tolist="${mail_to}" mailport="${mailport}" mailho
st="${mailhost}" user="${mail_from}" password="${password}" subject="TestRepo
rt-${ant.project.name}" messagemimetype="text/html">

        <message>

            本轮构建的结果如下 :

            ${report}

        </message>

    </mail>

```

```
<delete file="${jmeter.result.htmlName}" />

</target>

</project>
```

四、运行脚本：

- 运行Jmeter脚本
- 将 jmeter的extras目录中ant-jmeter-1.1.1.jar包拷贝至ant安装目录下的lib目录中
- 修改Jmeter的bin目录下jmeter.properties文件的配置：
jmeter.save.saveservice.output_format=xml
- 发邮件需要把（activation.jar， commons-email-1.2.jar， mail.jar）这几个jar包放到ant的lib下面
- cmd进入脚本目录： D:\apache-jmeter-2.13\work
- 输入： **ant** 或 **ant -buildfile build.xml**， **ant run**(run为build.xml中的task名)， 执行结果：
- D:\apache-jmeter-2.13\work>ant
- Buildfile: D:\apache-jmeter-2.13\work\build.xml
- 测试报告目录： D:\apache-jmeter-2.13\work\resultLog\html

， 执行结果：

```
D:\apache-jmeter-2.13\work>ant
Buildfile: D:\apache-jmeter-2.13\work\build.xml
```

```
all:
```

```
test:
```

```
[jmeter] Executing test plan: D:\apache-jmeter-2.13\work\api_2065.jmx ==> D:\
apache-jmeter-2.13\work\resultLog\jtl\TestReport201511090354.jtl
[jmeter] Creating summariser <summary>
[jmeter] Created the tree successfully using D:\apache-jmeter-2.13\work\api_2
065.jmx
[jmeter] Starting the test @ Mon Nov 09 15:54:53 CST 2015 (1447055693166)
[jmeter] Waiting for possible shutdown message on port 4445
[jmeter] summary =    1 in   22s =   0.0/s Avg: 21118 Min: 21118 Max: 211
```

18 Err: 1 (100.00%)
[jmeter] Tidying up ... @ Mon Nov 09 15:55:15 CST 2015 (1447055715021)
[jmeter] ... end of run
[jmeter] Executing test plan: D:\apache-jmeter-2.13\work\轨迹分析.jmx ==> D:\apache-jmeter-2.13\work\resultLog\jtl\TestReport201511090354.jtl
[jmeter] Creating summariser <summary>
[jmeter] Created the tree successfully using D:\apache-jmeter-2.13\work\轨迹分析.jmx
[jmeter] Starting the test @ Mon Nov 09 15:55:16 CST 2015 (1447055716113)
[jmeter] Waiting for possible shutdown message on port 4445
[jmeter] summary = 1 in 0.4s = 2.8/s Avg: 105 Min: 105 Max: 1
05 Err: 0 (0.00%)
[jmeter] Tidying up ... @ Mon Nov 09 15:55:16 CST 2015 (1447055716566)
[jmeter] ... end of run

report:

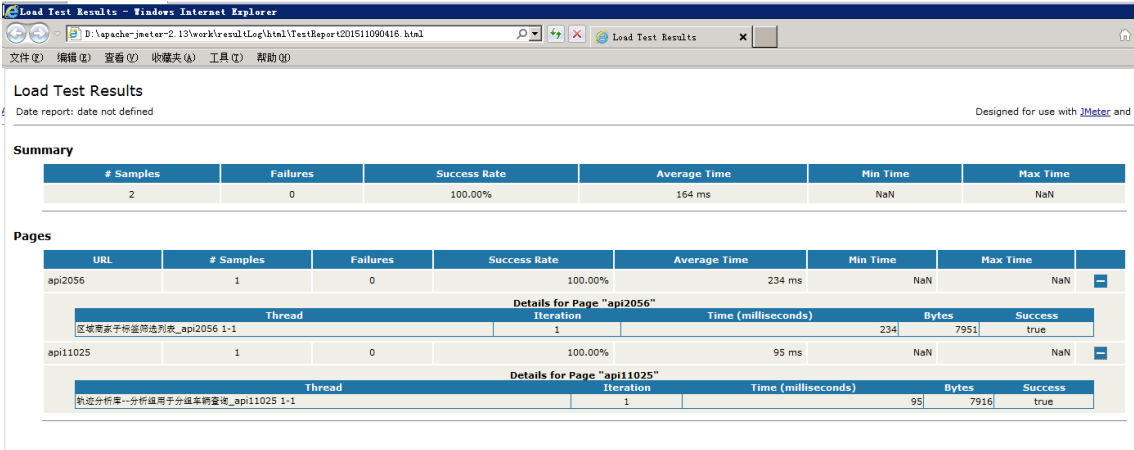
[xslt] Processing D:\apache-jmeter-2.13\work\resultLog\jtl\TestReport201511090354.jtl to D:\apache-jmeter-2.13\work\resultLog\html\TestReport201511090354.html
[copy] Copying 2 files to D:\apache-jmeter-2.13\work\resultLog\html

[xslt] Loading stylesheet D:\apache-jmeter-2.13\extras\jmeter-results-detail-report_21.xsl
[copy] Copying 2 files to D:\apache-jmeter-2.13\work\resultLog\html

BUILD SUCCESSFUL
Total time: 26 seconds

3 、 测 试 报 告 目 录 :

D:\apache-jmeter-2.13\work\resultLog\html



# Samples	Failures	Success Rate	Average Time	Min Time	Max Time
2	0	100.00%	164 ms	NaN	NaN

URL	# Samples	Failures	Success Rate	Average Time	Min Time	Max Time
api2056	1	0	100.00%	234 ms	NaN	NaN

Thread	Iteration	Time (milliseconds)	Bytes	Success
区域商家字标鉴图列表_api2056 1-1	1	234	7951	true

Thread	Iteration	Time (milliseconds)	Bytes	Success
轨迹分析---分析用于分组率调查_api11025 1-1	1	95	7916	true

四：最后，说一个我在使用过程中碰到一个问题，上面 build.xml 文件第 29 行有一句必须要加上：`<property name="jmeter.save.saveservice.output_format" value="xml"/>`，否则会报如下的错误：



report:

```
[xslt] Processing
D:\Tools\apache-jmeter-2.13\demo\report\jtl\TestReport201509210923.jtl to
D:\Tools\apache-jmeter-2.13\demo\report\html\TestReport201509210923.html
[xslt] Loading stylesheet
D:\Tools\apache-jmeter-2.13\extras\jmeter-results-detail-report_21.xsl
[xslt] : Error! 前言中不允许有内容。
[xslt] : Error! com.sun.org.apache.xml.internal.utils.WrappedRuntimeException:
前言中不允许有内容。
[xslt] Failed to process
D:\Tools\apache-jmeter-2.13\demo\report\jtl\TestReport201509210923.jtl
```



因为不加上这一句，生成的.jtl 文件是文本文件不是 xml 文件，使用 xsl 去转换.jtl 文件时就会报错。

五、部署到持续集成平台

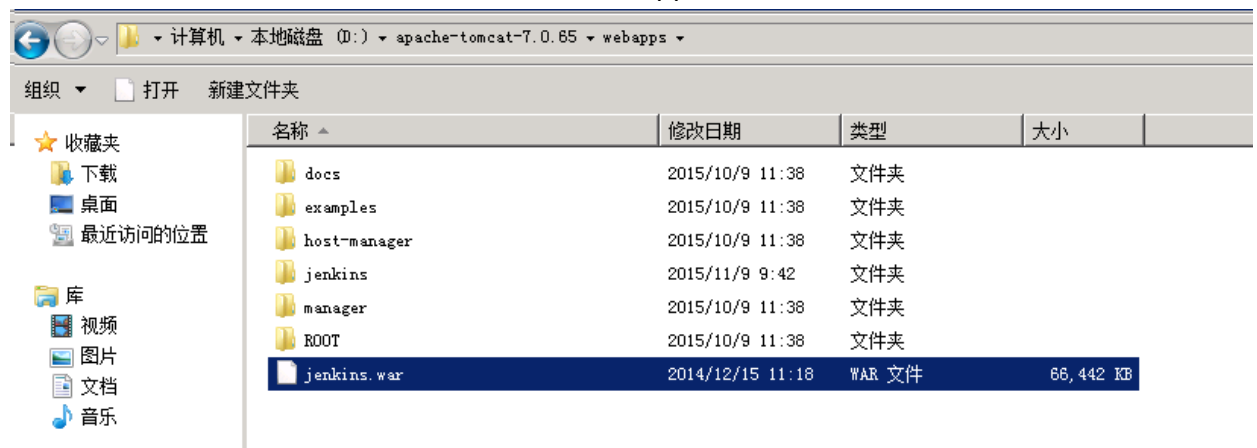
部署到持续集成平台可以实现脚本的定时运行，这是接口测试的核心。

这里我们选用了 jenkins,,jenkins 是一个强大的持续集成系统，使用起来也很简单。

使用步骤如下：

1、安装 jenkins

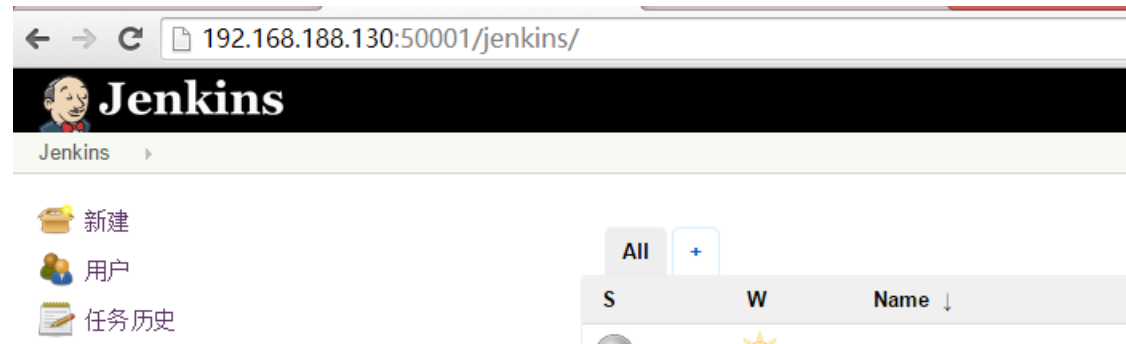
把 Jenkins 的 war 包复制到 tomcat 的 webapps 下面



2、启动 tomcat，访问 URL：http://ip:端口/jenkins

注意：请将 jenkins 安装到一个空间比较大的系统盘中。因为 jenkins 运行起来，生成的文件比较占空间。

出现如下界面说明 jenkins 部署成功



3、使用 jenkins

1)注册 jenkins 用户

第一次进入 jenkins 界面需要设置全局安全才能够注册用户



☒ 启用安全

JNLP节点代理的TCP端口 ☐ 指定端口： ☒ 随机选取 ☐ 禁用

Disable remember me ☐

访问控制

安全域

☒ Jenkins专有用户数据库

☒ 允许用户注册

☐ LDAP

☐ Servlet容器代理

☐ Unix用户/组数据库

授权策略

☒ 任何用户可以做任何事(没有任何限制)

☐ ...

点击应用按钮，保存按钮。
保存过后，就可以注册 jenkins 用户了

enkins/signup

查找 登录 | 注册

Sign up

用户名:

密码:

确认密码:

全名:

电子邮件地址:

Sign up

2) 新建一个项目

Jenkins

Jenkins + Add +

新建

任务历史

系统设置

Credentials

构建队列

队列中没有构建任务

构建执行状态

#	状态
1	空闲
2	空闲

Item名称

构建一个自由风格的软件项目

这是Jenkins的主要功能。Jenkins将会结合任何SCM和任何构建系统来构建你的项目。

构建一个maven2/3项目

构建一个maven2/3项目。Jenkins利用你的POM文件,这样可以大大减轻构建配置。

构建一个多配置项目

适用于多配置项目,例如多环境测试,平台指定构建,等等。

克隆一个外部的任务

这个类型的任务允许你记录运行在外部Jenkins的任务。任务甚至运行在远程机器上, 供参考。这个文档会验证配置。

复制已有的Item

复制已有的任务名称:

失败

项目名称

自动化构建

描述

[Escaped HTML] [预览](#)

☒ 丢弃旧的构建

策略

Log Rotation

保持构建的天数

1

如果非空，构建记录将保存此天数

保持构建的最大个数

1

如果非空，最多此数目的构建记录将被保存

☐ 参数化构建过程

☐ 关闭构建 (重新开启构建前不允许进行新的构建)

☐ 在必要的时候并发构建

高级项目选项

源码管理

☒ None

☐ CVS

☐ CVS Projectset

☐ Subversion

保存

应用

旧的构建保存了你在一次构建中的所有临时文件，如果构建没有保留的必要，就勾选丢弃旧的构建，同时设置保持构建天数和保持构建的最大个数两个参数。

注意：三个设置必须同时设置，否则无效

4、 设置定时运行间隔，这里，设置间隔时间为每天 8 点运行一次

Subversion

构建触发器

☐ 触发远程构建 (例如, 使用脚本)

☐ Build after other projects are built

☒ Build periodically

日程表

⚠ Spread load evenly by using 'H 8 * * *' rather than '0 8 * * *'

☐ Poll SCM

构建

Invoke Ant

Targets

Build File

Properties

Java Options

第一个参数代表的是分钟 minute，取值 0~59；

第二个参数代表的是小时 hour，取值 0~23；

第三个参数代表的是天 day，取值 1~31；

第四个参数代表的是月 month，取值 1~12；

最后一个参数代表的是星期 week，取值 0~7，0 和 7 都是表示星期天 特殊格式: * 指所有有效的值

M-N 指定值的范围 M-N/ X 或*/ X 是指以 X 为步长，指定的区域或整个有效范围区间的值 A, B, ..., Z

列举多个值 eg.H/15 * * * * 每十五分钟执行一次 H(0-29)/10 * * * * 每小时的 0-29 分内每隔十分钟执行一次

H 9-16/2 * * 1-5 每个工作日的 9-16 点内每隔两小时执行一次 H H 1,15 1-11 * 1-11 月的每月 1 号和 15 号各执行一次

JENKINS 新建项目流程

通过设置 invoke ant，就可以调用 ant，执行打包过程。在这里，也就是执行生成测试报告的步骤

- 查看报告

Jenkins_test

发给 liudefang, test, chenhai, wangpeisen

2015-12-23 20:00 详细信息

本构建项的结果如下

Load Test Results

Date report: date not defined

Designed for use with JMeter and Ant.

Summary

# Samples	Failures	Success Rate	Average Time	Min Time	Max Time
37	4	89.19%	79 ms	NaN	NaN

Pages

URL	# Samples	Failures	Success Rate	Average Time	Min Time	Max Time	
发送重置密码验证码	1	0	100.00%	460 ms	NaN	NaN	
获取注册验证码	1	0	100.00%	32 ms	NaN	NaN	
注册_输入已经注册的手机号码	1	0	100.00%	13 ms	NaN	NaN	
手机号码登录_手机号码错误	1	0	100.00%	31 ms	NaN	NaN	
手机号码登录_密码错误	1	0	100.00%	92 ms	NaN	NaN	

Failure Detail