

平台简介

一个完整的接口自动化测试平台需要支持接口的自动执行，自动生成测试报告，以及持续集成。Jmeter 支持接口的测试，Ant 支持自动构建，而 Jenkins 支持持续集成，所以三者组合在一起可以构成一个功能完善的接口自动化测试平台。

环境准备

环境依赖

- JDK 环境配置
- Jmeter 安装
- Ant 安装环境变量配置
- Jenkins 安装

Ant 简介

Apache Ant,是一个将软件编译、测试、部署等步骤联系在一起加以自动化的一个工具，大多用于 Java 环境中的软件开发。

下载安装

下载地址：<https://ant.apache.org/bindownload.cgi>

下载之后解压到任意文件路径，我这里是放到 C 盘根目录。

环境变量配置

- **ANT_HOME** C:\apache-ant-1.10.5
- **Path** : %ANT_HOME%\bin
- **ClassPath** %ANT_HOME%\lib

配置检测

输入如下命令即可检查是否安装成功

```
C:\Users\Shuqing>ant -version  
Apache Ant(TM) version 1.10.5 compiled on July 10 2018
```

注意：如果提示 ‘ant 不是内部命令也不是外部命令’ 说明环境变量可能配置错误。

Jenkins 简介

Jenkins 是一个开源软件项目，是基于 Java 开发的一种持续集成工具，用于监控持续重复的工作，旨在提供一个开放易用的软件平台，使软件的持续集成变成可能。

下载与安装

下载地址：<https://jenkins.io/download/>

下载后安装到指定的路径即可，默认启动页面为 localhost:8080,如果 8080 端口被占用无法打开，可以进入到 jenkins 安装目录，找到 jenkins.xml 配置文件打开，修改如下代码的端口号即可。

```
<arguments>-Xrs -Xmx256m -Dhudson.lifecycle=udson.lifecycle.WindowsServiceLifecycle -jar  
"%BASE%\jenkins.war" --httpPort=8080 --webroot="%BASE%\war"</arguments>
```



平台搭建

依赖文件配置

- 首先在 Jmeter 目录下面新建一个文件夹 loadTest (文件夹名称不要使用下划线, 空格字符), 并将 Jemter 测试脚本放置到该文件夹中。
- 将 Jmeter extras 文件中的 ant-jmeter-1.1.1.jar 放到 Ant 中的 lib 文件夹中
- 将 Jmeter extras 文件中的 jmeter-results-detail-report_21.xsl , build.xml、collapse.png、expand.png 放到 ant 目录中的 bin 目录下面。

build.xml 配置

在 Ant 的 bin 目录中打开 build.xml 文件找到以下内容

```
<property name="testpath" value="${user.dir}"/>
<property name="jmeter.home" value="${basedir}/../"/>
<property name="report.title" value="Load Test Results"/>

<!-- Name of test (without .jmx) -->
<property name="test" value="Test"/>
```

参数说明

- testpath 测试计划，这里用于存放测试脚本、测试生成的文件、测试报告
- jmeter.home Jmeter 目录路径
- report.title 测试报告的标题
- test jmeter 测试脚本的名称（无需后缀jmx）

这里根据自己的环境修改为

```
<property name="testpath" value="C:\apache-jmeter-4.0\loadTest"/>
```



```
<property name="jmeter.home" value="C:\apache-jmeter-4.0"/>
<property name="report.title" value="Httpbin API Test Report"/>

<!-- Name of test (without .jmx) -->
<property name="test" value="httpbin_test"/>
```

Ant 构建

执行如下命令进行构建

```
ant -buildfile C:\apache-ant-1.10.5\bin\build.xml
```

结果如下：

```
C:\Users\Shuqing>ant -buildfile C:\apache-ant-1.10.5\bin\build.xml
Buildfile: C:\apache-ant-1.10.5\bin\build.xml

run:
[echo] funcMode = false
[delete] Deleting: C:\apache-jmeter-4.0\loadTest\httpbin_test.html
[jmeter] Executing test plan: C:\apache-jmeter-4.0\loadTest\httpbin_test.jmx ==> C:\apache-jmeter-4.0\loadTest\httpbin_test.jtl
[jmeter] Creating summariser <summary>
[jmeter] Created the tree successfully using C:\apache-jmeter-4.0\loadTest\httpbin_test.jmx
[jmeter] Starting the test @ Tue Jul 17 09:50:30 CST 2018 (1531792230572)
[jmeter] Waiting for possible Shutdown/StopTestNow/Heapdump message on port 4445
[jmeter] summary + 1 in 00:00:01 = 0.8/s Avg: 871 Min: 871 Max: 871 Err: 0 (0.00%) Active: 1 Started: 1 Finished: 0
[jmeter] summary + 6 in 00:00:03 = 2.3/s Avg: 436 Min: 240 Max: 890 Err: 0 (0.00%) Active: 0 Started: 1 Finished: 1
[jmeter] summary = 7 in 00:00:04 = 1.8/s Avg: 498 Min: 240 Max: 890 Err: 0 (0.00%)
[jmeter] Tidying up ... @ Tue Jul 17 09:50:34 CST 2018 (1531792234934)
[jmeter] ... end of run

_message_xalan:

xslt-report:
[xslt] Processing C:\apache-jmeter-4.0\loadTest\httpbin_test.jtl to C:\apache-jmeter-4.0\loadTest\httpbin_test.html
[xslt] Loading stylesheet C:\apache-ant-1.10.5\bin\jmeter-results-detail-report_21.xsl

verify-images:

copy-images:

report:
[echo] Report generated at 2018/07/17 09:50

all:

BUILD SUCCESSFUL
Total time: 7 seconds

C:\Users\Shuqing>
```

进入到 loadTest 文件夹可以看到生成如下文件：



名称	修改日期	类型	大小
collapse.png	2018/7/17 9:54	PNG 文件	1 KB
expand.png	2018/7/17 9:54	PNG 文件	1 KB
httpbin_test.html	2018/7/17 10:00	HTML 文件	19 KB
httpbin_test.jmx	2018/7/16 17:38	JMX 文件	34 KB
httpbin_test.jtl	2018/7/17 10:00	JTL 文件	17 KB

打开 html 测试报告 httpbin_test.html 报告内容如下：

Httpbin API Test Results

Date report: 2018/07/17 10:00

Designed for use with [Jmeter](#) and [Ant](#).

Summary

# Samples	Failures	Success Rate	Average Time	Min Time	Max Time
70	0	100.00%	556 ms	240 ms	1729 ms

Pages

URL	# Samples	Failures	Success Rate	Average Time	Min Time	Max Time	
http-get	10	0	100.00%	769 ms	608 ms	1204 ms	+
get-num	10	0	100.00%	276 ms	249 ms	324 ms	+
http-post-51zxw	10	0	100.00%	282 ms	245 ms	334 ms	+
set-cookie	10	0	100.00%	574 ms	486 ms	921 ms	+
get-cookie	10	0	100.00%	441 ms	240 ms	1706 ms	+
basic-auth	10	0	100.00%	471 ms	288 ms	599 ms	+
digest-auth	10	0	100.00%	1077 ms	890 ms	1729 ms	+

报告优化

Jmeter 默认生成报告不是很详细，因此我们需要进行优化。这里我们使用新的报告模板：

jmeter-results-detail-report_30.xml 默认的报告模板是 jmeter-results-detail-report_21

打开 build.xml 将 21 改为 30

```
<condition property="style_version" value="_30">
```

补充删除 .jtl 的代码，如果不删除每次迭代的记录都会显示在报告里面，显得很冗余。

```
<target name="run">
    <echo>funcMode = ${funcMode}</echo>
    <delete file="${testpath}/${test}.html"/>
    <delete file="${testpath}/${test}.jtl"/>
</target>
```

进入 Jmeter 的 bin 目录，找到 jmeter.properties 文件打开。然后将以下代码取消注释状态，并改为 true

```
jmeter.save.saveservice.data_type=true
```



```

jmeter.save.saveservice.label=true
jmeter.save.saveservice.response_code=true
# response_data is not currently supported for CSV output
jmeter.save.saveservice.response_data=true
# Save ResponseData for failed samples
jmeter.save.saveservice.response_data.on_error=true
jmeter.save.saveservice.response_message=true
jmeter.save.saveservice.successful=true
jmeter.save.saveservice.thread_name=true
jmeter.save.saveservice.time=true
jmeter.save.saveservice.subresults=true
jmeter.save.saveservice.assertions=true
jmeter.save.saveservice.latency=true
# Only available with HttpClient4
jmeter.save.saveservice.connect_time=true
jmeter.save.saveservice.samplerData=true
jmeter.save.saveservice.responseHeaders=true
jmeter.save.saveservice.requestHeaders=true
jmeter.save.saveservice.encoding=true
jmeter.save.saveservice.bytes=true
# Only available with HttpClient4
jmeter.save.saveservice.sent_bytes=true
jmeter.save.saveservice.url=true
jmeter.save.saveservice.filename=true
jmeter.save.saveservice.hostname=true
jmeter.save.saveservice.thread_counts=true
jmeter.save.saveservice.sample_count=true
jmeter.save.saveservice.idle_time=true

```

最后执行即可生成最新的报告：样式如下，可以清晰看到每一个请求发送，响应内容。

Thread: 线程组 1-1	Sampler
http-get	Thread Name : 线程组 1-1
get-num	Timestamp : Tue Jul 17 2018 14:46:20 GMT+0800 (中国标准时间)
http-post-51zxw	Time : 611 ms
get-cookie	Latency : 610 ms
basic-auth	Bytes : 473
digest-auth	Sample Count : 1
	Error Count : 0
	Response Code : 200
	Response Message: OK
	Assertion
	1. 响应断言
	Failure : false
	Error : false
	Failure Message :
	Request
	Method/Url : GET http://httpbin.org/get?username=51zxw&num=45
	Query String :
	Cookies :
	Request Headers : Connection: keep-alive
	Host: httpbin.org
	User-Agent: Apache-HttpClient/4.5.5 (Java/1.8.0_06)
	Response
	Response Headers: HTTP/1.1 200 OK
	Connection: keep-alive
	Server: gunicorn/19.8.1
	Date: Tue, 17 Jul 2018 06:46:20 GMT
	Content-Type: application/json
	Content-Length: 227
	Access-Control-Allow-Origin: *
	Access-Control-Allow-Credentials: true
	Via: 1.1 varnish
	Response Data : [{"args":{"num":"45","username":"51zxw"},"headers":{"Connection":"close","Host":"httpbin.org","User-Agent":"Apache-HttpClient/4.5.5 (Java/1.8.0_06)","origin":"110.52.4.225","url":"http://httpbin.org/get?username=51zxw&num=45"}}
	Response File :

注意

如果生成的报告为空，则说明 jmeter 脚本有错误，此时可以打开 jmeter 的 bin 目录中的 jmeter.log 日志文件，查看具体的报错信息。

集成到 Jenkins

在 Jenkins 新建一个任务 httpbin_Api_test

在构建选项中选择 Invoke Ant 然后在 Build File 输入 build.xml 配置文件路径。注意：不要输入到 Targets 里面去了，需要点击高级选项后才可以显示除 Build File



构建

Invoke Ant

Targets

Build File

C:\apache-ant-1.10.5\bin\build.xml

Properties

Java Options

增加构建步骤

执行之后可以看到控制台输出和 cmd 的控制台输出是一样的



- 返回到工程
- 状态集
- 变更记录
- 控制台输出
- 文本方式查看
- 编辑编译信息
- 删除本次生成
- 前一次构建

Executed Ant Targets

- run
- _message_xalan
- xslt-report
- verify-images
- copy-images
- report
- all

控制台输出

```
Started by user shuging
Building in workspace D:\Program Files (x86)\jenkins\workspace\httpbin_Api_test
[bin] $ cmd.exe /C "ant.bat -file build.xml %* exit %%ERRORLEVEL%%"
Buildfile: C:\apache-ant-1.10.5\bin\build.xml

run:
[echo] funcMode = false
[delete] Deleting: C:\apache-jmeter-4.0\loadTest\httpbin_test.html
[jmeter] Executing test plan: C:\apache-jmeter-4.0\loadTest\httpbin_test.jmx ==> C:\apache-jmeter-4.0\loadTest\httpbin_test.jtl
[jmeter] Creating summariser <summary>
[jmeter] Created the tree successfully using C:\apache-jmeter-4.0\loadTest\httpbin_test.jmx
[jmeter] Starting the test @ Tue Jul 17 09:28:35 CST 2018 (1531790915359)
[jmeter] Waiting for possible Shutdown/StopTestNow/Heapdump message on port 4445
[jmeter] summary = 7 in 00:00:04 = 1.8/s Avg: 483 Min: 262 Max: 1123 Err: 0 (0.00%)
[jmeter] Tidying up ... @ Tue Jul 17 09:28:39 CST 2018 (1531790919602)
[jmeter] ... end of run

_message_xalan:

xslt-report:
[xslt] Processing C:\apache-jmeter-4.0\loadTest\httpbin_test.jtl to C:\apache-jmeter-4.0\loadTest\httpbin_test.html
[xslt] Loading stylesheet C:\apache-ant-1.10.5\bin\jmeter-results-detail-report_21.xsl

verify-images:

copy-images:

report:
[echo] Report generated at 2018/07/17 09:28

all:

BUILD SUCCESSFUL
Total time: 7 seconds
Finished: SUCCESS
```

如果想定制构建则可以在**构建触发器**中选择 Build periodically 如果想每个工作日下班 18 时执行，则可以如下设置：

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

0 18 * * 1-5

⚠ Spread load evenly by using 'H 18 * * 1-5' rather than '0 18 * * 1-5'

Would last have run at 2018年7月16日 星期二 下午06时00分38秒 CST; would next run at 2018年7月17日 星期二 下午06时00分38秒 CST.

- ☐ GitHub hook trigger for GITScm polling
- ☐ Poll SCM

扩展资料：[Appium jenkins 自动化测试平台搭建](#)

邮件推送

参考 [python 邮件发送](#)





参考资料

- <https://baike.baidu.com/item/apache ant/1065741>
- <http://www.cnblogs.com/puresoul/p/5049433.html>
- <https://testerhome.com/topics/2580>
- <https://blog.csdn.net/yinjing8435/article/details/71159403>