

Jenkins+selenium+ant+svn+ testng+eclipse

持续集成框架手册

制作人：王绍坤

文档版本号：V1.0.1

目录

Jenkins+selenium+ant+svn+	1
testng+eclipse	1
持续集成框架手册	1
一、环境准备:	4
二、eclipse 创建 java 工程和 testng 类.....	9
三、Ant build.xml 代码:	13
四、teseng.xml 代码	16
五、jenkins 配置.....	17
六、项目构建.....	22
七、Jenlins 设置自动构建	23
八、Jenkins 生成 htmlreport 和发送邮件设置.....	24

文档修订记录

版本编号	变化状态	简要说明	变更人	变更日期
1.0	所有章节	创建	王绍坤	2016.06.20
1.0.1	Jenkins svn 配置	增加 jenkins 系统设置 svn 版本与 svnplugin 版本兼容截图说明	王绍坤	2016.06.23

一、环境准备：

JDK:

建议安装 JDK1.7

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

[环境变量配置不再多说](#)

命令行输入：java -version，出现如下提示说明安装成功

```
C:\Users\Administrator>java -version
java version "1.6.0_45"
Java(TM) SE Runtime Environment (build 1.6.0_45-b06)
Java HotSpot(TM) Client VM (build 20.45-b01, mixed mode, sharing)
```

Eclipse+Testng

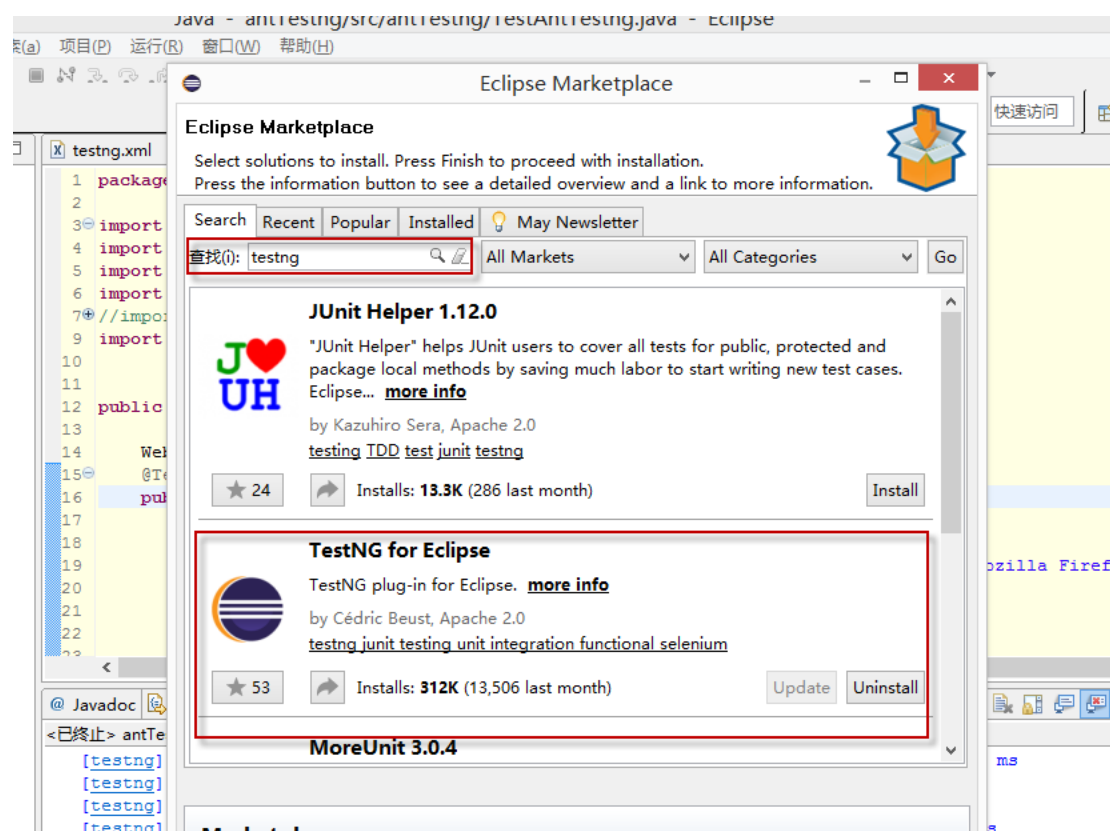
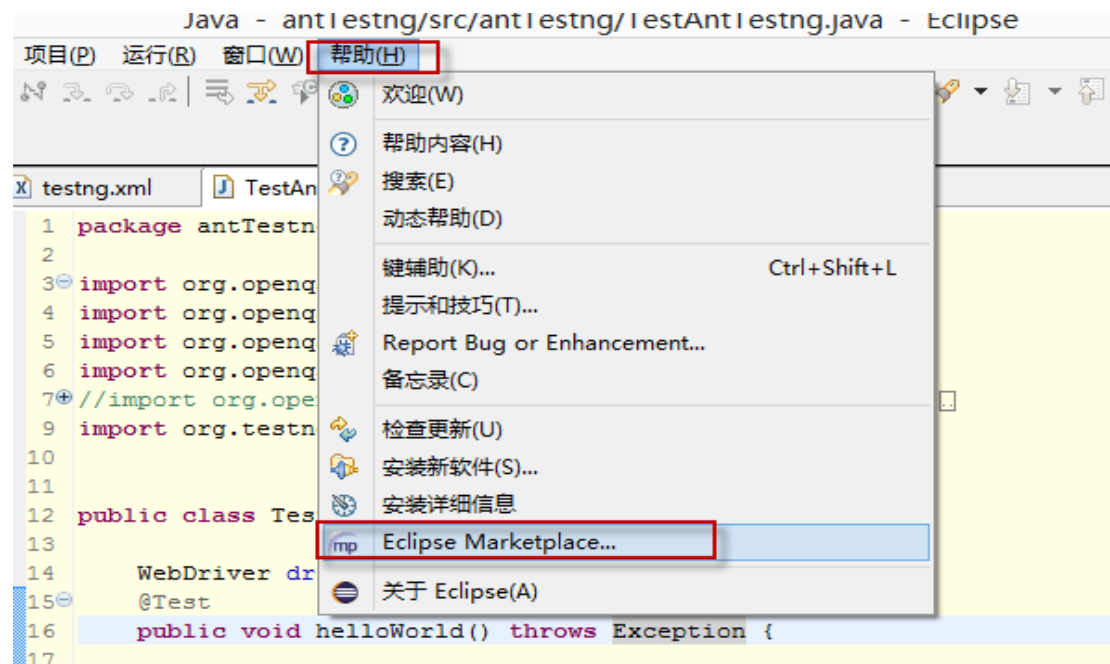
1.安装版本Eclipse IDE for Java Developers Version: Luna Service Release 1 (4.4.1)

2.testng Eclipse本身自带ant，Testng可以在线安装方法：

启动Eclipse，选择“Install New Software”，输入网址为“<http://beust.com/eclipse>”，然后单击“Add”。打开添加存储库对话框输入名称为“TestNG”，然后点击“OK”，具体方法可参

http://www.yiibai.com/selenium/selenium_test_ng.html

也可以在eclipse搜索安装：亲测成功



SeleniumIDE, seleniumRC, firefox

- 1 seleniumIDE 下载 selenium-ide-2.5.0.xpi，下载后直接拖入 firefox
- 2 firefox 版本要求为 Firefox_Setup_25.0.1_chs.exe，其他较高版本不兼容 selenium-ide-2.5.0，

安装 selenium 相关插件重启浏览器即可

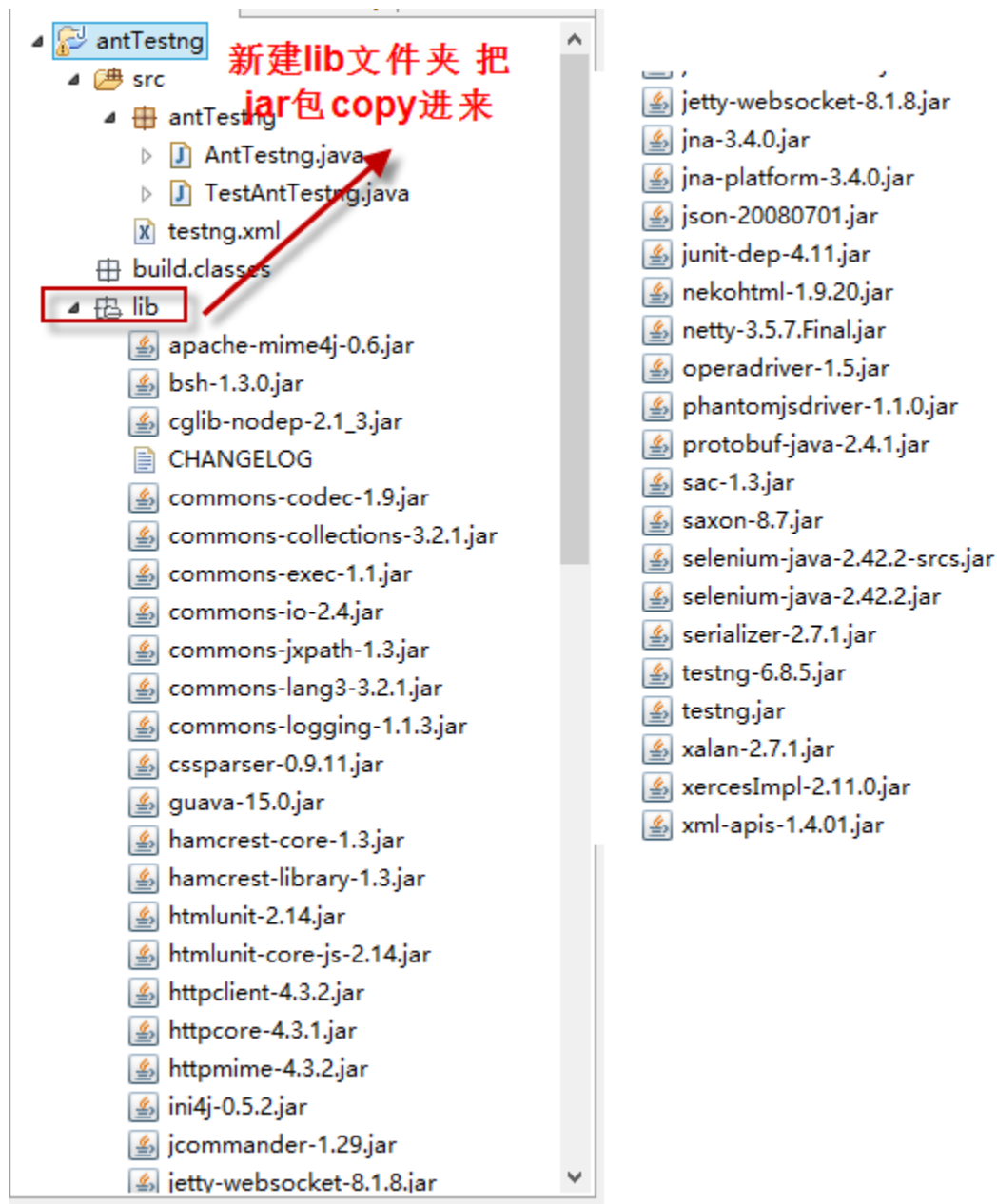


录制的代码可以直接 copy 到 eclipse 进行调优

3 下载 seleniumrc 相关 jar 包

selenium-java-2.42.2.jar selenium-server-standalone-2.7.0-patched.jar

selenium-remote-control-1.0.3.jar



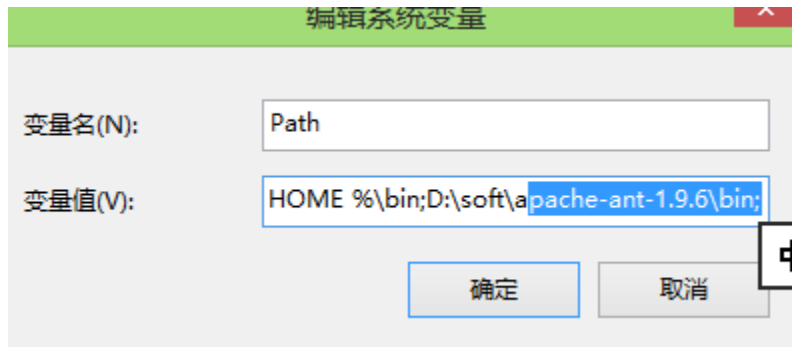
Ant:

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

1 配置环境变量

ANT_HOME = D:\soft\apache-ant-1.9.6 (本地安装位置)

Path = %ANT_HOME%\bin;%ANT_HOME%\lib;

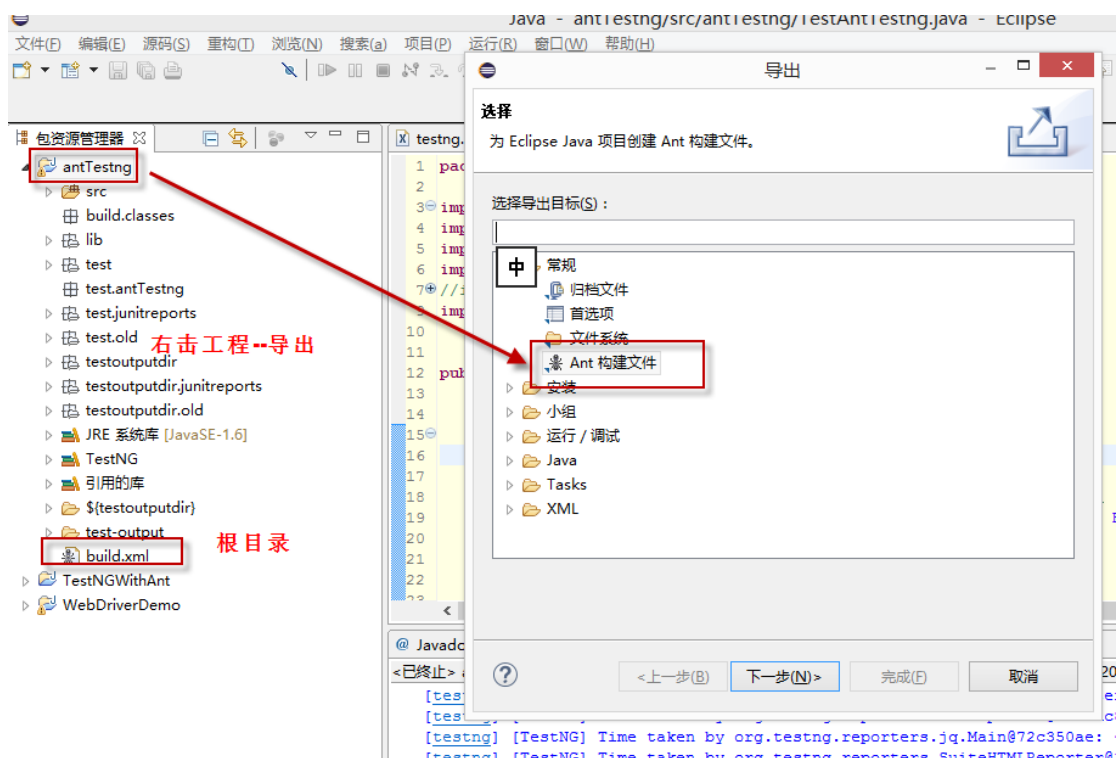


命令行输入: `ant -v` 或 `ant`, 出现如下提示说明安装成功

```
C:\Users\Administrator>ant -v
Apache Ant(TM) version 1.9.6 compiled on June 29 2015
Trying the default build file: build.xml
Buildfile: build.xml does not exist!
Build failed
```

2. eclipse 可以生成 build.xml

直接在所需编译工程右击—导出—ant 构建文件, 生成在根目录, 然后编译 testng 的 ant-build.xml code



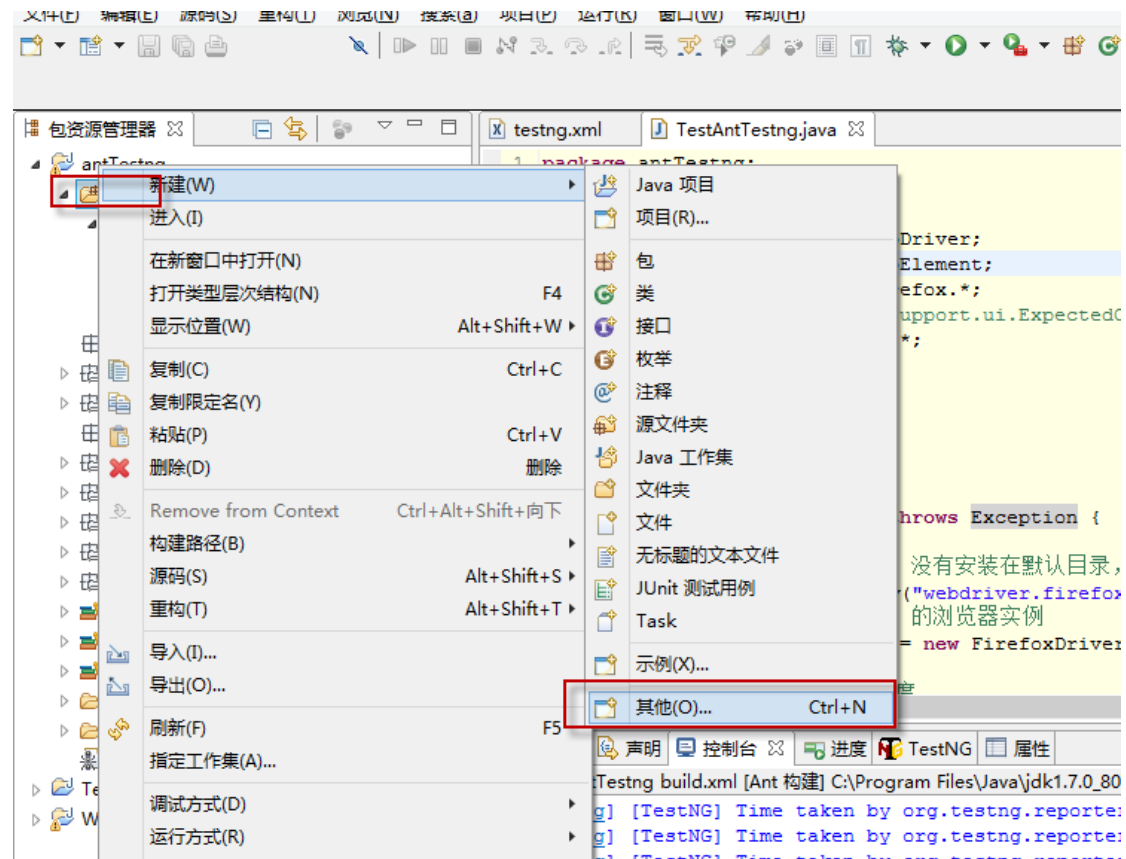
Jenkins:

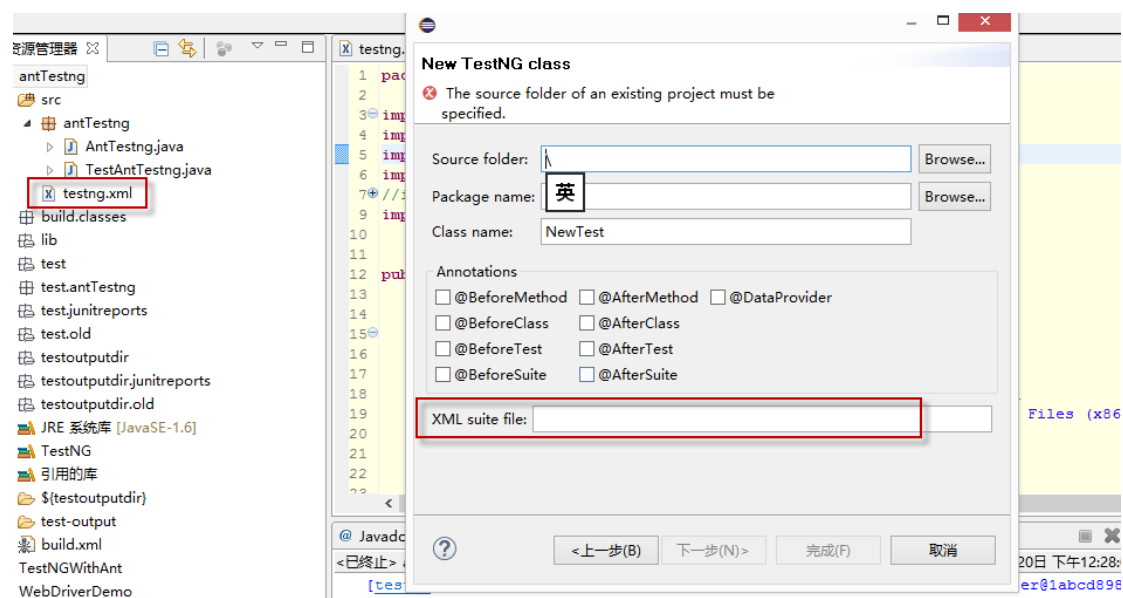
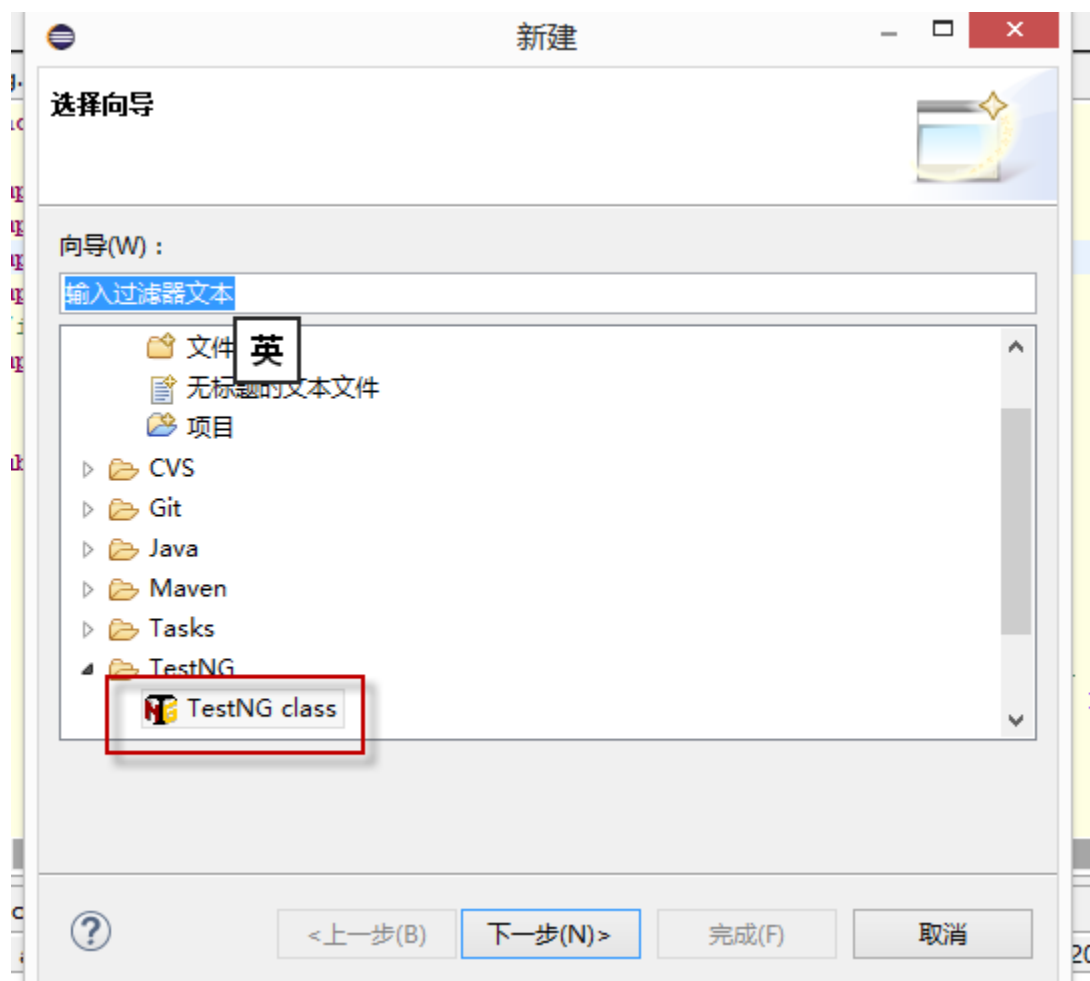
<http://jenkins-ci.org/>

可以在本地搭建 jenkins 集成环境, 方便调试。Jenkins 也可以安装到 tomcat 服务器

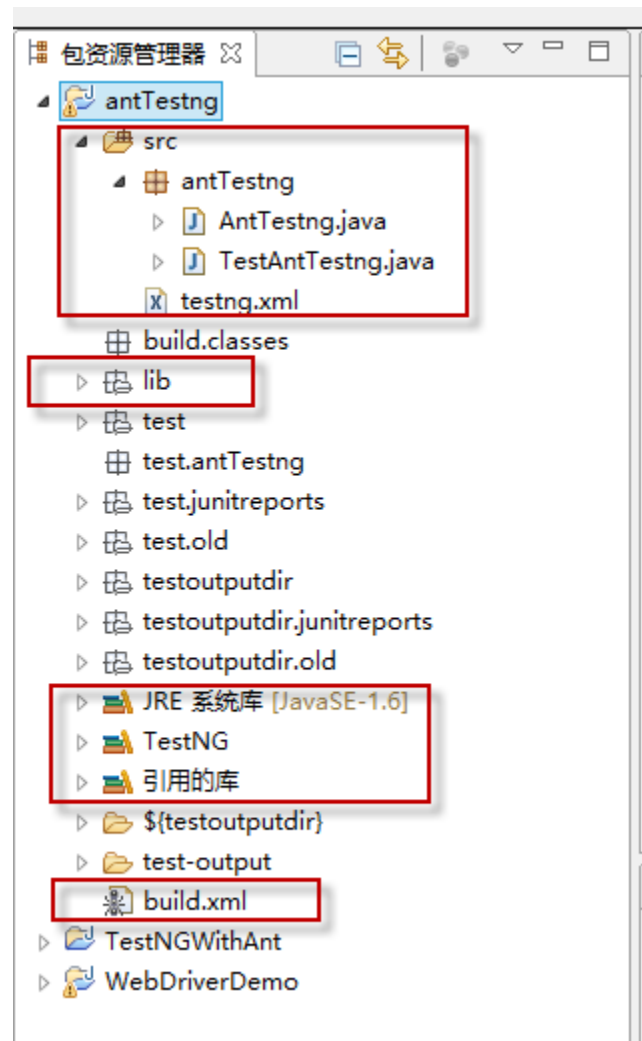
二、eclipse 创建 java 工程和 testng 类

右击 src 创建 testng 类和 testng.xml



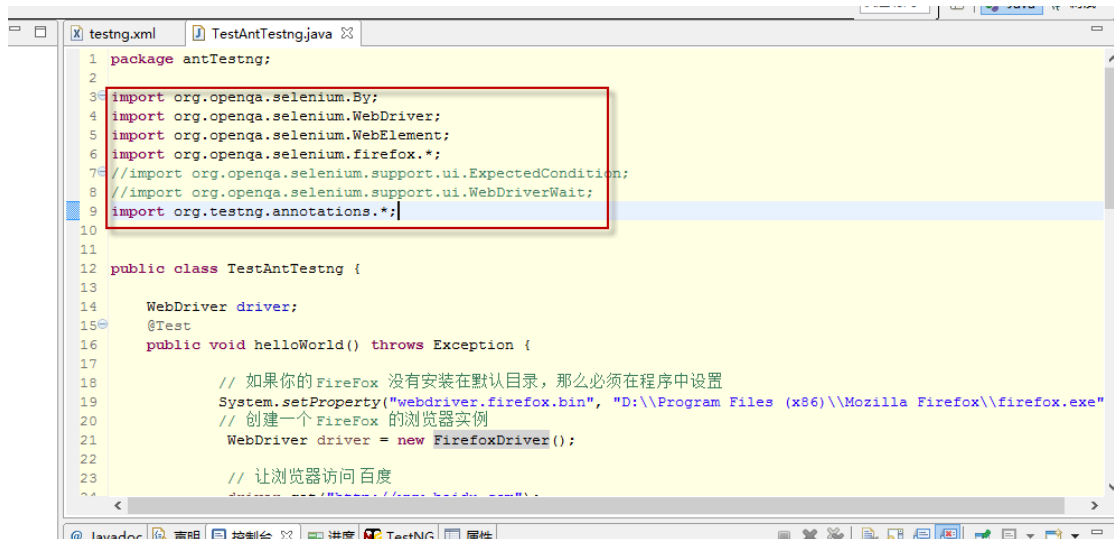


创建成功后引入所需要的 jar 包具体结构可以参考 demo:

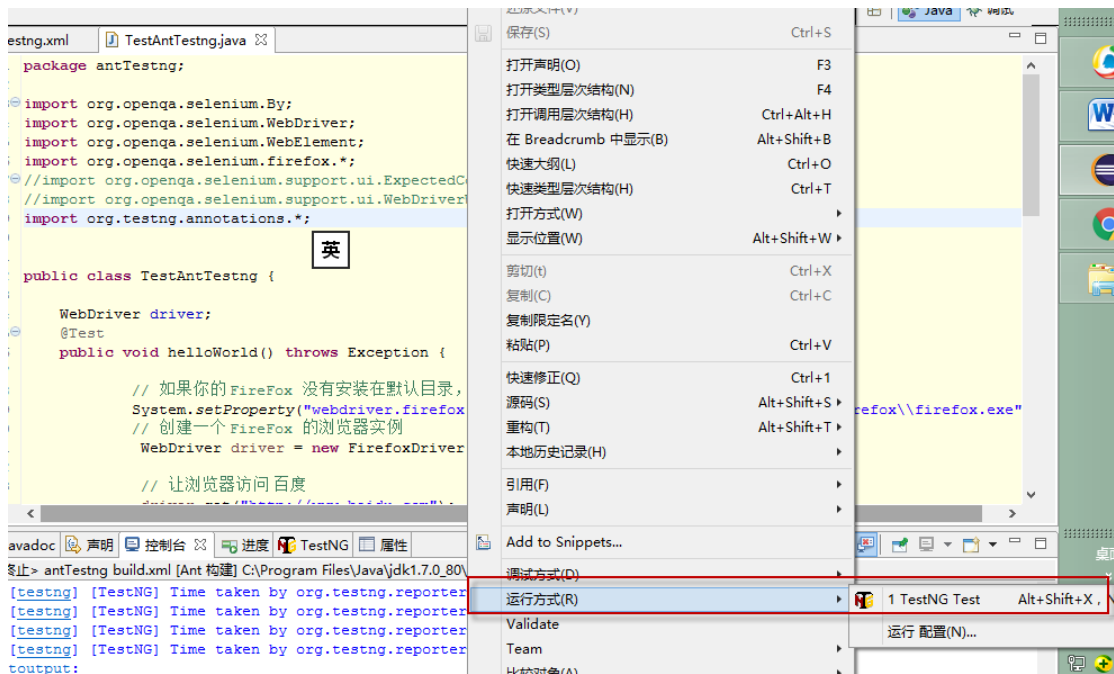


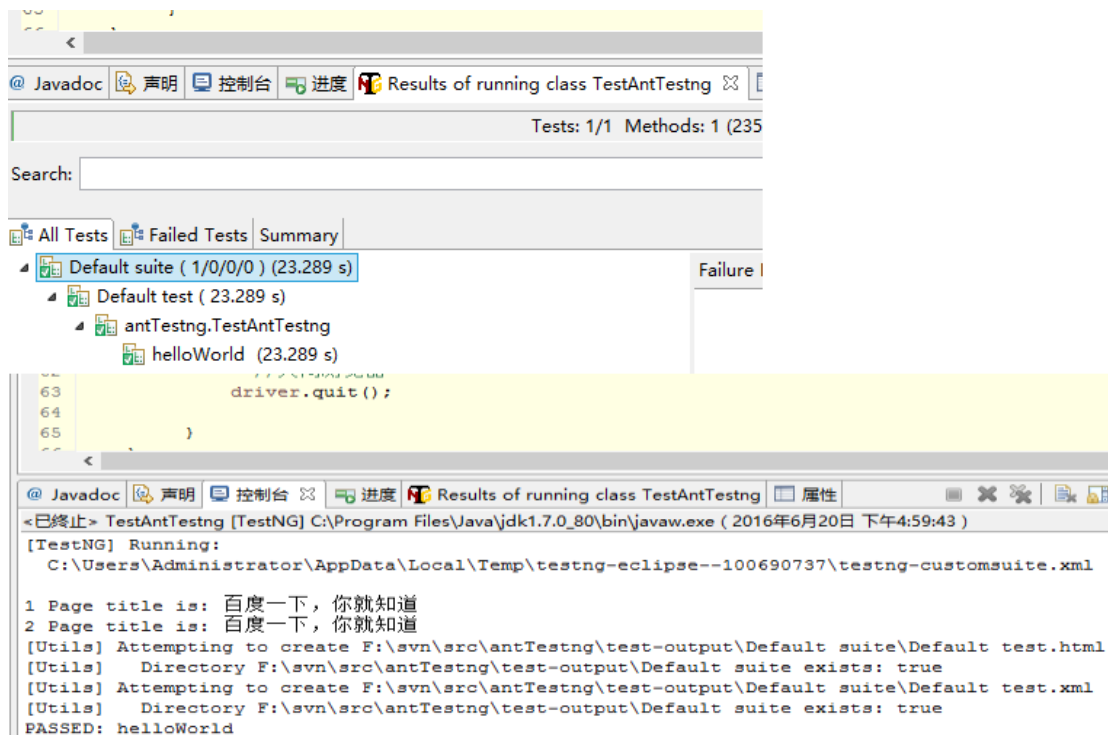
Selenium 脚本录制及调优

可以先使用 ide 录制脚本然后 eclipse 进行优化，并增加所需要的 case，



编写完成后可以使用 testng test 验证是否能够 build 成功





三、Ant build.xml code:

完整代码如下目前能在jenkins跑起来,生成报告只能在代码根目录testoutput查看.html,jenkins在线生成还在探索。。。

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<project name="TestNGTest" default="testoutput" basedir=".">

    <echo message="import lib" />
    <!-- Define <testng> task -->
    <taskdef name="testng" classname="org.testng.TestNGAntTask">

        <classpath>
            <pathelement location="lib/testng.jar"/>
        </classpath>

    </taskdef>
    <property name="testdir" location="test" />
    <property name="srcdir" location="src" />
    <property name="testoutputdir" location="testoutputdir" />
    <property name="libdir" location="lib" />
```

```
<property name="full-compile" value="true" />
<path id="classpath.base"/>
<path id="classpath.test">
  <fileset dir="${libdir}">
    <include name="**/*.jar" />
  </fileset>
  <pathelement location="${testdir}" />
  <pathelement location="${srcdir}" />
  <pathelement location="${testoutputdir}" />

  <path refid="classpath.base" />
</path>
<target name="clean" >
  <delete verbose="${full-compile}">
    <fileset dir="${testdir}" includes="**/*.class" />
  </delete>
</target>
<target name="compile" depends="clean">
  <javac srcdir="${srcdir}" destdir="${testdir}"
    verbose="${full-compile}" includeantruntime="on">
    <classpath refid="classpath.test"/>
  </javac>
</target>

<target name="test" depends="compile">
  <testng outputdir="${testoutputdir}"
classpathref="classpath.test">
    <!--此目录下的所有TestNG的XML文件，使用*号表示此目录下所有 -->
    <xmlfileset dir="${srcdir}" includes="testng.xml">
      </xmlfileset>
    </testng>

  </target>
  <!--TestNG执行完毕后，根据输出的testng-results.xml文件，按照
testng-results.xsl的设定生成html格式的测试报告 -->

  <target name="testoutput" depends="test">
    <xslt in="${testoutputdir}/testng-results.xml"
style="${testoutputdir}/testng-results.xsl"
out="${testoutputdir}/index1.html ">
    <param name="testNgXslt.outputDir"
expression="F:/wskworkspace/antTestng/test-output" />
    <classpath refid="classpath.test" />
  </xslt>
```

```

</target>
</project>

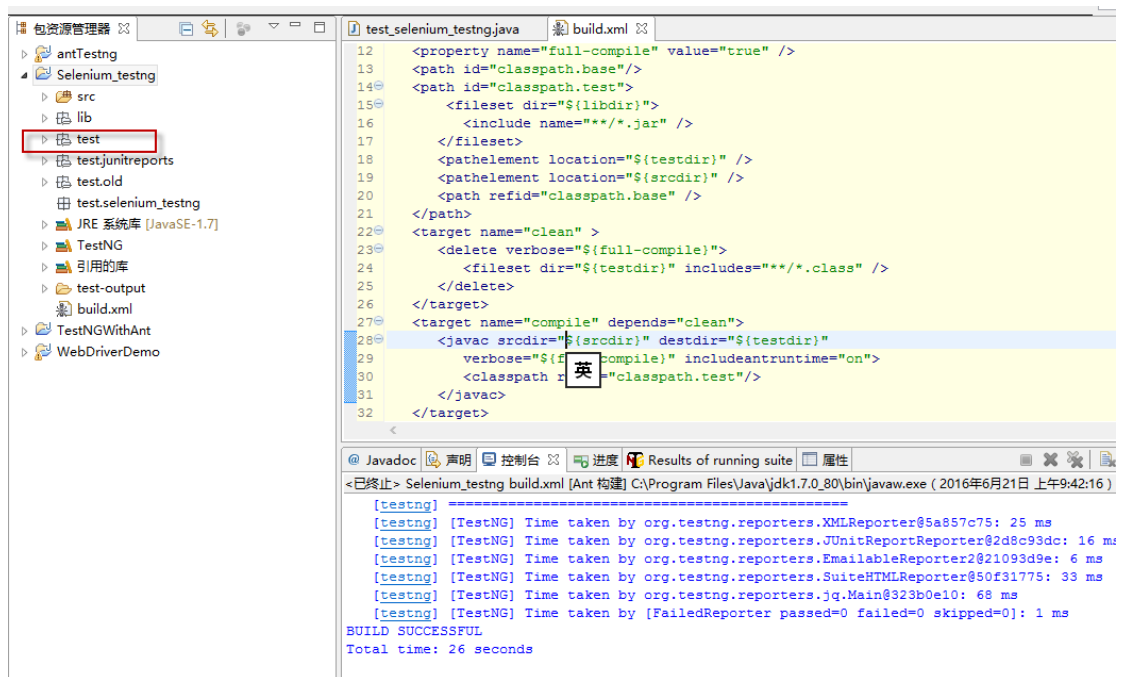
```

如下代码 **jenkins** 可以生成报告, 亲测成功

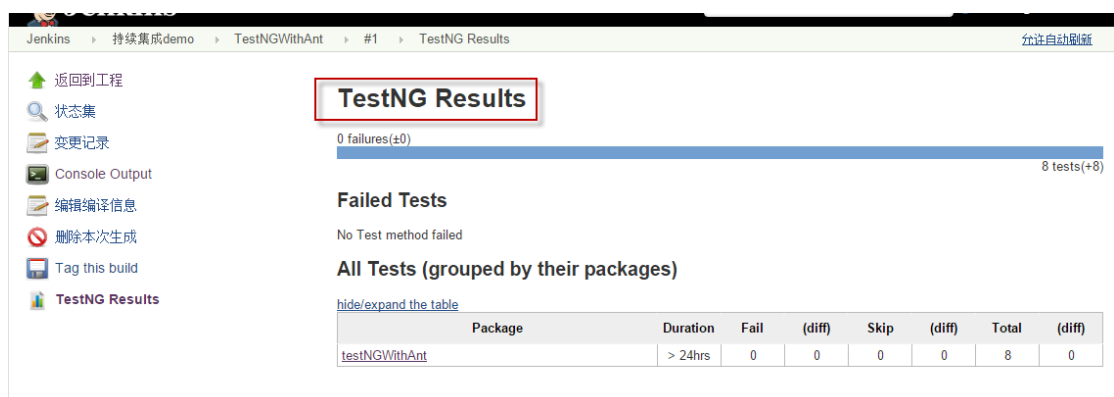
```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<project name="TestNGTest" default="test" basedir=".">
<!-- Define <testng> task -->
  <taskdef name="testng" classname="org.testng.TestNGAntTask">
    <classpath>
      <pathelement location="lib/testng.jar"/>
    </classpath>
  </taskdef>
  <property name="testdir" location="test" />      <!--需要在工程中创建test
文件夹否则eclipse编译会报错找不到test文件夹 -->
  <property name="srcdir" location="src" />
  <property name="libdir" location="lib" />
  <property name="full-compile" value="true" />
  <path id="classpath.base"/>
  <path id="classpath.test">
    <fileset dir="${libdir}">
      <include name="**/*.jar" />
    </fileset>
    <pathelement location="${testdir}" />
    <pathelement location="${srcdir}" />
    <path refid="classpath.base" />
  </path>
  <target name="clean" >
    <delete verbose="${full-compile}">
      <fileset dir="${testdir}" includes="**/*.class" />
    </delete>
  </target>
  <target name="compile" depends="clean">
    <javac srcdir="${srcdir}" destdir="${testdir}"
      verbose="${full-compile}" includeantruntime="on">
      <classpath refid="classpath.test"/>
    </javac>
  </target>
  <target name="test" depends="compile">
    <testng outputdir="${testdir}" classpathref="classpath.test">
      <xmlfileset dir="${srcdir}" includes="testng.xml"/>
    </testng>
  </target>
</project>

```



下图为 jenkins 生成 report



[XML](http://tool.oschina.net/codeformat/xml) 格式化工具 <http://tool.oschina.net/codeformat/xml>

四、teseng.xml code

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite guice-stage="DEVELOPMENT" name="Default suite">
  <test verbose="2" name="Default test">
    <classes>
      <class name="antTestng.TestAntTestng"/>
    </classes>
  </test> <!-- Default test -->
</suite> <!-- Default suite -->
```

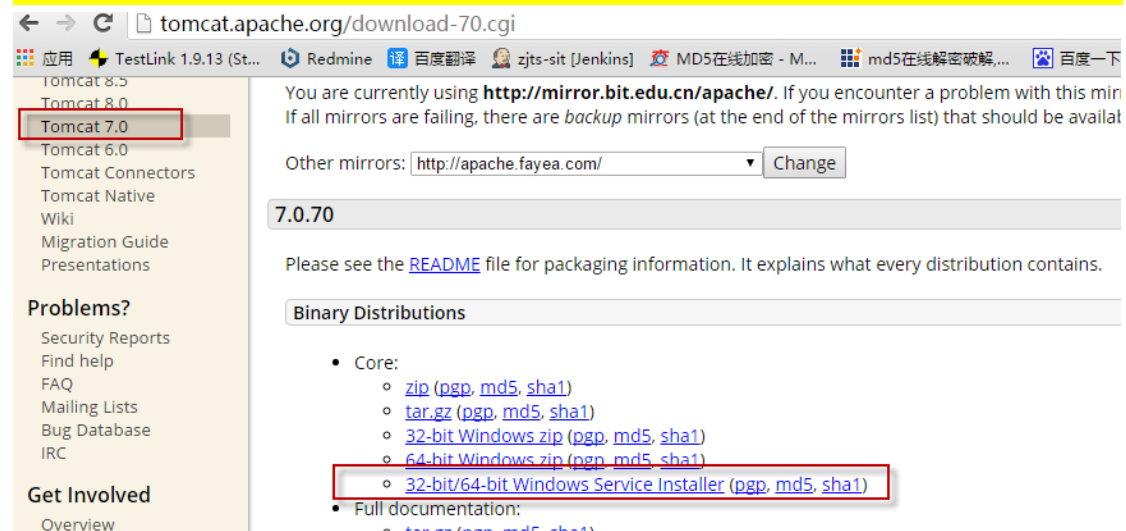

五、jenkins+tomcat 本地安装配置

1、tomcat 本地安装

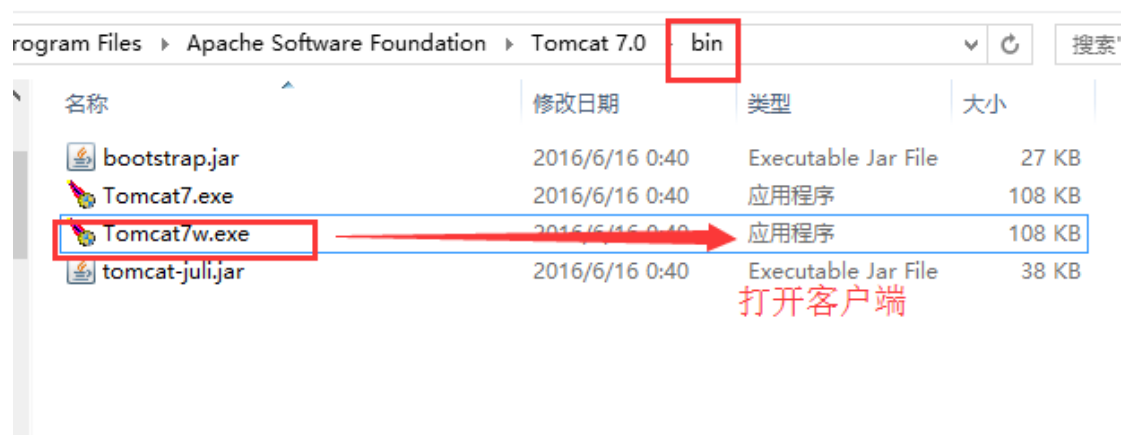
<http://tomcat.apache.org/download-70.cgi>

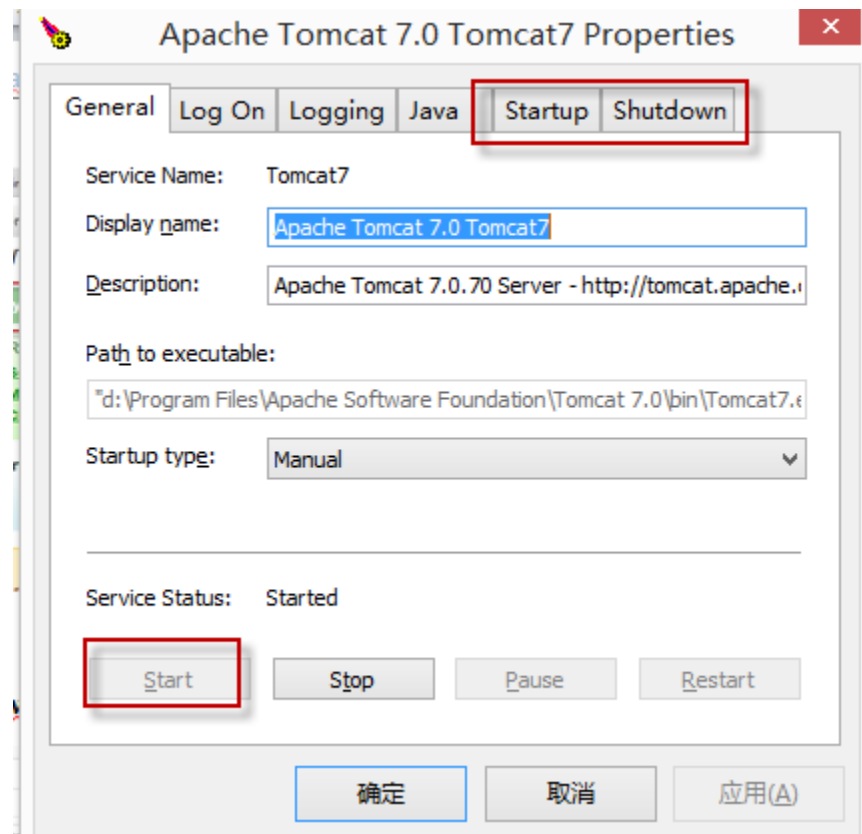
apache-tomcat-7.0.70.exe

安 装 步 骤 可 自 行 百 度

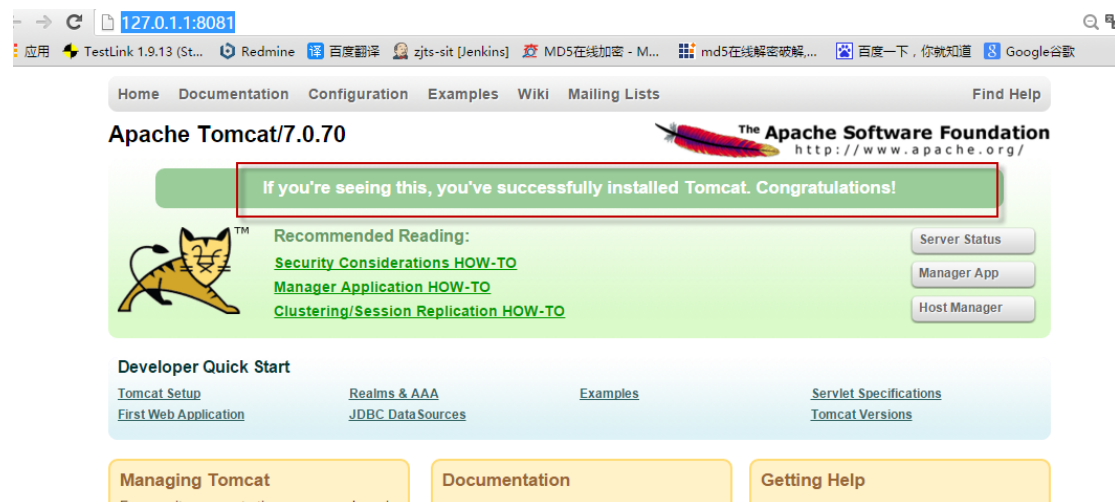


启用或关闭服务

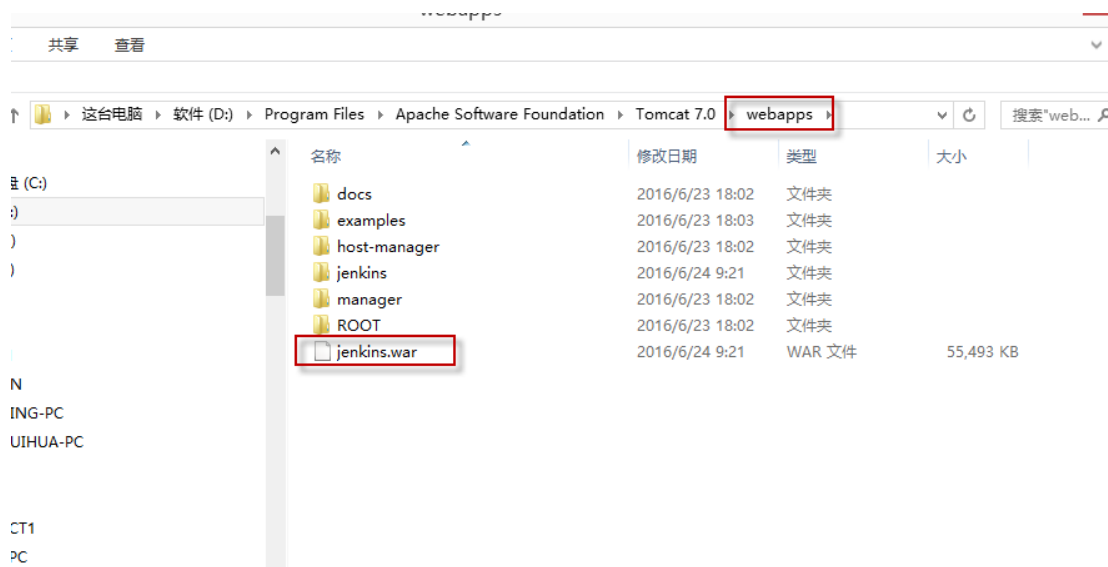




安装过程默认端口为 8080，因为我本地安装 jenkins 已占用，所以把默认端口修改为 8081 或者其他。。，安装成功后浏览器访问 <http://127.0.1.1:8081/> 成功会出现如下页面



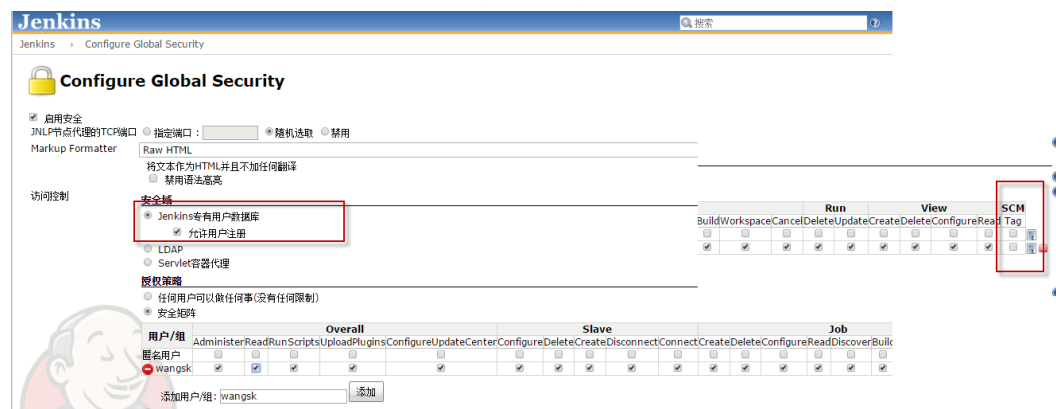
2 、下载 jenkins.war 包，下载成功后放到 tomcat webapps 目录



然后访问：<http://127.0.1.1:8081/jenkins/> jenkins 环境部署成功如下图



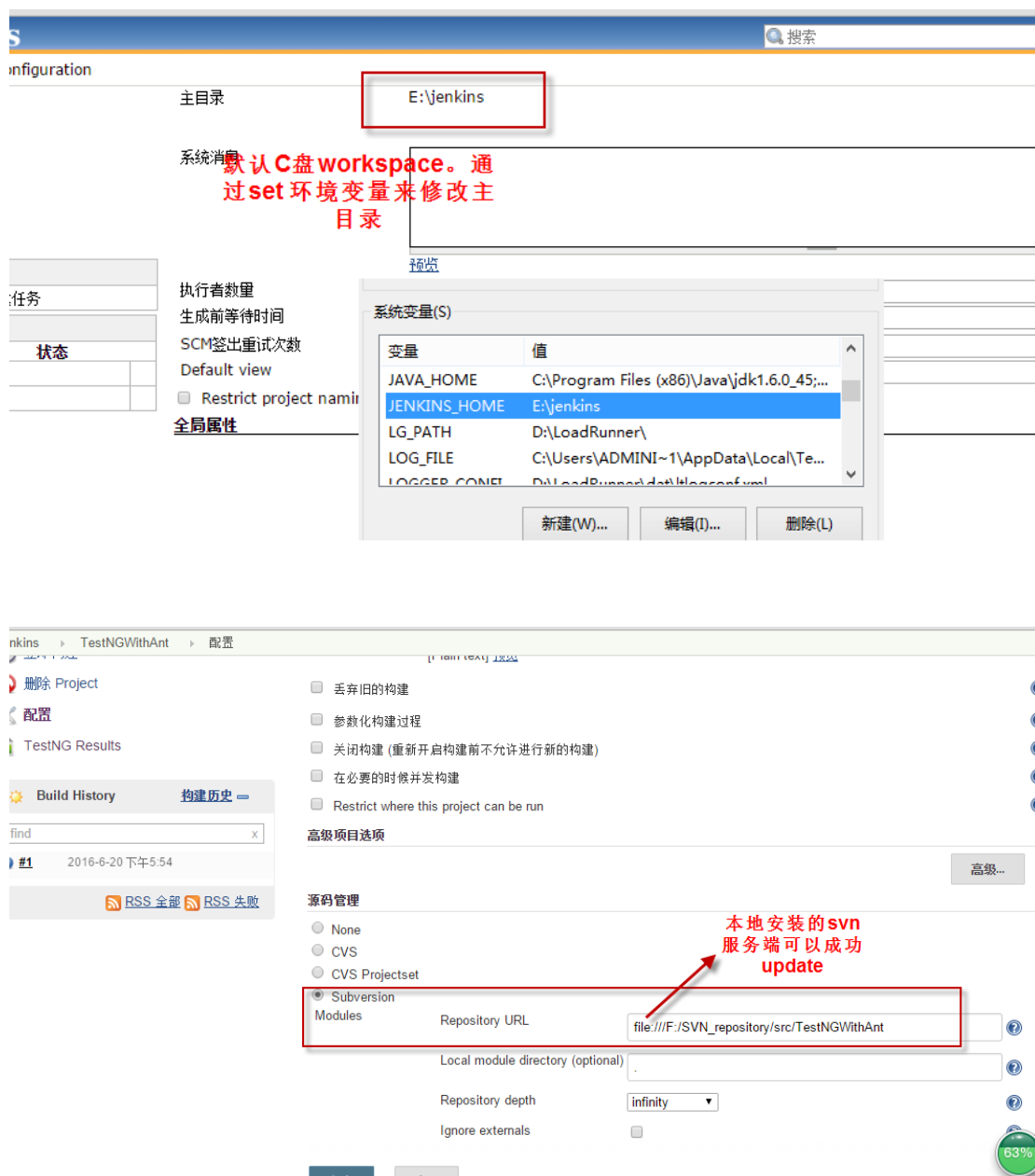
Jenkins 配置比起 jmeter 配置起来相对简单一些，如图配置即可运行
首先系统设置启用安全，添加用户信息及权限



注册用户



1 job 配置和主目录修改



注意 svn 版本与插件版本 否则会报错的

Jenkins 配置

jenkins系统设置 注意svn版本与插件版本的兼容问题

Default Compression Level: 3 (Recommended)

Private Key Location: C:\windows\system32\config\systemprofile\.ssh\id_rsa

Private Key Password:

Known Hosts Location: C:\windows\system32\config\systemprofile\.ssh\known

Authentication: 增加

Exclusion revprop name:

Subversion

Subversion Workspace Version: 1.6 (svn:externals to file)

Exclusion revprop name:

Subversion Plug-in

This plugin adds the Subversion support (via SVNKit) to Jenkins.

2.5.7 降到

解除绑定 卸载

2

构建

Invoke Ant

Ant Version: Default

Targets: test

Build File: F:\svn\src\TestNGWithAnt

Properties:

Java Options:

svn update到本地执行，不需要放在jenkins workspace，如果jenkins安装到tomcat需要修改buildfile，?

增加构建步骤

3

构建后操作

Publish TestNG Results

TestNG XML report pattern: **/testng-results.xml

高级...

删除

增加构建后操作步骤

保存 应用

七、Jenkins 设置自动构建

配置

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

No schedules so will never run

☐ Poll SCM

构建

Invoke Ant

Targets: run

高级...

删除

如：设每一个小时我们需要重新构建一次。选择 Build periodically，在 Schedule 中填写 0 * * * *。

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。

所以 0 * * * * 表示的就是每个小时的第 0 分钟执行一次构建。

PS：

Poll SCM：定时检查源码变更（根据 SCM 软件的版本号），如果有更新就 checkout 最新 code 下来，然后执行构建动作。我的配置如下：

*** / 5 * * * *** （每 5 分钟检查一次源码变化）

Build periodically：周期进行项目构建（它不 care 源码是否发生变化），我的配置如下：

0 2 * * * （每天 2:00 必须 build 一次源码）

八、Jenkins 生成 htmlreport 和发送邮件设置

1、发送邮件设置，插件管理安装 [Email Extension Plugin](#)

邮件通知

SMTP服务器: smtp.qq.com

用户默认邮件后缀: @qq.com

☒ 使用SMTP认证

用户名: 492982887@qq.com

密码:

☐ 使用SSL协议

SMTP端口: 25

Reply-To Address:

字符集: UTF-8

☒ 通过发送测试邮件测试配置

Test e-mail recipient: 492982887@qq.com

Email was successfully sent

Test configuration

在设置 Jenkins URL 底下有一个文本框 System Admin e-mail address，这里要设置发送者的邮箱地址，否则会报错

报错信息：

Failed to send out e-mail

com.sun.mail.smtp.SMTPSendFailedException: 501 mail from address must be same as authorization user; nested exception is:

com.sun.mail.smtp.SMTPSenderFailedException: 501 mail from address must be same as authorization user

at hudson.util.PluginServletFilter\$1.doFilter(PluginServletFilter.java:95)

Maven项目配置

全局MAVEN_OPTS:

Local Maven Repository: Default (~/.m2/repository)

☒ Help make Jenkins better by sending anonymous usage statistics and crash reports to the Jenkins project.

Jenkins Location

Jenkins URL: http://127.0.0.1:8080/

系统管理员邮件地址: 492982887@qq.com

SSH Server

SSH Port: ☐ 指定端口: ☒ 随机选取 ☐ 禁用

CVS

Default Compression Level: 3 (Recommended)

Private Key Location: C:\windows\system32\config\systemprofile\.ssh\id_rsa

Private Key Password:

Known Hosts Location: C:\windows\system32\config\systemprofile\.ssh\known_hosts

Authentication: 增加

项目中增加构建步骤 email notification

删除

E-mail Notification

Recipients

492982887@qq.com

Whitespace-separated list of recipient addresses. May reference build parameters like \$PARAM. E-mail will be sent when a build fails, becomes unstable or returns to stable.

☒ 每次不稳定的构建都发送邮件通知

☐ 单独发送邮件给对构建造成不良影响的责任人

删除

Editable Email Notification

Disable Extended Email Publisher

Allows the user to disable the publisher, while maintaining the settings

Project Recipient List

492982887@qq.com

Comma-separated list of email address that should receive notifications for this project.

Project Reply-To List

\$DEFAULT_REPLYTO

Comma-separated list of email address that should be in the Reply-To header for this project.

Content Type

HTML (text/html)

Default Subject

\$DEFAULT_SUBJECT

Default Content

\$DEFAULT_CONTENT

Attachments

Can use wildcards like 'module/dist/**/*'.zip'. See the @includes of Ant fileset for the exact format. The base directory is the workspace.

Attach Build Log

Do Not Attach Build Log

Content Token Reference

Editable Email Notification

Disable Extended Email Publisher

Allows the user to disable the publisher, while maintaining the settings

Project Recipient List

492982887@qq.com

Comma-separated list of email address that should receive notifications for this project.

Project Reply-To List

\$DEFAULT_REPLYTO

Comma-separated list of email address that should be in the Reply-To header for this project.

Content Type

HTML (text/html)

Default Subject

\$DEFAULT_SUBJECT

Default Content

\$(JELLY_SCRIPT,template="html")

Attachments

Can use wildcards like 'module/dist/**/*'.zip'. See the @includes of Ant fileset for the exact format. The base directory is the workspace.

Attach Build Log

Do Not Attach Build Log

Content Token Reference

Advanced Settings...

\$(JELLY_SCRIPT,template="html") JELLY 是内置的插件，直接用即可

Content Token Reference

Pre-send Script `$DEFAULT_PRESEND_SCRIPT`

Additional groovy classpath

Save to Workspace ☐

Triggers

☒ Failure - Any
Send To ☐ Developers

Recipient List `492982887@qq.com`

Reply-To List `$PROJECT_DEFAULT_REPLYTO`

Content Type

Subject `$PROJECT_DEFAULT_SUBJECT`

☒ Success
Send To ☐ Developers

Recipient List `492982887@qq.com`

Reply-To List `$PROJECT_DEFAULT_REPLYTO`

Content Type

Subject `$PROJECT_DEFAULT_SUBJECT`

Content `$PROJECT_DEFAULT_CONTENT`

Attachments

Can use wildcards like 'module/dist/**/*.zip'. See the [@includes of Ant fileset](#) for the exact format. The base directory is [the workspace](#).

Content Type 选择 html

```
C:\Program Files (x86)\Jenkins\workspace\jmeter_interface01\results>cd 46
C:\Program Files (x86)\Jenkins\workspace\jmeter_interface01\results\46>exit 0
[htmlpublisher] Archiving HTML reports...
[htmlpublisher] Archiving at PROJECT level C:\Program Files (x86)\Jenkins\workspace\jmeter_interface01\results\46 to C:\Program Files
(x86)\Jenkins\jobs\jmeter_interface01\htmlreports\TestReport
Performance: Percentage of relative difference outside -0.0 to +0.0 % sets the build as failure
Performance: Percentage of relative difference outside -0.0 to +0.0 % sets the build as unstable
配置成功触发发送邮件

Performance: Parsing JMeter report file 'C:\Program Files (x86)\Jenkins\jobs\jmeter_interface01\builds\46\performance-
reports\JMeter\TestReport.jtl'.
Email was triggered for: Success
Sending email for trigger: Success
Sending email to: 492982887@qq.com
Finished: SUCCESS
```

收到邮件内容

BUILD SUCCESS

Build URL http://127.0.0.1:8080/job/jmeter_interface01/52/

Project: jmeter_interface01

Date of build: Thu, 28 Jan 2016 15:23:56 +0800

Build duration: 43 秒

CHANGES

No Changes