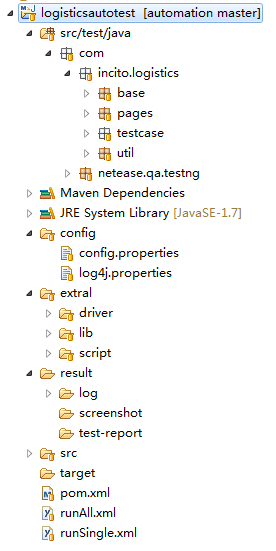
**物流web自动化测试框架介绍与使用**

## **一、框架介绍篇**

物流web自动化测试框架是基于selenium + maven + testng + github +Jenkins搭建的，其中selenium用来编写测试用例，maven用于项目的构建，testng用于执行测试，相当于执行者，github用于代码的托管，jenkins用于持续集成。整个框架可实现用例的自动执行测试、失败的用例自动重试、失败的用例自动截图，加之整个框架集成了jenkins,可使所有功能模块的测试用例定时并持续的集成下去。最后的测试报告可以通过jenkins邮件服务系统发送给相关人员。 在此着重介绍下selenium，selenium是一个用于Web应用程序测试的工具。Selenium测试直接运行在浏览器中，就像真正的用户在操作一样。支持的浏览器包括IE(7、8、9、10、11)、Mozilla Firefox、Google Chrome等。由于是JAVA语言编写，所以具有跨平台性，支持windows、Linux和MAC OS等。Selenium 是ThoughtWorks专门为Web应用程序编写的一个验收测试工具。

注：本框架中用到的工具都是开源的，不涉及到收费工具，故不需要担心版权问题。

## **二、目录解析篇**



项目整体是一个maven项目，因为maven可以自动帮助我们构建项目、自动下载我们所需要的依赖（jar包），只需要在pom.xml文件下配置好相关数据即可。

* src/test/java：maven项目中存放测试代码的目录
  + com/incite/logistics：物流项目目录
    - base：里面有个BaseParpare.java用于初始化浏览器和结束浏览器的操作
    - pages：存放每个页面上的元素的java类（每个页面就是一个java类，里面存放着声明此页面上要用到的所有元素）
    - testcase：存放测试用例的目录，此目录下按照功能模块再细分目录（模块），每个子目录都是各个模块的测试用例
    - util：存放逻辑操作的代码目录，此目录下目前有2个java类：SelectExplorer.java和SeleniumUtil.java
      * SelectExplorer.java：顾名思义，这是一个浏览器选择的逻辑操作类
      * SeleniumUtil.java：包装了selenium所有常用的方法包括自定义的方法
  + com/netease/qa/testing：这个是网易的开发的一个基于testng的插件，名字叫做：arrow，此插件可以实现的功能有：用例失败自动重试、失败用例会自动截图，生成的测试报表比testng自带的更直观，自动去重结果（比如第一次用例跑失败了，但是经过第二次重试，用例成功执行，arrow就会把第一轮失败的记录移除，保留第二次正确的测试记录）
* config：存放log4j的配置文件（log4j.properties）和arrow插件的配置文件（config.properties）,关于log4j的配置网上比比皆是，可以参考[这里](http://blog.csdn.net/azheng270/article/details/2173430/)，这里介绍下arrow的配置：

config.properties：testng插件arrow的配置文件,内容如下：

**retrycount=n** //定义重跑次数，就是用例跑失败之后，再跑n（n是正整数）遍

**sourcecodedir=src/test/java/com/incito/logistics/testcase/**  //指定测试用例的目录

**sourcecodeencoding=UTF-8** //指定源码的字符编码

* extral：存档驱动，类库和第三方脚本的目录

driver：存放不同平台下的不同位数不同浏览器的的driver

lib：里面存放的备用jar包，比如有的jar包在maven中央仓库找不到（自己编写的），就只能存档，便于以后直接导入使用

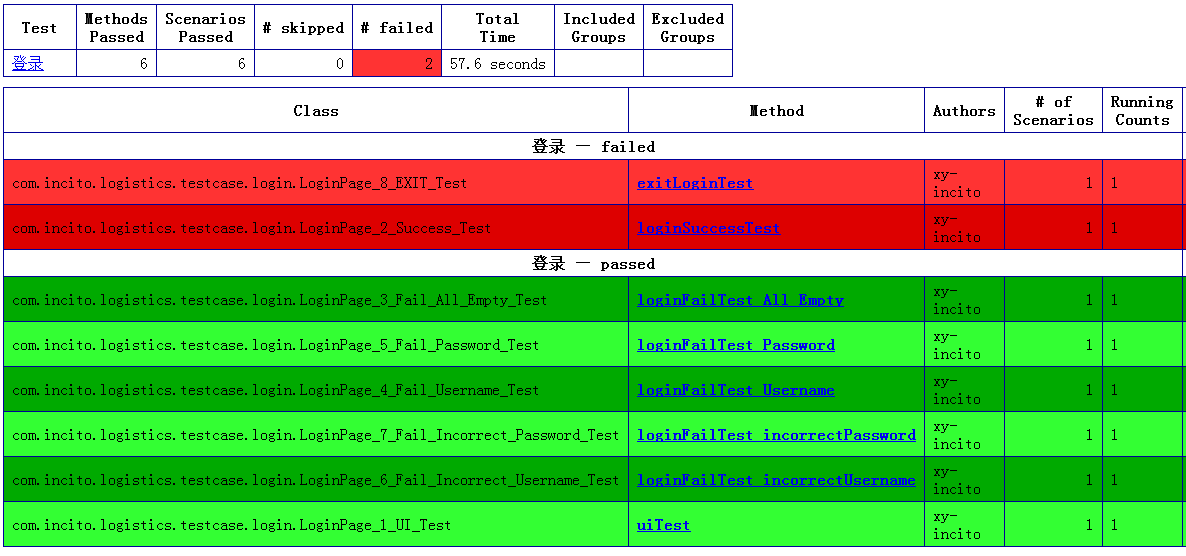
script：存放第三方的脚本辅助测试

* result：存放测试结果的目录

log：存放log4j的输出日志，只有运行了测试用例才会生成

screenshot：存放失败的用例的网页截图

test-report：测试报告生成目录，以HTML的方式显示，如图所示，附上一份完整的测试报表：



* pom.xml：maven的配置文件，项目核心配置，用于构建项目、自动下载项目依赖以及后续的和testng、jenkins配合持续集成等
* runAll.xml：这是testng的配置文件，用于存放部分测试数据以及测试的平台，浏览器的配置、加入第三方插件监听（arrow插件）、设置用例执行策略（多线程还是单线程，顺序执行还是无序执行以及是否依赖执行等）以及设置要执行的用例。之所以改成runAll顾名思义就是说执行整个web项目的所有模块的用例的测试
* runSingle.xml：和runAll.xml一样，不多解释，不一样的地方就是用它来做单个用例的调试改错，只涉及到单个的类（用例），所以调试改错专用。比如我用runAll.xml跑完了所有的用例，但是发现个别用例失败，此时先去分析log，如果是代码问题就去调试代码，调试完毕之后就去用runSingle.xml去跑一边此用例。

## **三、如何使用篇**

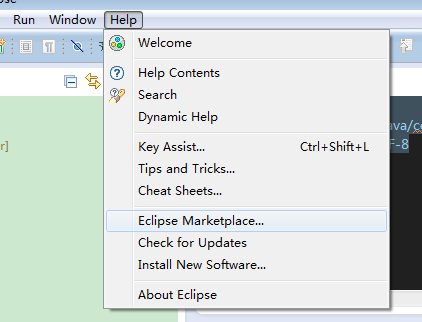
注：下面所讲到的是基于windows平台

### 3.1、安装配置JDK

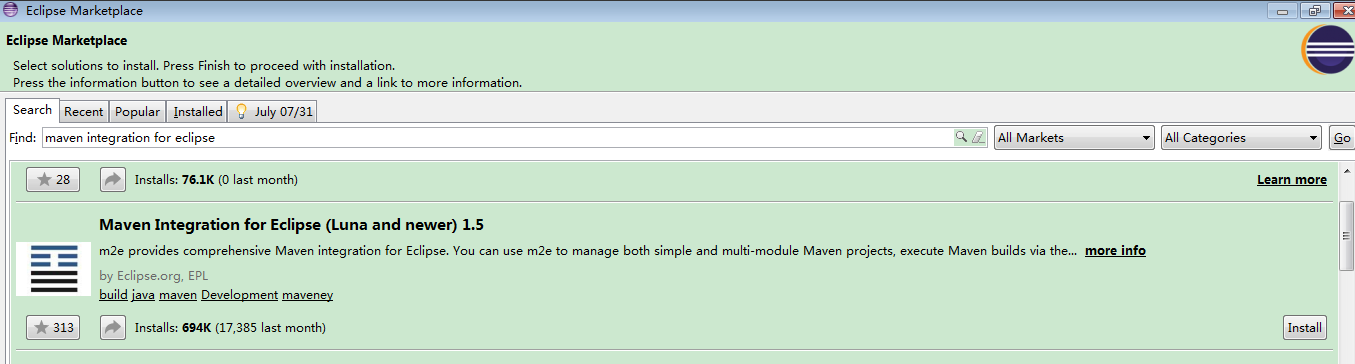
1. 下载安装JDK 7(32bit)，[点我下载](http://download.oracle.com/otn-pub/java/jdk/7u67-b01/jdk-7u67-windows-i586.exe)
2. 配置jdk环境变量，[点我查看如何配置](http://jingyan.baidu.com/article/6dad5075d1dc40a123e36ea3.html)

### 3.2、安装配置Maven

1. 下载maven,[点我下载](http://apache.dataguru.cn/maven/maven-3/3.2.2/binaries/apache-maven-3.2.2-bin.zip)
2. 配置maven环境变量，[点我查看如何配置](http://blog.csdn.net/lijingronghcit/article/details/9167075)
3. 下载安装Eclipse（32bit）,[点我下载](http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/kepler/SR2/eclipse-standard-kepler-SR2-win32.zip)
4. 解压刚才下载好的Eclipse到任意目录，双击Eclipse.exe打开
5. 安装必须的插件 maven integration for eclipse
6. 打开Eclipse菜单栏->Help->Eclipse Marketplace… 如图：

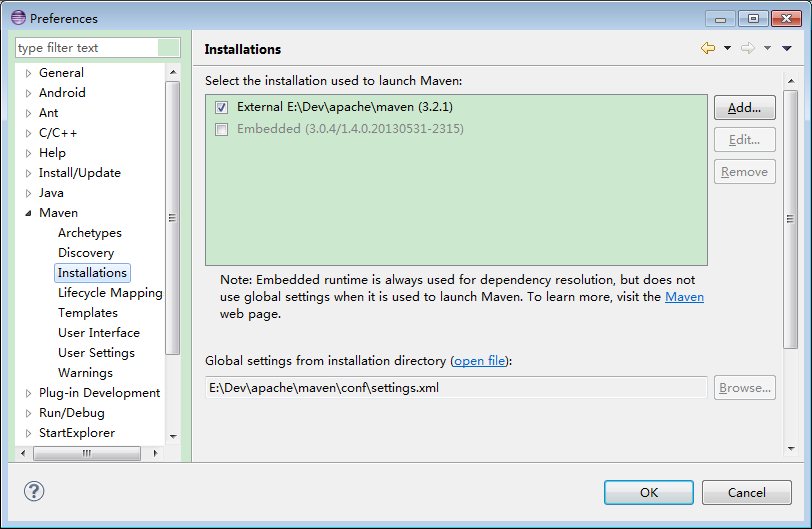


1. 搜索” maven integration for eclipse”安装，如图，点击install安装会重启Eclipse



1. 重启之后在Eclipse菜单栏->Window->Preferences点击打开，找到maven项，这个地地方需要配置2个地方（如图）：

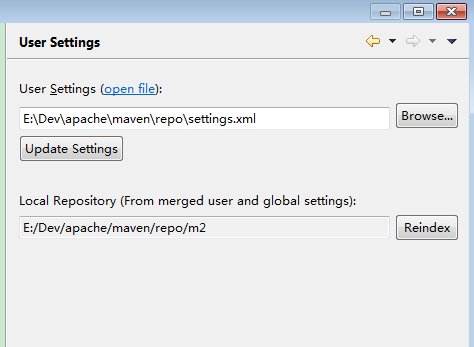
一、 Installations,点击Add，指向第三部中下载的maven的目录，比如我的是e:\Dev\appache\maven(就是maven home路径)，指定成功以后 下图中的Global Settings from installation diectory会自动定位到maven所在路径下的conf/settings.xml文件（settings.xml是maven的设置配置文件，它可以指定本地仓库的存放路径、可以指定远程中央仓库的地址），在这里提供下我自己的settings.xml文件： 里面已经配置了中央仓库为中国的服务器（开源中国的maven仓库）



二、User Settings：　用户设置（User Settings）这里会再次指定一个settings.xml这里是相当于局部变量，对当前用户适用，刚才前面的settings.xml是全局变量 针对所有用户。

由于前面我提供的settings.xml指定了本地仓库的路径为：e:\Dev\apache\maven\repo\m2这个是maven的目录下的，如果repo\m2目录不存在，请自建，然后把settings.xml文件放入repo目录 作为 user settings的配置文件。如果想更改本地仓库目录，请修改刚才提供的settings.xml文件中的：

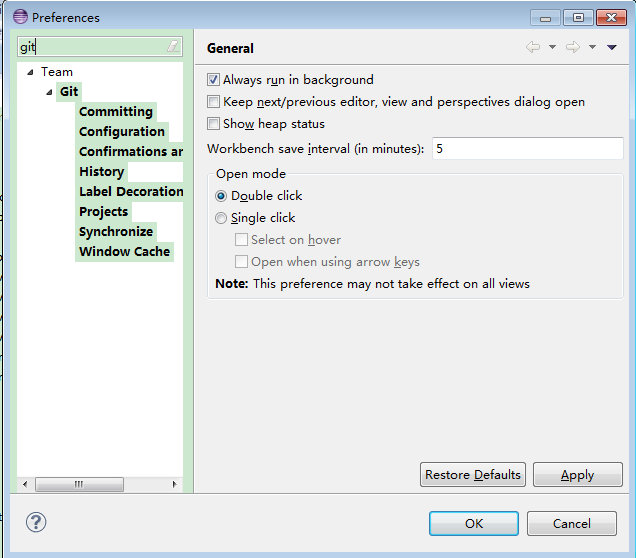
<localRepository>E:/Dev/apache/maven/repo/m2</localRepository> 即可



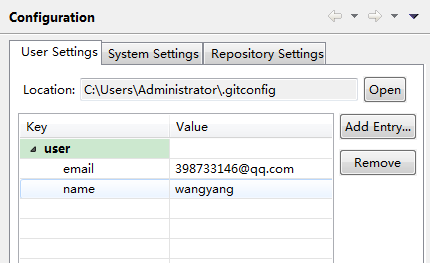
### 3.3、配置git

1. 从Eclipse中搜索安装egit插件
2. 配置egit，步骤：Eclipse菜单栏->Window->Preferences点击打开，搜索”git”(Team->Git->Congiguation->Add Entry..),然后分别建立一个key为user.email和user.name的变量， value分别填上你自己申请的github的email，name可以随便填写（区分身份）

图一：



图二：

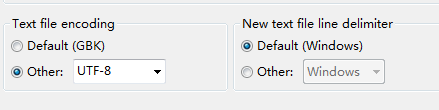


### 3.4、安装testng

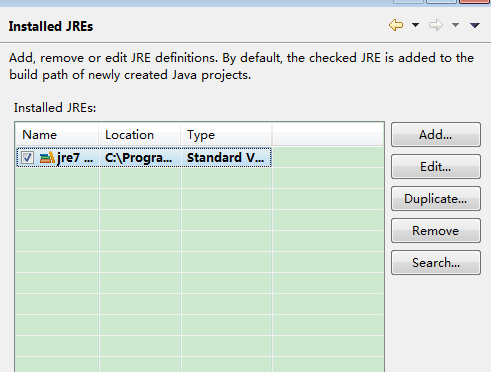
直接从Eclipse中搜索安装testng即可，不需要额外配置

### 3.5、eclispe中的一些设置

1. 更改Eclipse编码为UTF-8，操作如下：Eclipse菜单栏->Window->Preferences点击打开->General->Workspace->Text file encoding –>other:UTF-8,如图：

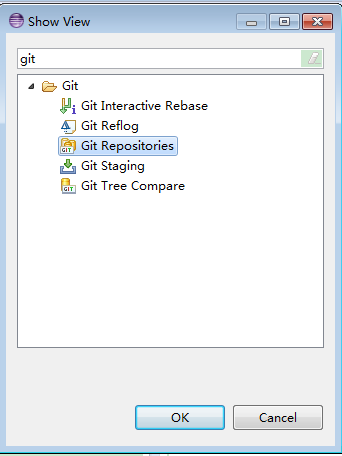


1. 更改Installed JREs的路径，操作如下：Eclipse菜单栏->Window->Preferences点击打开->Java->Installed JREs->Edit->然后选择前面步骤安装好的 java的jdk目录，比如我的：C:\Program Files (x86)\Java\jdk1.7.0\_65

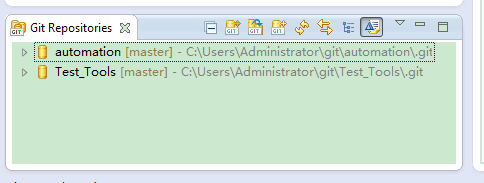


### 3.6、从github克隆项目

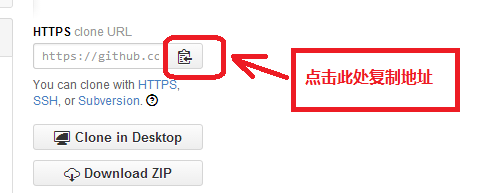
1. 进入Eclipse菜单栏->window->Show view->other->搜索git->选择Git Repositories，然后点击OK



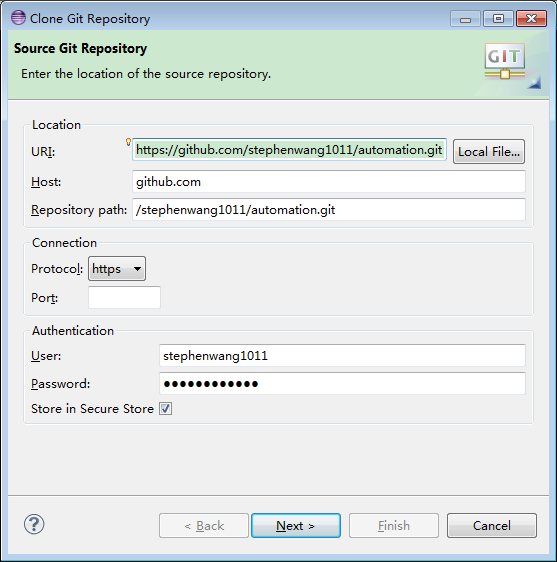
1. 之后Eclipse中会出现这样一个视图：



1. 打开此网址：<https://github.com/stephenwang1011/automation>，然后按照下图操作



1. 复制完成以后点击第二步中的clone图标：，你刚才复制的地址会自动补全，如图：

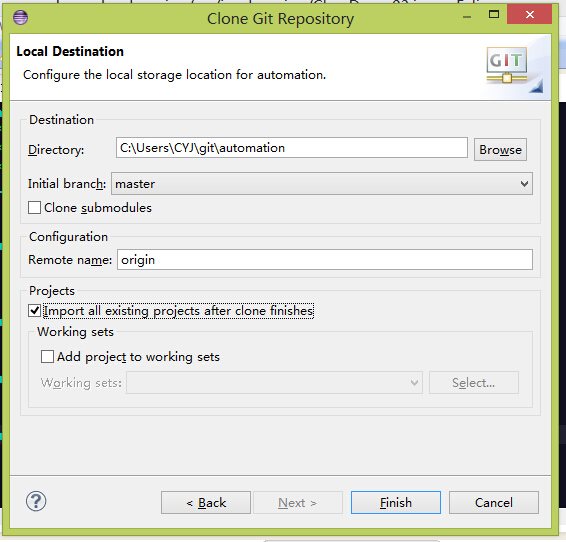


其中的

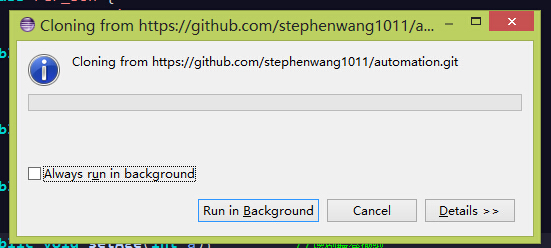


需要填写你的github用户名和密码，前提是你需要先注册一个。

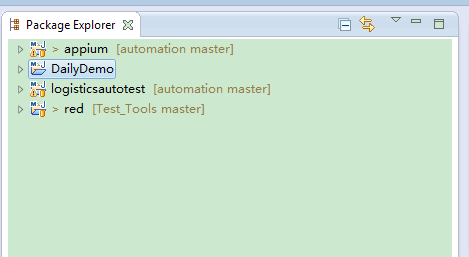
1. 点击上图的“Next”直到下个页面，此时勾选下“Import all existing projects after clone finshes”



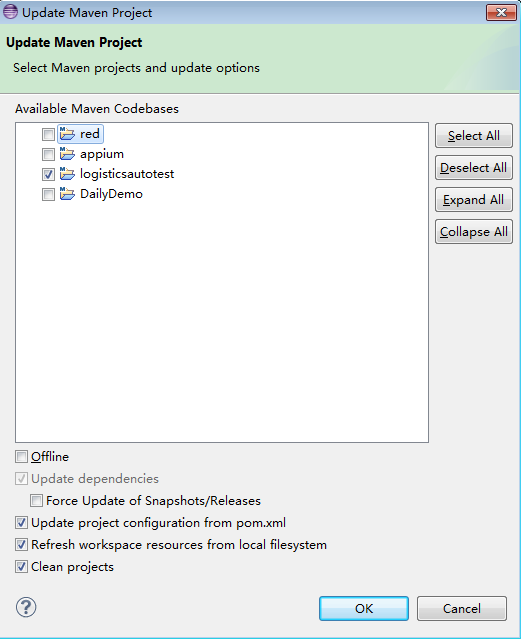
1. 点击“Finish”即可



1. 完成clone之后，会在package explorer中显示出github中的项目，如图

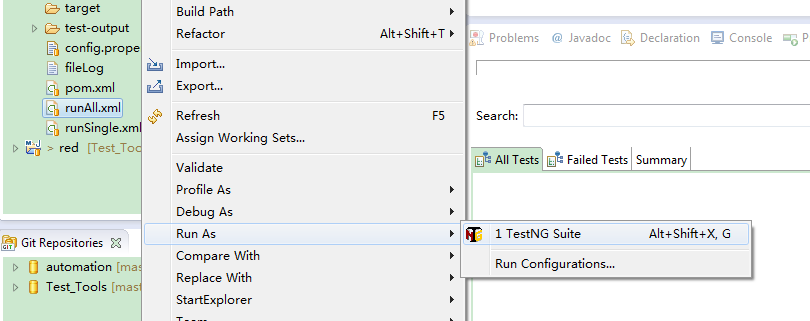


1. 其中的logisticsautotest项目就是物流货代web端的自动化测试，这个时候还需要借助maven 构建下。首先选中这个项目，然后右键->maven->Update Project..->然后按照下图操作->点击OK即可。首次导入项目maven会下载项目依赖以及各种插件，耗时会长点，有可能会由于网络或者其他原因导致失败，多试几次就可以了



### 3.7、执行测试用例

1. 确保以上步骤全部准备完毕并且无误，鼠标选中runAll.xml->Run as ->TestNG Suite



1. 此时会运行配置好的所有测试用例

### 3.8、测试配置更改

1. 如果说你想更改测试的浏览器或者说测试的平台，测试数据等等，可以在runAll.xml里面修改，如何修改，可以直接打开此文件，里面我做了详细的注释，哪个地方是做什么的看了就懂
2. 如何修改用例重跑次数？打开config.properties，修改参数retrycount=n n是整数
3. 其他配置待补充…

## 四、jenkins持续集成测试篇

### 4.1、安装jenkins

1. 官网下载jenkins windows安装版：[点击下载](http://mirrors.jenkins-ci.org/windows/latest)
2. 点击exe文件执行安装

### 4.2、配置jenkins（全局）

1. 在jenkins的安装修改jenkins.xml中的

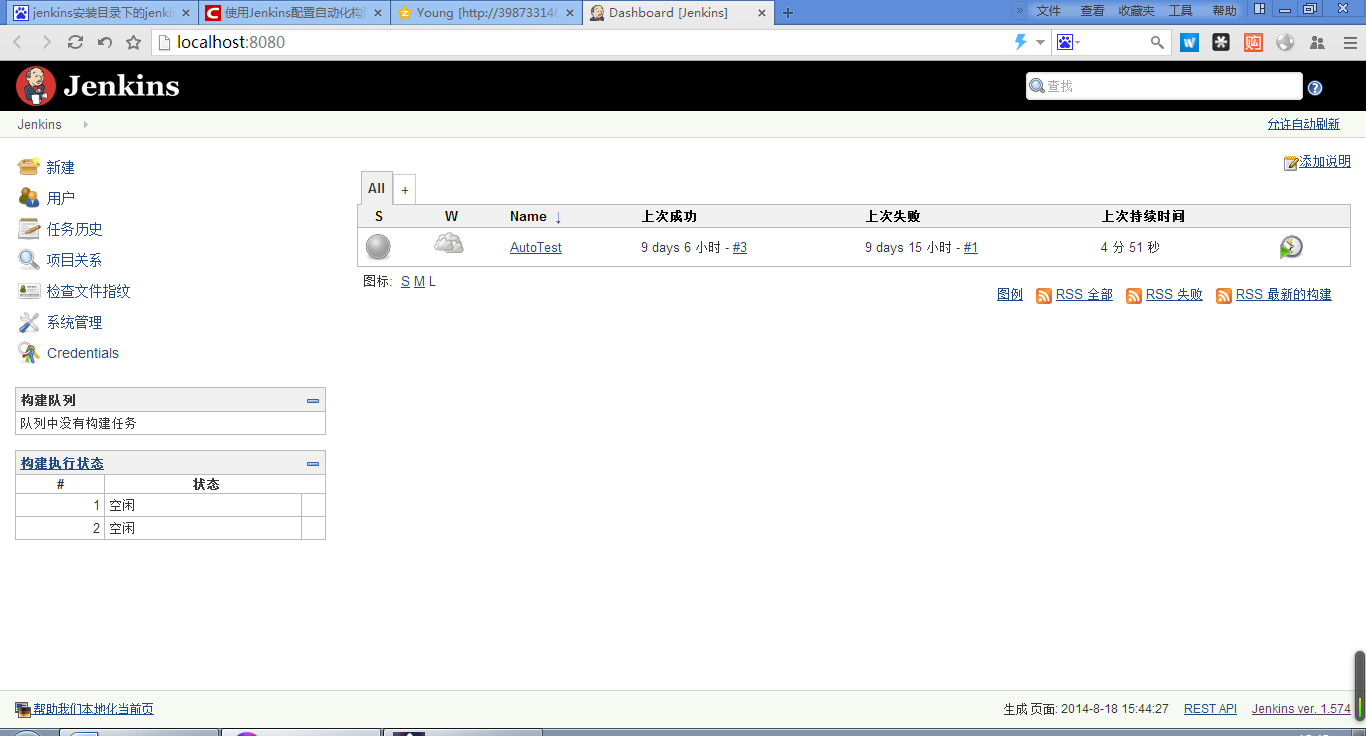
<arguments>-Xrs -Xmx256m -Dhudson.lifecycle=hudson.lifecycle.WindowsServiceLifecycle -jar "%BASE%\jenkins.war" --httpPort=8080</arguments>

为

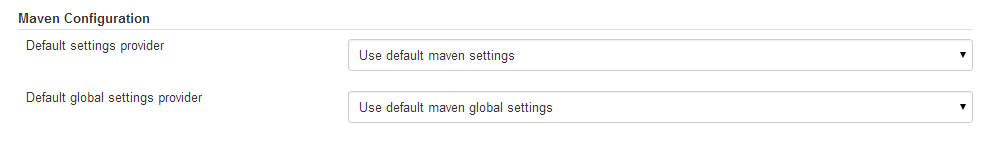
<arguments>-Xrs -Xmx256m –Dfile=utf-8 -Dhudson.lifecycle=hudson.lifecycle.WindowsServiceLifecycle -jar "%BASE%\jenkins.war" --httpPort=8080</arguments>

这一步主要是为了解决jenkins构建时中文出现乱码的情况

1. 浏览器中输入<http://localhost:8080>此时会打开jenkins主页，如图



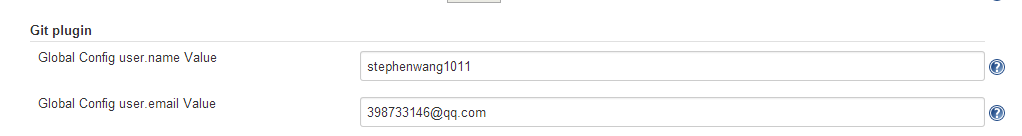
1. 在主页点击“系统管理”->插件管理，在可选插件中搜索git plugins和github plugins两个插件进行安装，安装后重启jenkins生效
2. 再进入“系统管理”->“系统设置”进行更详细的配置
3. Maven配置，如图



1. Git配置：可以手动安装下载git之后制定git.exe路径，也可以直接在配置处勾选自动安装。比如我的是手动安装的git,如图：



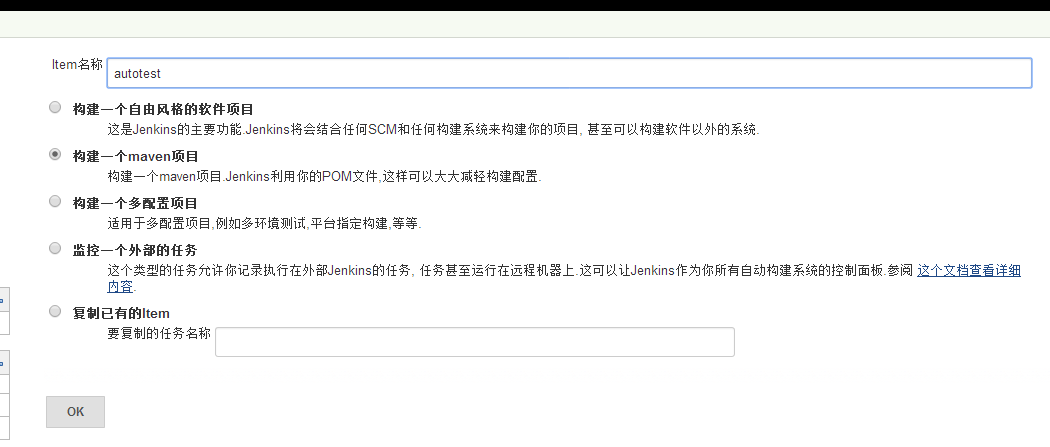
1. 配置github的用户名和邮箱



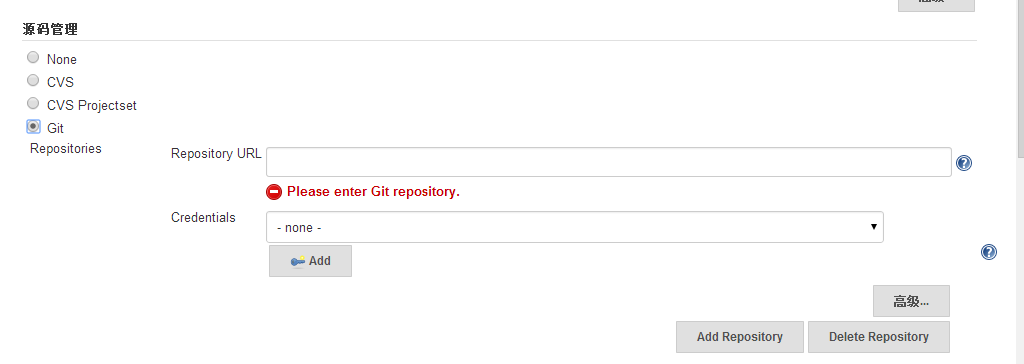
1. 点击应用按钮即可保存此页的设置

### 4.3、新建构建项目

1. 点击首页新建按钮
2. 构建一个maven项目，如图



1. 源码管理：选择git，URL中输入项目库地址



1. 构建定时触发器：勾选Poll SCM，日程表中输入：H 8 \* \* \*意思是：每天上午八点自动运行项目



1. 构建：指定好项目pom.xml。本项目路径：logistics/pom.xml。Goals and options输入 test即可



1. 点击应用按钮本页设置保存生效

### 4.4、手动执行构建

1. 在jenkins首页会看到自己刚才新建的maven项目:test
2. 如图所示：进行手动构建



## 五、其他说明

### 5.1、页面定位工具

一般使用firefox中插件firebug和firepath可以精准定位页面元素位置，其他浏览器可以通过右键->”审查元素”查看页面元素

### 5.2、提供selenium入门文档

1. [Selenium Testing Tools Cookbook中文](http://yunpan.cn/QaAXSEFubNtNA) （提取码：2073）
2. [selenium webdriver 学习](http://yunpan.cn/QaAXMFghkQq6z)（提取码：df50）

### 5.3、联系方式

如果文档中有不明确或者表达不清楚或者理解有难度的可以联系我。

Email: wangyang@incito.com.cn

QQ: 398733146

Mobile: 18872573204

2014年8月

