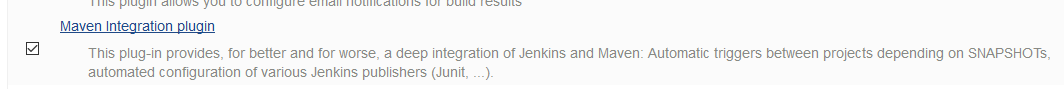
第一步、安装maven插件

1.进入【Manage Jenkins】->【Manage Plugins】->【可选插件】



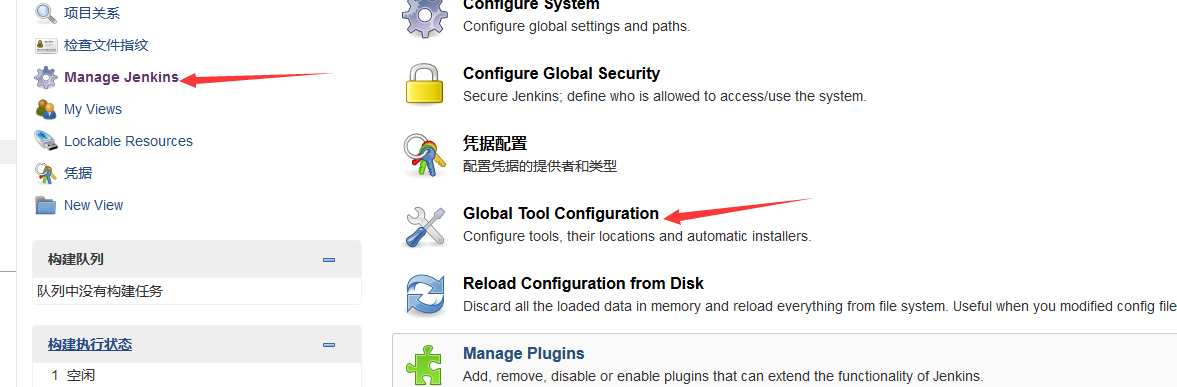
2.搜索【maven】->选择【Maven Integeration plugin】安装





第二步、配置maven环境

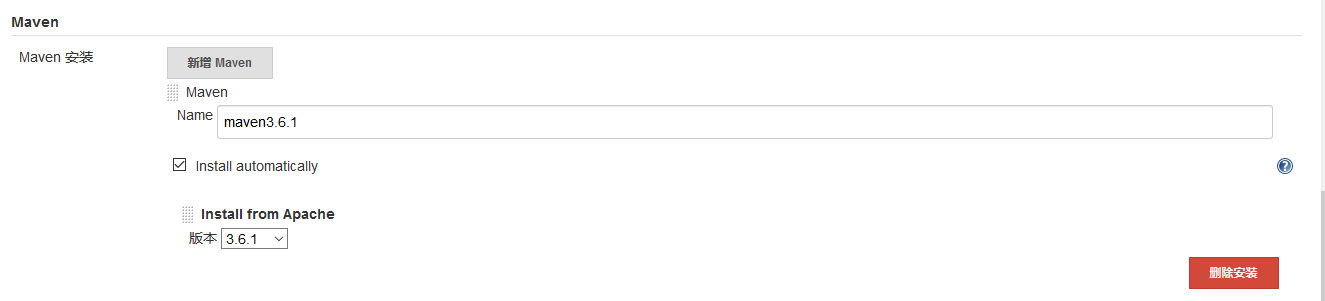
1. 进入【Manage Jenkins】->【Global Tool …】



1. 配置maven的配置文件（我这里是本地仓库）



1. 添加maven



第三步、构建maven项目

1. 新建项目。点击【新建Item】

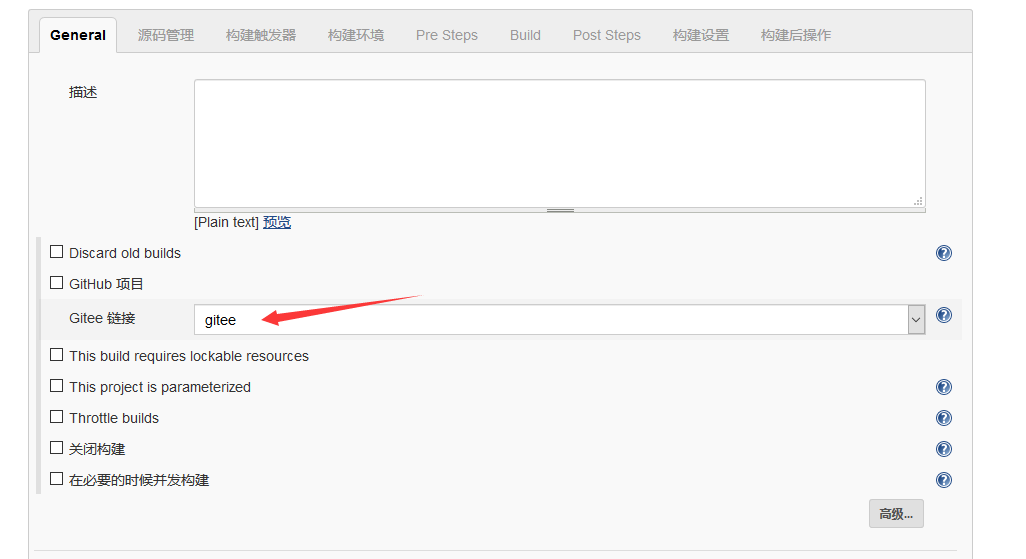


输入【任务名称】->选择【构建一个Maven项目】->点击【确定】

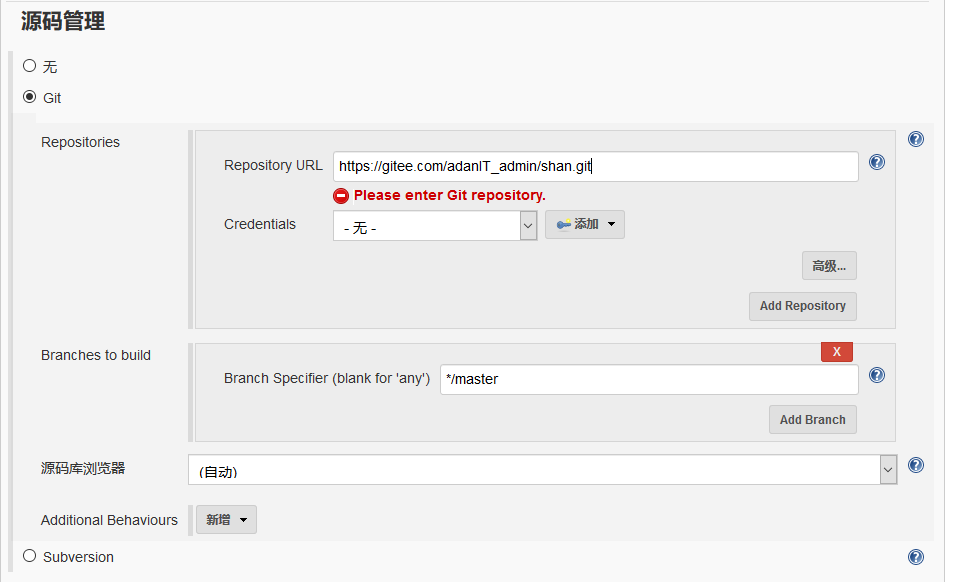


1. 配置源码

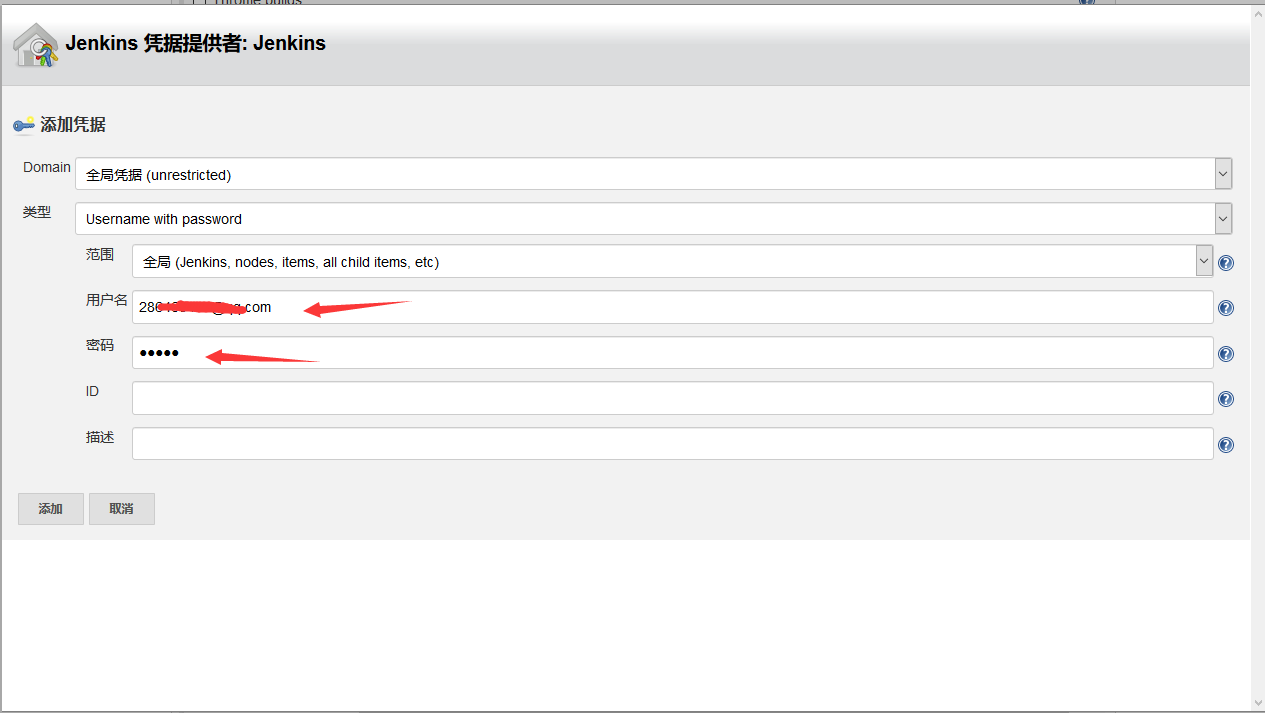
我的是gitee项目，所以配置了gitee；



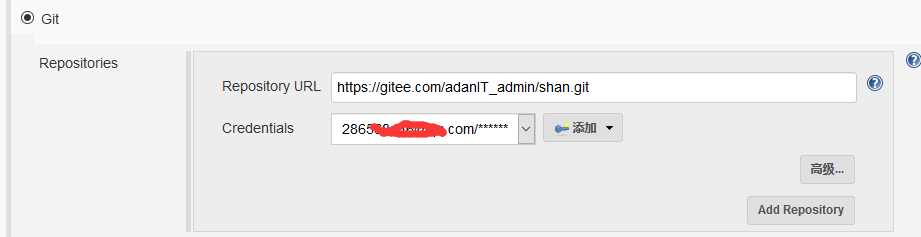
点击【git】->填入【Repository URL】



选择【Credentials】,没有配置就【添加】我是获取gitee的源码，配置的是gitee的账号与密码。具体看下图



添加完后再选择刚配置好的



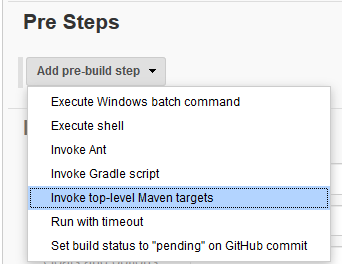
1. 配置触发器



1. 配置项目依赖

打包项目前做的事情

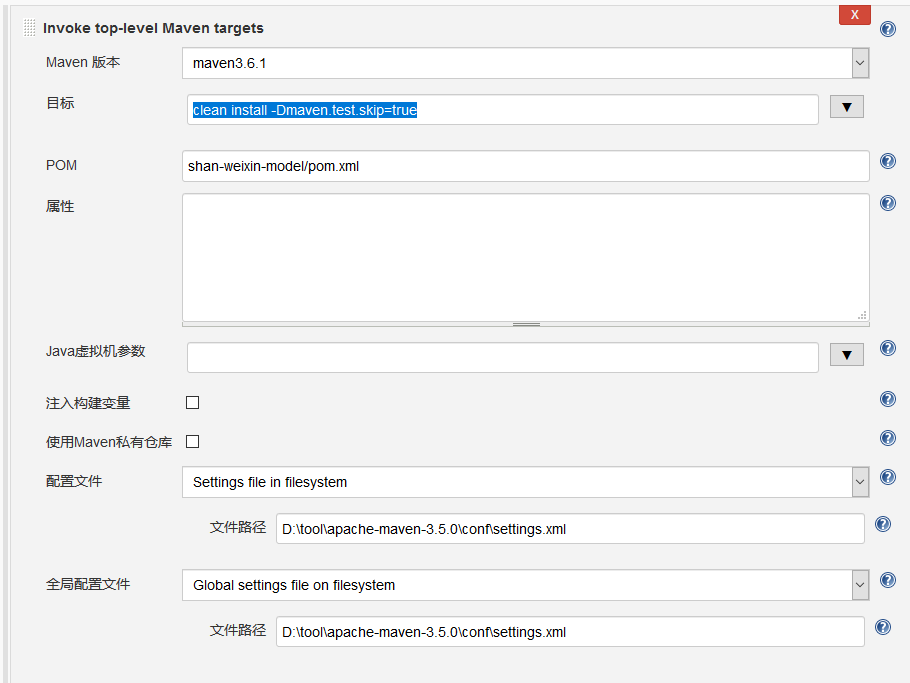
选择【Invoke top-level Maven targets】



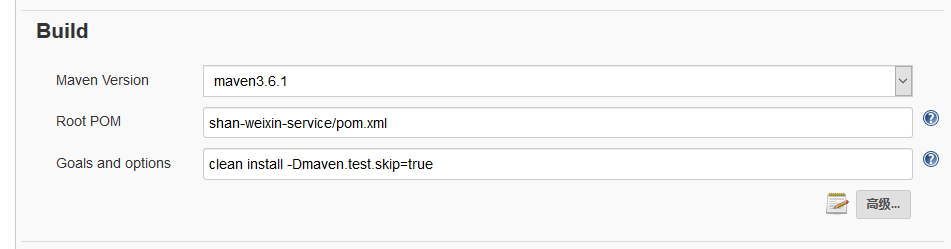
Maven版本：选择上面配置的

目标：clean install -Dmaven.test.skip=true 意思是执行maven命令 清除、打包、不执行测试

POM:项目pom.xml文件



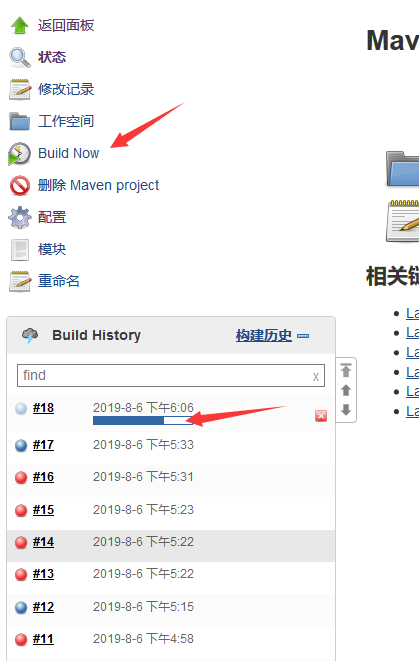
1. 构造主项目



1. 构建项目

点击【Build Now】

下面可以看到正在构建



第四步、Maven的profile功能

描述：在实际业务中，系统需要在多种环境上运行，如：开发环境、测试环境、生产环境。不同的环境配置往往不同，比如数据库链接，开发环境与生产环境不同。如果我们使用一套配置，当系统从开发环境迁移到生产环境那么需要更改代码或配置文件来更换配置。这样就增加了工作复杂度，容易出错。但Maven的profile功能，能帮我们解决这一问题。

1. 在项目中pom.xml中添加profile元素

<profiles>

<profile>

<!-- 开发环境 -->

<id>dev</id>

<properties>

<profileActive>dev</profileActive>

</properties>

<!-- 没有指定，此元素就是默认 -->

<activation>

<activeByDefault>true</activeByDefault>

</activation>

</profile>

<profile>

<!-- 测试环境 -->

<id>test</id>

<properties>

<profileActive>test</profileActive>

</properties>

</profile>

<profile>

<!-- 生产环境 -->

<id>prod</id>

<properties>

<profileActive>prod</profileActive>

</properties>

</profile>

</profiles>

1. 在pom中添加resource元素

<resources>

<resource>

<!-- 标识配置文件所在的目录 -->

<directory>src/main/resources</directory>

<filtering>true</filtering>

<!-- 构建时将这些配置文件全都排除掉 -->

<excludes>

<exclude>application.yml</exclude>

<exclude>application-dev.yml</exclude>

<exclude>application-test.yml</exclude>

<exclude>application-prod.yml</exclude>

</excludes>

</resource>

<resource>

<!-- 标识配置文件所在的目录 -->

<directory>src/main/resources</directory>

<filtering>true</filtering>

<!-- 构建时将哪些文件加入打包 -->

<includes>

<include>application.yml</include>

<!-- ${profileActive}这个值会在maven构建时传入 -->

<include>application-${profileActive}.yml</include>

</includes>

</resource>

</resources>

1. 在pom中添加插件maven-resources-plugin该插件用来在Maven构建时参数替换

<plugin>

<artifactId>maven-resources-plugin</artifactId>

<configuration>

<delimiters>

<!-- 表示用$来替换 -->

<delimiter>$</delimiter>

</delimiters>

<useDefaultDelimiters>false</useDefaultDelimiters>

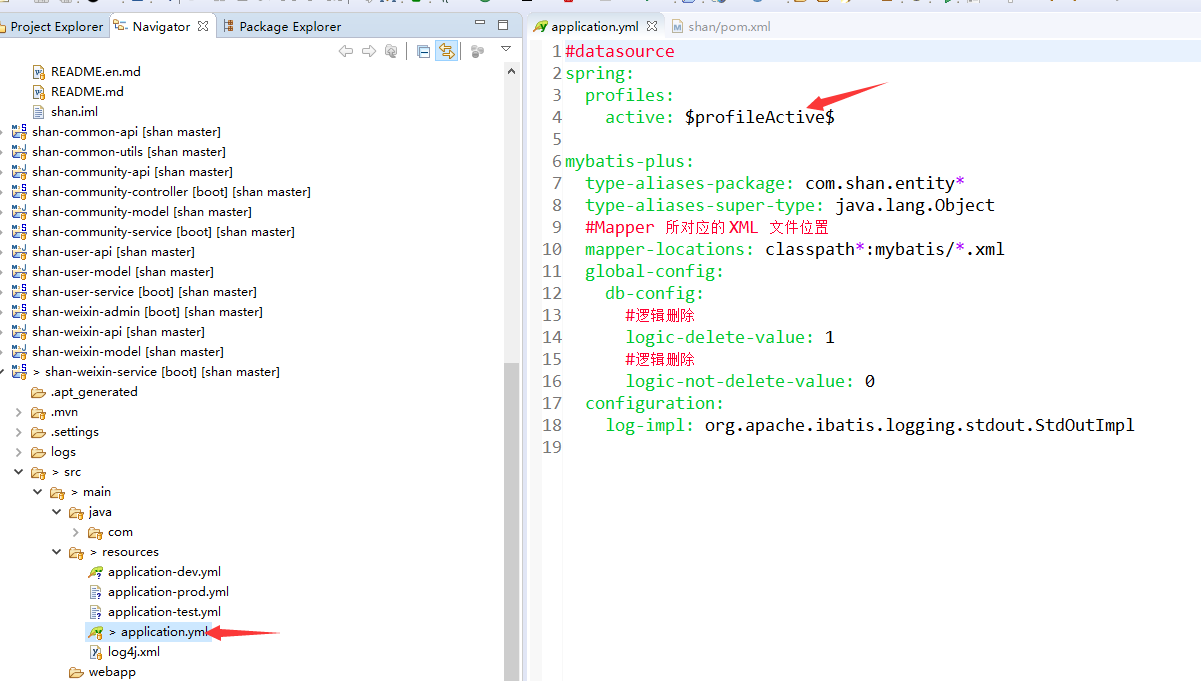
</configuration>

</plugin>

1. 在项目中创建配置文件

分别为dev开发环境、test测试环境、prod生产环境对应的配置文件。

在总配置中application.yml配置如下：



1. 更改jenkins配置

在所有Jenkins中所有Maven命令的末尾添加 -P test，在打包的时候-P后面的参数将会作为$profileActive$的值传入系统中，从而根据该值打包相应的application-{profileActive}.yml文件。