## 其实我们部署和启动一个java项目:无非做的是:

1:把项目打成war包

2:把war包部署到tomcat/webapps目录下:

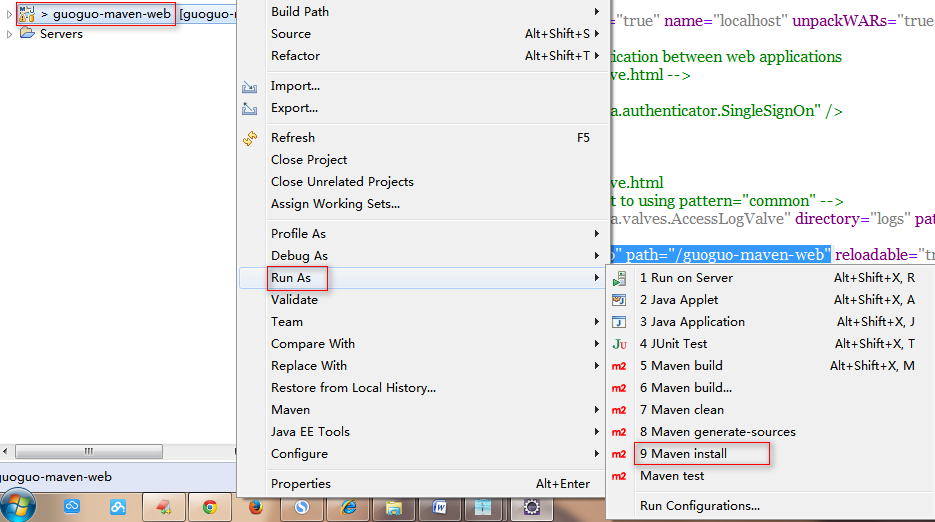
2:启动tomcat服务并访问

其实maven项目部署到tomcat的方式很多，**我从一开始的打war包到tomcat/webapps目录**，到使用tomcat-maven插件，到直接使用servers部署，一路来走过很多弯路。

下面就一一介绍这几种部署方式：

## 1.打war包到tomcat/webapps目录

这种方式其实跟非maven项目没什么区别，就是打包的方式不同



之后在target目录下会生成war包，复制到tomcat/webapps目录即完成部署。

## 2.使用tomcat-maven插件，在pom.xml的</dependencies>之后添加以下代码，并做相应修改

<build>

<finalName>guoguo-maven-web</finalName>

<plugins>

<plugin>

<groupId>org.apache.tomcat.maven</groupId>

<artifactId>tomcat7-maven-plugin</artifactId> <!-- 命令为tomcat7:redeploy -->

<!-- <groupId>org.codehaus.mojo</groupId> -->

<!-- <artifactId>tomcat-maven-plugin</artifactId> --> <!-- 命令为tomcat:redeploy -->

<!-- <version>2.2</version> -->

<configuration>

<url>http://localhost:8080/manager/text</url> <!-- tomcat7部署管理路径 -->

<username>admin</username> <!-- tomcat的管理员账号 -->

<password>admin</password>

<port>8080</port>

<path>/guoguo-maven-web</path> <!-- 部署路径 -->

<charset>UTF-8</charset>

<encoding>UTF-8</encoding>

<!-- 运行redeploy命令前，要能正常访问http://localhost:8080/manager-->

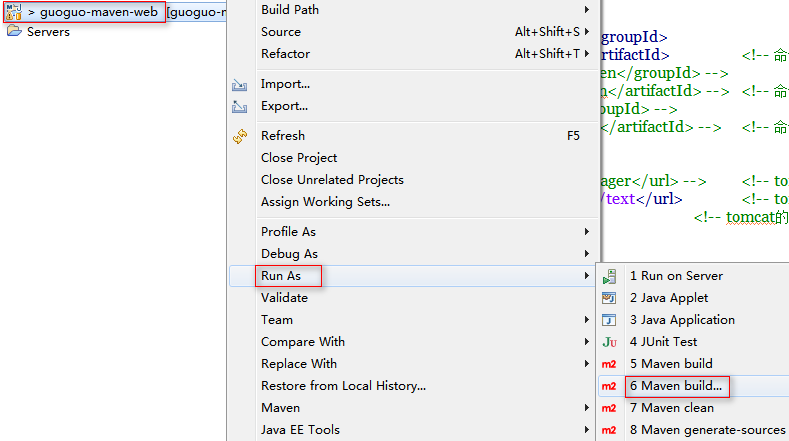
</configuration>

</plugin>

</plugins>

</build>

这样就配置好了tomcat maven插件

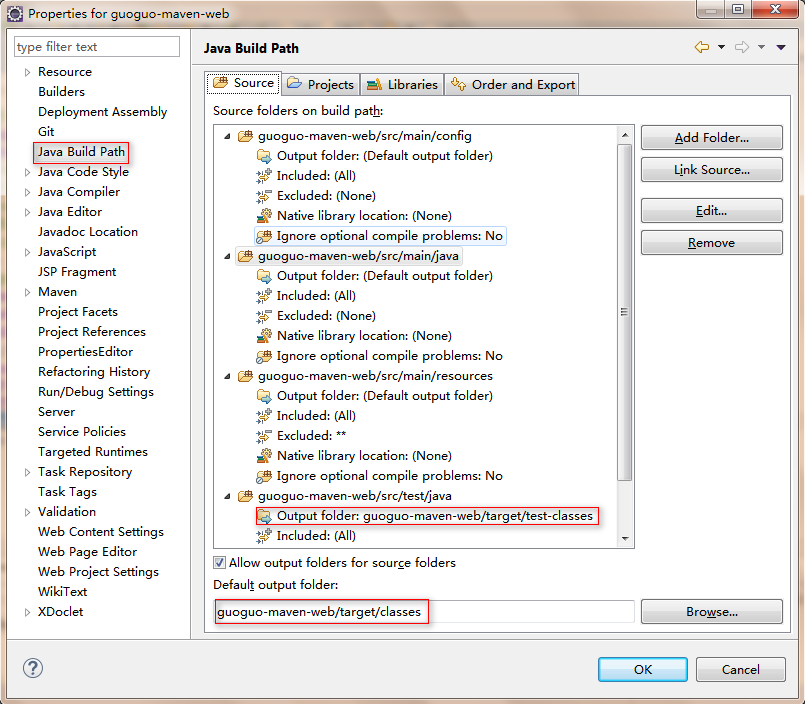


运行redeploy命令前，要启动tomcat，并能正常访问http://localhost:8080/manager

通过项目右键 run as --> maven build... --> main --> goals 中填入 tomcat6:redeploy命令即可部署成功，这样部署有时会使tomcat出错，出错需要重启tomcat

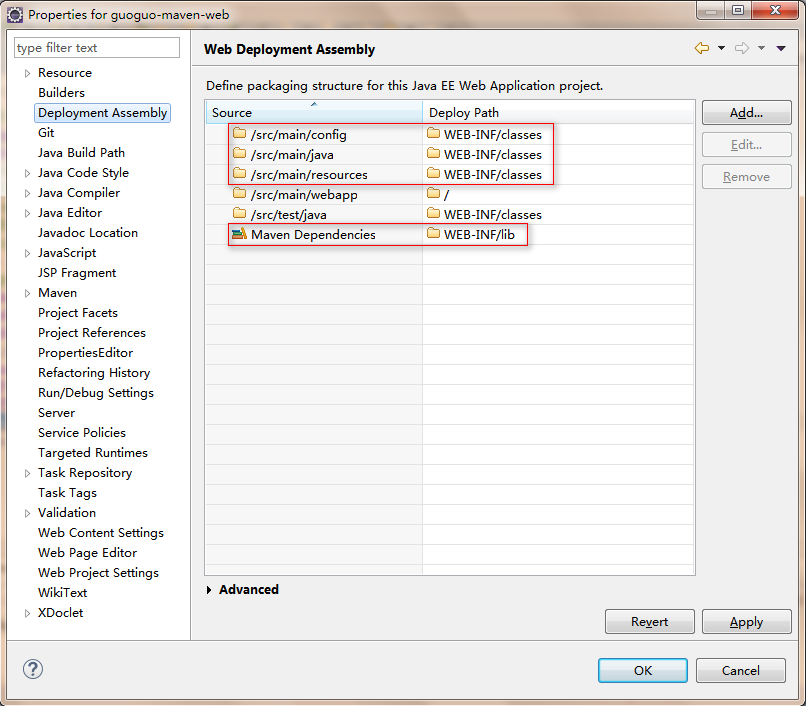
## 3.直接使用servers部署

首先确保编译配置正常



test下的目录编译到target/test-classes，其他编译到target/classes目录即可，其他一般默认不需要改变什么

然后进行部署的配置：



配置好之后，通过右键servers中tomcat，add and remove...添加项目，重启tomcat即可

第一种我已经不用了，第二种适合直接部署到测试服务器，第三种适合本地的调试

附录：

tomcat管理员配置，在servers项目的tomcat-users.xml中添加如下配置，如果你是直接使用bin/startup.bat启动tomcat，则修改conf/tomcat-users.xml

----------tomcat6管理员配置----------  
<role rolename="manager"/>  
<user username="admin" password="admin" roles="manager"/>  
----------tomcat6管理员配置----------

----------tomcat7管理员配置----------  
<role rolename="manager-script" />  
<role rolename="manager-gui"/>  
<user username="admin" password="admin" roles="manager-gui,manager-script"/>  
----------tomcat7管理员配置----------

## 持续快捷部署Web项目时候,我们有多个选择:

1:可以使用eclipse的插件tomcat6:redeploy

2:jenkins

3:Hudson

Hudson本身已经跟SVN、Git、Maven集成并且支持添加各种插件。但如果使用Hudson，我需要配置两个任务：

一个任务负责打包项目成WAR，

另外一个任务负责部署打包好的WAR包到目标服务器的Tomcat上。

虽然任务只需要配置一次，但每次修改代码提交后，都得切换到浏览器点Hudson构建任务页面上的Build Now按钮;

下面是使用eclipse的插件进行部署:

1、准备工作

    下载安装并配置好Tomcat和Maven。

    准备好一个Maven Web项目。

2、Maven部署Web项目到Tomcat的配置

2.1、配置Tomcat角色

Maven自动部署实际上调的是Tomcat安装目录下的manager功能。而为了能正常访问[http://localhost:8080/manager](http://localhost:8080/manager%E9%A1%B5%E9%9D%A2%EF%BC%8C%E6%88%91%E4%BB%AC%E9%9C%80%E8%A6%81%E4%BF%AE%E6%94%B9$TOMCAT_HOME/conf%E7%9B%AE%E5%BD%95%E4%B8%8B%E7%9A%84tomcat-users.xml%EF%BC%8C%E4%BF%AE%E6%94%B9%E6%88%90%E7%B1%BB%E4%BC%BC%E4%B8%8B%E9%9D%A2%E7%9A%84%EF%BC%9A)页面，我们需要修改$TOMCAT\_HOME/conf目录下的tomcat-users.xml：

<tomcat-users>

<role rolename=*"tomcat"*/>

<role rolename=*"manager"*/>

<role rolename=*"manager-gui"*/>

<role rolename=*"manager-script"* />

<role rolename=*"admin-gui"*/>

<user username=*"tomcat"* password=*"tomcat"* roles=*"tomcat,manager,manager-gui,manager-script,admin-gui"* />

</tomcat-users>

2.2、修改pom.xml增加Tomcat Maven插件

   我使用的Tomcat7，pom.xml中增加如下配置：

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<spring.version>3.2.2.RELEASE</spring.version>

<finalName>web-loab</finalName>

</properties>

<build>

<plugins>

<plugin>

<groupId>org.apache.tomcat.maven</groupId>

<artifactId>tomcat7-maven-plugin</artifactId>

<configuration>

<url>http://localhost:8080/manager/text</url>

<server>tomcat</server>

<username>tomcat</username>

<password>tomcat</password>

<path>/${finalName}</path>

</configuration>

</plugin>

</plugins>

</build>

上面username、password来自tomcat-users.xml。server是Tomcat服务器名称。path是访问应用的路径。url指定Tomcat管理页路径。

2.3、修改Maven的settings.xml

   在$USER\_HOME/.m2目录下找到settings.xml，添加server节点：

<servers>

<server>

<id>tomcat</id>

<username>tomcat</username>

<password>tomcat</password>

</server>

</servers>

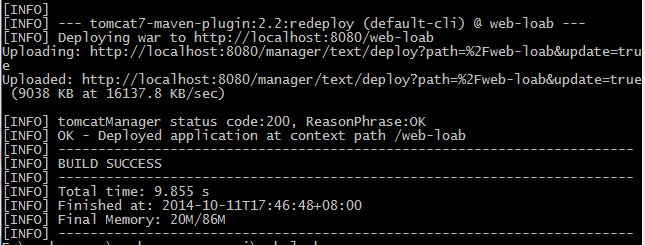
上面的username、password依然与tomcat-users.xml中相同，id与2.2中的server相同

2.4、部署项目到Tomcat

    先确保Tomcat服务器已经启动，然后cd到项目根目录，运行下面的命令：

mvn clean tomcat7:redeploy

部署成功，如下：



    然后我们就能在Tomcat安装目录的webapps目录下找到新部署的WAR包。

    初次部署用的是tomcat7:deploy命令，重新部署可以用tomcat7:redeploy命令（推荐统一用这个），Tomcat Maven插件支持的命令包括：run、shutdown、run-war-only、exec-war、standalone-war-only、deploy、standalone-war、undeploy、run-war、redeploy等。