# 各种网址

官方下载 <http://maven.apache.org/download.html>

Maven仓库

|  |
| --- |
| <http://repo1.maven.org/>  <http://repo2.maven.org>  上面两个地址已换成  <http://search.maven.org/#browse> |
| 最新仓库地址:  <http://mirrors.ibiblio.org/pub/mirrors/maven2/> |

# 安装

## 安装maven

|  |
| --- |
| 解压到一个盘下 D:\apache-maven-2.2.1\bin; 然后把maven的bin目录配置到path中  用mvn -v测试一下安装是否成功;  注意: 使用maven, JDK的配置不能是D:\jdk1.6.0\_11这样的目录, 应该是D:\jdk |

## 安装eclipse插件:m2clipse

## 修改mvn tomcat:run运行内存

在系统环境变量添加:

|  |
| --- |
| MAVEN\_OPTS="-Xmx512m -XX:MaxPermSize=256m" |

# maven各种命令

|  |  |
| --- | --- |
| maven pacage | 编译打包 |
| mvn test | JUnit测试 |
| java -cp target\helloworld-1.0-SNAPSHOT.jar org.marshal.App | 运行main方法 |
| mvn eclipse:eclipse | 使项目支持eclipse |

|  |  |
| --- | --- |
| mvn archetype create | 创建 Maven 项目 |
| mvn compile | 编译源代码 |
| mvn test-compile | 编译测试代码 |
| mvn test | 运行应用程序中的单元测试 |
| mvn site | 生成项目相关信息的网站 |
| mvn clean | 清除目标目录中的生成结果 |
| mvn package | 依据项目生成 jar 文件 |
| mvn install | 在本地 Repository 中安装 jar |
| mvn eclipse eclipse | 生成 Eclipse 项目文件 |
| mvn compiler:compiler | 编译（如果编译不通过，就直接用eclipse编译，F5） |
| mvn install -Dmaven.test.skip=true | 打包，不将测试代码打包 |
| mvn tomcat:run | 不改变端口启动web项目 |
| mvn tomcat:run -D maven.tomcat.port=8001 | 不改变端口启动web项目(可同时启动多个) |
|  |  |

# 新建工程

## 用mavne命令新建java工程和web工程

|  |
| --- |
| 创建java工程  mvn archetype:create -DgroupId=com.broadin.community -DartifactId=community-model |
| 创建web工程  mvn archetype:create -DgroupId=com.broadin.community -DartifactId=community-internet-webapp -DarchetypeArtifactId=maven-archetype-webapp |
| 把刚创建的工程导入eclipse中:  方法一: 在项目中执行mvn -Dwtpversion=1.0 eclipse:eclipse(为该项目增加eclipse支持), 才可以在eclipse中导入java工程;  方法二: 也可以在eclipse中安装maven插件, 然后导入maven工程. |
| test: 创建社区工程  mvn archetype:create -DgroupId=com.broadin.community -DartifactId=community-util  mvn archetype:create -DgroupId=com.broadin.community -DartifactId=community-model  mvn archetype:create -DgroupId=com.broadin.community -DartifactId=community-dao  mvn archetype:create -DgroupId=com.broadin.community -DartifactId=community-service  mvn archetype:create -DgroupId=com.broadin.community -DartifactId=community-internet-webapp -DarchetypeArtifactId=maven-archetype-webapp  mvn archetype:create -DgroupId=com.broadin.community -DartifactId=community-manager-webapp -DarchetypeArtifactId=maven-archetype-webapp |

mvn archetype:create -DgroupId=com.ipub -DartifactId=ipub-webapp -DarchetypeArtifactId=maven-archetype-webapp

## 用带m2clipse插件的myeclipse创建maven工程

|  |
| --- |
| File-->new--other |

# eclipse 3.6.1 安装maven插件失败的解决办法

|  |
| --- |
| eclipse 3.6.1 安装maven插件失败的解决办法  文章分类:软件开发管理  一、eclipse 3.6.1下载地址[eclipse-jee-helios-SR1-win32.zip]  http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/heliossr1  二、插件地址  1、gef 插件地址：  http://download.eclipse.org/tools/gef/updates/interim/  2、subclipse 插件地址：  http://subclipse.tigris.org/update\_1.6.x  3、maven2 插件地址：  m2eclipse-core Update Site: http://m2eclipse.sonatype.org/sites/m2e  m2eclipse-extras Update Site: http://m2eclipse.sonatype.org/sites/m2e-extras  注意 ：必须两个插件都安装，不安装extras 插件时，点击“run as server”来运行项目时会报错：“the selected did not contain any resources that can run on server maven project”  三、安装时出现问题：  直接在线安装maven2 会出现依赖插件找不到的问题，无法安装。必须先安装gef 插件后才能安装m2eclipse-core 插件，然而安装m2eclipse-extras 插件又依赖subclipse 插件。所以，三个插件的正确的安装顺序是：gef插件 》subclipse插件 》m2eclipse插件。  m2eclipse-core插件 ——依赖——＞gef插件  m2eclipse-extras插件 ——依赖——＞subclipse插件    安装完成后，启动eclipse,你将还会看到如下提示：  安装m2eclipse插件后，重启Eclipse老是提示：  Eclipse is running in a JRE, but a JDK is required  Some Maven plugins may not work when importing projects or updating source folders.    -vm配置项必须在 –vmargs配置项前，是因为执行到-vmargs配置项已经选择了一个默认的JVM了。  所以在–vmargs前面加上这一句：  -vm  D:/Java/jdk1.6.0\_10/bin/javaw.exe |

# pom文件详解

## pom大纲

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd"*>  <modelVersion>4.0.0</modelVersion> 当前POM模型的版本, 对于maven 2及maven 3来说, 它只能是4.0.0  <groupId>com.ismp</groupId> 项目属于哪个组, 例如:com.xjgz(组织或公司名).ismp(项目名)  <artifactId>ismp</artifactId> 当前maven工程在组中的唯一ID, 也就是工程名, 模块名  <version>1.0-SNAPSHOT</version> 版本(SNAPSHOT开发中, SNAPSHOT快照:开发中不稳定的版本)  <packaging>pom</packaging> 包  <name>\*\*\* ISMP Project \*\*\*</name> 名称, 随意起, 只供人工查看, 程序不识别  <url>/ismp/</url>  <inceptionYear>2010</inceptionYear> 开始时间  <organization> 开发组织(只供人工查看, 程序不识别)  <name>新疆公众信息产业股份有限公司北京研发中心</name> 公司名称  <url>http://www.xjgzinfo.com</url> 公司主页地址  </organization>  <repositories>  <repository>  <id>ibiblio-repository</id>  <name>ibiblio-repository</name>  <url>http://mirrors.ibiblio.org/pub/mirrors/maven2</url>  </repository>  <repository>  <id>ws.zones.apache.org</id>  <name>ws.zones.apache.org</name>  <url>http://ws.zones.apache.org/repository</url>  </repository>  </repositories>  <build>  <pluginManagement>  <plugins>  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-compiler-plugin</artifactId>  <inherited>true</inherited>  <configuration>  <source>1.5</source>  <target>1.5</target>  </configuration>  </plugin>  </plugins>  </pluginManagement>  </build>  <reporting>  <plugins>  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-javadoc-plugin</artifactId>  <configuration>  <!-- <aggregate>true</aggregate> -->  <minmemory>128m</minmemory>  <maxmemory>512m</maxmemory>  <links>  <link>http://java.sun.com/javase/6/docs/api/</link>  <link>http://www.hibernate.org/hib\_docs/v3/api/</link>  <link>http://ldaptemplate.sourceforge.net/apidocs/</link>  <link>http://static.springframework.org/spring/docs/2.0.x/api/</link>  </links>  </configuration>  </plugin>  </plugins>  </reporting>  <modules> 组件/模块  <module>libs</module> 子pom.xml文件的artifactId  <module>wars</module>  </modules>  <dependencyManagement> 依赖管理  <dependencies>  <dependency> 依赖, 可以有一个或多个依赖, 比如添加spring, hibernate等  <groupId>org.hibernate</groupId>  <artifactId>hibernate</artifactId>  <version>3.2.1.ga</version>  <exclusions> 排除  <exclusion>  <groupId>commons-collections</groupId>  <artifactId>commons-collections</artifactId>  </exclusion>  </exclusions>  </dependency>  </dependencies>  </dependencyManagement>  <dependencies> 依赖(一般都是测试依赖)  <dependency>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  <version>4.8.1</version>  <scope>test</scope>  </dependency>  </dependencies>  <properties> 属性  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding> 工程源码编码格式  <spring.version>2.5.6</spring.version> spring的版本  <commons-logging.version>1.1</commons-logging.version> commons-logging.version的版本  </properties>  </project> |
|  |

## junit

|  |
| --- |
| <dependencies>  <dependency>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  <version>4.8.1</version>  <scope>test</scope>  </dependency>  </dependencies> |

|  |
| --- |
| <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  <spring.version>2.5.6</spring.version>  <commons-logging.version>1.1</commons-logging.version>  </properties> |

mvn [options] []

-e 异常信息

-X 全部信息

-f 指定pom

-D 设置变量, 比如设置不测试

# MVN:指定编译时的文字编码和语法级别

|  |
| --- |
| 方法1  <build>  <defaultGoal>package</defaultGoal>  <plugins>  <plugin>  <artifactId>maven-compiler-plugin</artifactId>  <configuration>  <source>1.6</source>  <target>1.6</target>  <encoding>UTF-8</encoding>  </configuration>  </plugin>  </plugins>  </build> |
| 方法2  <properties>  <maven.compiler.encoding>UTF-8</maven.compiler.encoding>  <maven.compiler.target>1.6</maven.compiler.target>  <maven.compiler.source>1.6</maven.compiler.source>  </properties> |

web工程的pom.xml配置

|  |
| --- |
| jsp |
| servlet |
|  |
| log4j |
| junit |
| 配置文件 |

Spring

|  |
| --- |
| core  aop |

Hibernate

|  |
| --- |
| 数据源 |
| 连接池 |
|  |

# 使用maven, 在eclipse中配置debug详解

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **启动web应用** | 1。      **tomcat:run -D maven.tomcat.port=8007**    **这两个参数是：**   |  |  | | --- | --- | | **name** | **value** | | MAVEN\_BATCH\_PAUSE | on | | MAVEN\_OPTS | -Xdebug -Xnoagent -Djava.compiler=NONE -Xrunjdwp:transport=dt\_socket,address=8000,suspend=n,server=y -Xms256m -Xmx512m -XX:MaxPermSize=128m -Dfile.encoding=UTF-8 |   =  = |
| **配置debug** |  |