

本篇文章给大家谈谈mavencenter，以及对应的知识点，致力于为用户带来全面可靠的币圈信息，希望对各位有所帮助！

maven打包的时候默认是不加入依赖的jar包的，所以想打出一个独立的可运行jar包的话直接mvn clean install package是不行的。需要略改动下pom文件，加入如下plugin

build

```
sourceDirectorysrc/main/java/sourceDirectory
```

```
plugins
```

```
plugin
```

```
groupIdorg.apache.maven.plugins/grogroupId
```

```
artifactIdmaven-compiler-plugin/artifactId
```

```
configuration
```

```
defaultLibBundleDirlib/defaultLibBundleDir
```

```
source1.5/source
```

```
target1.5/target
```

```
encodingUTF-8/encoding
```

```
/configuration
```

```
/plugin
```

```
plugin
```

```
groupIdorg.apache.maven.plugins/grogroupId
```

```
artifactIdmaven-jar-plugin/artifactId
```

configuration

archive

manifest

addClasspathtrue/addClasspath

classpathPrefix/classpathPrefix

mainClasscom.xx.xx.xx/mainClass

/manifest

/archive

/configuration

/plugin

plugin

groupIdorg.apache.maven.plugins/groupId

artifactIdmaven-dependency-plugin/artifactId

executions

execution

idcopy/id

phaseinstall/phase

goals

goalcopy-dependencies/goal

/goals

configuration

outputDirectory

\${project.build.directory}

/outputDirectory

/configuration

/execution

/executions

/plugin

plugin

groupIdorg.apache.maven.plugins/groupId

artifactIdmaven-resources-plugin/artifactId

version2.2/version

configuration

encodingUTF-8/encoding

/configuration

/plugin

/plugins

/build

maven打包的时候有些jar包是不在mavencenter的。需要在pom中引入相关的本地jar包，那么相关的dependency应该像如下更改

dependency

groupIdorg.wltea.ik-analyzer/groupId

artifactIdik-analyzer/artifactId

version3.2.8/version

scopesystem/scope

systemPath\${project.basedir}/lib/ik-analyzer-3.2.8.jar/systemPath

/dependency

对于war包系统来说，有时候需要把jar包打到相关的war包中，可以使用plugin，默认将lib下的所有jar文件打包到WEB-INF/lib下。当然也是可以打包其他的文件的，诸如xml,properties等的。相关的plugin如下：

plugin

groupIdorg.apache.maven.plugins/groupId

artifactIdmaven-war-plugin/artifactId

version2.3/version

configuration

warName\${project.artifactId}/warName

webResources

resource

directorylib//directory

targetPathWEB-INF/lib/targetPath

includes

```
include**/*.jar/include
```

```
/includes
```

```
/resource
```

```
/webResources
```

```
/configuration
```

```
/plugin
```

maven项目移除MavenDependencies后如何在添加进去

昨天一不留神，把Maven

Dependencies给移除掉了。一阵痛苦...！

1：找到工程的上一个版本

2：直接替换你的.classpath文件

3：找到工程-右键-Maven-Update

Project... Alt+F5

4：找到工程-右键-Build Path-Configure Build Path...在

LiBraries列表中 是不是已经看到了Maven Dependencies。

如果你的工程已经没有了上一个版本

1：打开.classpath文件

在你的.classpath文件中加入上文中蓝色背景部分就可以了

上面的代码前面都加行号是因为新浪博客把代码直接显示了，避免这个问题出现。

已投稿到：

以上网友发言只代表其个人观点，不代表新浪网的观点或立场。Eclipse使用Maven构建web项目 – Java – ITeye技术网站

博客分类：

1.创建Maven项目：

点击“File”菜单，或者通过工具栏的“New”创建Project，如下图所示：

选择Maven-》Maven Project，弹出向导对话框，如下图所示：

选中Create a simple project.....复选框，其它的设置不变，并点击Next，输入maven项目必须的信息（groupId，artifactid，version），如下图所示：

这里需要选择Packaging，web项目一般选择war即可。点击Finish，完成项目创建，下图为创建的项目结构（项目名称为：sample-webapp）：

2.配置web项目：

右键点击sample-webapp项目，依次点击Properties-》Project Facets-》Convert to faceted form...，如下图所示：

勾选Dynamic Web Module，并选择版本（这里使用2.5，由于3.0版本需要tomcat7），如下图所示：

点击OK，此时，项目结构树增加了WebContent文件夹，如下图所示：

将WebContent下的WEB-INF、META-INF文件夹copy到src/main/webapp目录下，并且删除WebContent目录，完成之后，如下图所示：

右键点击sample-webapp项目，依次点击Properties-》Deployment Assembly，如下图所示：

此时，通过Remove按钮删除/src/test/java、/src/test/resources、/WebContent，并且通过Add添加webapp与maven依赖，如下图所示：

首先Add-》Folder，选择webapp目录并Finish，再选择Java Build Path

Entries-》 Maven Dependencies

完成之后，如下图所示：

3.部署web项目：

添加tomcat server，并部署web项目，这里不描述，与非maven项目基本相同。

浏览 50198

不要这么麻烦有更好的话,可以共享哈

Dead\_knight

浏览: 864524 次

来自: 杭州

浏览量：169349

浏览量：28011

浏览量：147916

厉害了 困扰我大半个月的问题解决了

双向的时候谷歌提示“不接受您的登录证书，或者您可能没有提供登录 ...

不成功啊。启动有问题 Security It ...

写的太棒了，中国需要你这样的人才。Eclipse中创建Web Maven Project并部署到Tomcat中（Maven的Web Project加入另一个Maven Project依赖） – Josh\_Persistence – ITeye技术网站

博客分类：

大家都知道，在进行J2EE项目的开发过程中，在调试阶段如果只是修改了页面是不需要重启应用服务器的，比如不需要重启Tomcat。只需要在浏览器中进行页面刷新即可。其实之所以不用重启Tomcat等应用服务器，其根本原因是因为我们可以

在应用服务器的配置文件中设置虚拟目录，这样就可以知道web项目所在的目录，于是就可以省去打包、然后再重新发布到服务器的步骤。

为什么不需要重启呢？

在Eclipse中开发j2ee项目，所有的jsp文件都是放在WebRoot目录下面，用户需要把所有用到的jar包放到WEB-INF（WebRoot的子目录）目录下的lib目录下（这就是没有用maven管理的缺点，所有jar都要自己去找）；Eclipse编译时，会把java源代码产生的class文件放到WEB-INF目录的classes目录下。所以大家在应用服务器中指定虚拟目录时，直接指定到WebRoot目录就行了，这样在调试过程中，如果只修改了jsp文件，则不需要重启服务器，刷新页面，就可以看到修改后的效果。

一. 在没有用Maven管理项目时，在Eclipse中将一个项目发布到Tomcat是非常方便的。

如果是一个web项目，那么只需要在Eclipse中的Servers View中简单的添加（因为Eclipse能够检测到web项目），

选中想要添加的web项目，然后点击Finish按钮即可。

如果不是web项目，有两种解决方法：

第一种是也在Eclipse的Servers View中添加一个Server，然后直接点击Finish按钮。然后在Eclipse的Servers View中双击刚添加的Server，并点击Add External Web Module按钮，并直接找到该项目的绝对路径然后将其添加到Eclipse中，并指定path路径为src/main/webapp。  
C:\Users\sswang6\workspace\AutoStressTest\ASTWeb\src\main\webapp

第二种方法是通过如下步骤将一个非Web Project修改为Web Project：

右击工程选择properties—project facets

2. 点击链接“Convert to faccted from...”（注意，该链接只在第一次出现，即在点击过一次后，不再出现）然后选上Dynamic Web Module，如果Version是2.4，可把Version由2.4改为2.5。（注意如果你当前是使用Tomcat6及Tomcat6以前的版本，不能使用Version3.0，因为Version3.0支持的JDK是7.0以上的版本，Tomcat6不支持）

然后点击下面的Further configuration available...，弹出Modify Faceted Projec



t窗口此处是设置web.xml文件的路径,输入src/main/webapp,该操作会在src/main/webapp中创建一个新的web.xml文件,即该目录为web项目的虚拟目录.然后直接和前面提到的操作一样,只需要在Eclipse的Servers view中增加一个tomcat server,并直接将web moudle加到该工程中.(注:Further configuration available...链接也是再点击一次后,以后就不会再出现。)

## 二. 在用Maven管理j2EE项目时

在用Maven管理项目时,也可以使用如上提到的方法.但有人会问,在Eclipse中,如何创建一个Maven的Web项目?如果多个Maven项目之间有依赖,如何在部署到Tomcat的时候将互相依赖的项目也部署进去?

问题1: 如何在Eclipse中创建一个Maven的web项目。(为了让更多的人了解怎么创建有相互依赖关系的Maven项目,即有父子关系,下面以一个这样的例子为例说明)

Eclipse中安装maven插件: m2e -

### 2. 创建一个maven 的web 项目:

使用eclipse直接创建maven的web项目:

1): 在eclipse建立一个maven项目,弹出窗口选着webapp, 点击next, 输入相关的信息就行. 在下面的对话框中选择maven-archetype-webapp. 我们假定该项目为parent项目, 注意parent项目的packaing必须是pom,如果是war, 因为我们这个地方创建的是一个web项目, 默认的packaging方式是war, 所以需要将其改为pom

groupIdcom.wsheng/groupId

artifactIdwsheng-spring-base/artifactId

packagingpom/packaging

2) 在有的Eclipse版本中, 可能需要自己手动创建如下3个源文件: 分别为: src.main.java,src.test.java和src.test.resources

3) 在第一步结束后, 你会发现现在Eclipse的工作区间中多了一个maven的project项目: wsheng-spring-base。

4) 右击wsheng-spring-base，选择创建一个Maven的module

输入module名字为wsheng-spring-mvc, 同1)一样，选择创建一个maven-archetype-webapp的web项目。

5) 创建结束后，wsheng-spring-base和wsheng-spring-mvc都是maven的web项目，当然你知道可以在Eclipse的Servers View中新一个Server，你会发现，Eclipse中你可以直接部署这两个项目到Tomcat中，所以他们也就是Web项目了。这种方法是最简单的方法。直接可以在Eclipse中修改页面文件，会自动部署到Tomcat中而不用重启Tomcat。

特别需要注意的是，不要执行mvn eclipse:eclipse命令，执行该命令后，会将一个maven的project转化为一个eclipse的project，即此时在eclipse中就不再是一个maven的web project了，就不能直接部署到Tomcat中了。

问题二：如果我们在Eclipse中创建的项目不是像上面一样创建Maven的Web项目，而是普通的项目，怎样将其部署到Tomcat中？

注：该方法显然很麻烦，直接在Eclipse中创建一个Maven的Web项目是多么的简单，但介绍该方法主要是为了让大家都明白Eclipse中部署项目到Tomcat的本质所在。有兴趣的朋友不妨一试。

1. 可以参照上面的《第二种方法是通过如下步骤将一个非Web Project修改为Web Project：》现将一个非maven的web项目转化为一个maven的web项目。如现在有一个testMaven的project我们经过转换，将其转化为一个Maven的Web项目后。

2. 创建的几个java类和一个abc.jsp页面进行测试。

(其中target目录是用来存放项目打包之后生成的文件的目录，图一中显示target目录下并没有文件，是因为我们还没有用maven对项目进行打包。)

运行打包命令：mvn clean package 【对项目进行打包】

运行此命令后，项目的结构如图二：

执行上面的命令后你会发现图二中target目录下多出了几个子目录和文件，这是因为运行命令mvn clean package后，生成的文件默认会放在target目录下。其中testMaven.war文件可以直接拷贝到应用服务器如Tomcat的发布目录下，启动服务

器可以直接运行！但在开发阶段，相信不会有人这么调试web项目！因为这样不停的拷贝到Tomcat中太麻烦了！！！

其实如果你认真观察，你会发现生成的子目录testMaven也是一个完整的web应用，展开后如图三：

根据前面所说的，如果我们能够制定一个web项目的虚拟目录，直接将其部署到Tomcat中即可。在此例子中虚拟目录应该是“D:/EclipseWorkspace/testMaven/target/testMaven”。但这样的话，有一个缺点，那就是每次只要修改了项目中的文件，即使是jsp文件或者静态页面，你都要运行mvn clean package命令重新打包；要不然你的修改就不会更新到target目录下的文件中去！这样的话感觉比较麻烦，万一忘了执行此命令就刷新页面，发现没有看到修改后的效果，找问题都不好找！因为maven遵循“约定优先于配置”的思想。所以一般所有的页面都是放在目录webapp下面的，就我这例子而言也是如此！图二和图三都是执行了mvn clean package命令之后的项目的目录结构，认真观察可以发现在webapp的子目录WEB-INF下面除了页面的源文件，并没有classes目录和lib目录！要是有的话，那直接把虚拟目录指定到此webapp目录下即“D:/EclipseWorkspace/testMaven/src/main/webapp”那该多好，就不用每次修改了页面都需要运行命令mvn clean package之后才能看到效果！

对于这个问题是可以解决的，由于maven遵循“约定优先于配置”的思想，所以如果不做特殊配置，它约定默认是把打包之后产生的文件都放在target目录下的。要想改变此默认行为，我们修改在项目中的pom.xml文件。在此项目中，添加如下配置：

!- 更改maven默认的打包目录：将class文件和lib目录打包放到指定的目录 —

```
artifactIdmaven-war-plugin/artifactId
```

```
configuration
```

```
webappDirectory${basedir}/src/main/webapp/webappDirectory
```

```
warSourceDirectory${basedir}/src/main/webapp/warSourceDirectory
```

```
/configuration
```

```
/plugins/build
```

4. 在Pom文件中增加如上命令后运行打包命令mvn clean package，项目的目录结构图如图四：

从图四可以看到，webapp的子目录WEB-INF下面多出了classes和lib目录。这样的话，如果你只修改了页面，就不需要再执行mvn clean package命令对项目进行打包了（因为页面的源文件都直接在webapp目下的），直接享受修改页面即可刷新页面的快捷了（但是如果修改的是java文件，则还是需要执行mvn clean package命令重新打包的；同理每次修改了pom.xml文件也是一样，即也需要重新执行mvn clean package）！

总之：该方法一旦启动tomcat后，如果只是jsp页面的变化，不需要重新运行mvn clean package命令，而如果是修改了java代码后，不需要将tomcat停掉，只需要运行 mvn clean package命令即可。会自动部署。

附：不使用Eclipse怎样创建Maven的项目？

使用命令行：`mvn archetype:create -DgroupId=my.maven.test -`

`DartifactId=testMaven`

`-DarchetypeArtifactId=maven-archetype-webapp`

该篇文章的另一个重要的主题：如何在Eclipse中将两个有Dependency的Maven Project关联起来：

比如现在有一个maven web项目：storeweb, 该项目可以直接部署在Tomcat中，但该项目本身又直接依赖另一个maven 项目，如store\_rest项目，该项目为storeweb项目为store\_rest项目提供Rest风格的API. 那么在将storeweb项目部署到Tomcat的时候，同时也需要将store\_rest部署到Tomcat中，可以使用下面的方法：分为2步，第一步是Add maven Dependencies,第二步是Add project dependencies.

#### 4.1 Add maven dependencies

Right click the web project “properties” —— click the “Deployment Assembly” ——

Click add button

4.2 Choose Java Build Path Entire — click Next

4.3 Choose Maven Dependencies——

Add project dependencies , Click add button

4. 5 : Choose Project—— Click Next button

Choose your dependencies project —— Click Finish button

浏览 21009

Josh\_Persistence

浏览: 721787 次

来自: 上海

VCenter、ESXServer、Cluster这些实体类在 ...

感谢楼主分享

感觉博主的图有点问题吧~觉得暂存区到工作区应该是git res ...

这两天正在学习solrcloud , 相当受用。发现solrclo ...

请教一下 : 1我的Solr安装在d:solr-5.4. ...Maven Repository: org.mongodb  
mongo-java-driver

mongo-java-driver

The MongoDB Java Driver uber-artifact, containing mongodb-driver,  
mongodb-driver-core, and bson

LicenseApache 2.0CategoriesTagsUsed

ByVersionRepositoryUsagesDate3.4.x(Jan, 2017)(Dec, 2016)(Nov, 2016)(Oct,  
2016) 0

(Sep, 2016)3.3.x(Jul, 2016)3.2.x(Feb, 2016)(Jan, 2016)(Dec, 2015)(Nov,

2015)3.1.x(Nov, 2015)(Oct, 2015) 0

(Sep, 2015)3.0.x(Sep, 2015)(Jul, 2015)(May, 2015)(May, 2015)(Mar, 2015) 0

(Mar, 2015) 0

(Mar, 2015) 0

(Feb, 2015) 0

(Feb, 2015) 0

(Jan, 2015)2.14.x(Jul, 2016)(Mar, 2016)(Jan, 2016)(Dec, 2015) 0

(Nov, 2015)2.13.x(Sep, 2015)(May, 2015)(Apr, 2015)(Jan, 2015)(Jan, 2015)(Dec, 2014)(Nov, 2014)2.12.x(Jan, 2015)(Oct, 2014)(Jul, 2014)(May, 2014)(Apr, 2014)(Apr, 2014)(Mar, 2014)(Mar, 2014)(Mar, 2014)(Feb, 2014)2.11.x(Jan, 2014)(Sep, 2013)(Jun, 2013)(Apr, 2013)(Mar, 2013) 0

(Mar, 2013)2.10.x(Dec, 2012)(Nov, 2012)2.9.x(Oct, 2012)(Oct, 2012)(Sep, 2012)(Aug, 2012) 0

(Aug, 2012)(Aug, 2012)2.8.x(Jun, 2012)(May, 2012)2.7.x(Jan, 2012)(Nov, 2011)(Nov, 2011) 0

(Nov, 2011) 0

(Nov, 2011) 0

(Oct, 2011) 0

(Oct, 2011) 0

(Oct, 2011)2.6.x(Aug, 2011) 0

(Nov, 2011)(Jun, 2011) 0

(Nov, 2011) 0

(Nov, 2011)(Jun, 2011)(May, 2011)(May, 2011)2.5.x(Apr, 2011) 0

(Nov, 2011) 0

(Nov, 2011)(Mar, 2011) 0

(Nov, 2011) 0

(Mar, 2011)(Mar, 2011)2.4.x(Dec, 2010) 0

(Nov, 2011)2.3.x(Oct, 2010)2.2.x(Oct, 2010)2.1.x(Aug, 2010)2.0.x(Jun, 2010)(May, 2010) 0

(May, 2010) 0

(May, 2010) 0

(May, 2010)1.4.x(Apr, 2010)1.3.x(Mar, 2010)1.2.x(Dec, 2009) 0

(Dec, 2009)1.1.x(Dec, 2009)1.0.x(Dec, 2009)0.11.x 0

(Oct, 2009)0.9.x 0

(Oct, 2009)Related Books (2018)by Shannon Bradshaw, Kristina Chodorow (2016)by Krishna Rungta (2016)by AWS Whitepapers, Amazon Web Services (2016)by Max Lemann (2016)by Dan Warnock (2016)by Kyle Banker, Peter Bakkum, Shaun Verch, Doug Garrett, Tim Hawkins (2016)by Daniel Perkins (2016)by Cyrus Dasadia, Amol Nayak (2015)by Eelco Plugge, David Hows, Peter Membrey, Tim Hawkins (2015)by Exponential Mobile (2015)by

(2015)by Derek Rangel (2015)by Francesco Marchioni (2015)by sumit saraswat (2015)by Wilson da Rocha Franca (2015)by George Duckett (2014)by Peter Membrey, David Hows, Eelco Plugge (2014)by Amol Nayak (2014)by Brad Dayley (2014)by John Lynn (2014)by Afshin Mehrabani (2014)by Steve Hoberman (2014)by Anand Patil (2014)by Bo Borland (2013)by Kristina Chodorow (2013)by Rick Copeland (2011)by Kyle Banker (2011)by Kristina Chodorow (2010)by Kristina Chodorow, Michael Dirolf

Web site developed by

Powered by:

第一种方法：

Myeclipse10本身已经内置了maven的插件，咱们就不需要单独安装了，唯一需要修改的是Myeclipse10中使用的maven版本，不管 Myeclipse10使用的是哪个版本，咱们都统一改成刚才下载到本地安装的版本，方法是：

1.菜单依次进入：windows-Ppreferences-MyEclipse-Mave4MyEclipse-Maven-Installations

2.增加本地的maven到 Myeclipse10 中，：

maven2

3.修改setting.xml文件，：

maven3

到此，第一种方法完成了。

第二种方法：

下载最新的Maven

我这里安装时最新的3.0.5版本

MyEclipse10安装Maven的两种方法-解决

安装到电脑

1 将下载回来的包解压到任意地址，我的路径是D:mavenapache-maven-3.0.5

2 到环境变量中，这里我选择系统变量M2\_HOME 指向D:mavenapache-maven-3.0.5

3 在path尾部添加%M2\_HOME%bin

4检查是否安装成功



打开cmd

```
echo %M2_HOME%
```

```
mvn -v
```

:

MyEclipse10安装Maven的两种方法-解决

安装成功后就给myeclipse10安装该插件

打开myeclipse10

Help-MyEclipse Configuration Center-SoftWare

1.在My Software栏中，先删除Maven相关的东西，避免安装maven插件时引起冲突，这里删除“Maven Support”项。

2.点击add site(左边)，弹出对话框，分别填写：

Name:Maven（名字随意）

点击OK，myeclipse就开始下载maven插件，耐心等待片刻即可。

3.选择dashboard更新插件

MyEclipse10安装Maven的两种方法-解决

耐心等待片刻即可...!

最后检查是否安装成功：MyEclipse10安装Maven的两种方法-解决

好了，两种方法要那种自己选择吧！

相信经过小编对mavencenter和的介绍，你对mavencenter了解更加地透彻了，感谢你对我们地支持与关注！