<http://www.blogjava.net/zhb8015/articles/375116.html>

settings.xml设置localRepository如下代码所示：(地址可以自定义)

1 <localRepository>D:/Development/m2/repository</localRepository>

<http://www.cnblogs.com/quanyongan/archive/2013/04/17/3025971.html>

### **Using an Archetype**

To create a new project based on an Archetype, you need to call mvn archetype:generate goal, like the following:

1. mvn archetype:generate

Please refer to [Archetype Plugin page](http://maven.apache.org/archetype/maven-archetype-plugin/usage.html).

### **Provided Archetypes**

Maven provides several Archetype artifacts:

|  |  |
| --- | --- |
| **Archetype ArtifactIds** | **Description** |
| maven-archetype-archetype | An archetype to generate a sample archetype project. |
| maven-archetype-j2ee-simple | An archetype to generate a simplifed sample J2EE application. |
| maven-archetype-mojo | An archetype to generate a sample a sample Maven plugin. |
| maven-archetype-plugin | An archetype to generate a sample Maven plugin. |
| maven-archetype-plugin-site | An archetype to generate a sample Maven plugin site. |
| maven-archetype-portlet | An archetype to generate a sample JSR-268 Portlet. |
| maven-archetype-quickstart | An archetype to generate a sample Maven project. |
| maven-archetype-simple | An archetype to generate a simple Maven project. |
| maven-archetype-site | An archetype to generate a sample Maven site which demonstrates some of the supported document types like APT, XDoc, and FML and demonstrates how to i18n your site. |
| maven-archetype-site-simple | An archetype to generate a sample Maven site. |
| maven-archetype-webapp | An archetype to generate a sample Maven Webapp project. |

<http://blog.csdn.net/testcs_dn/article/details/36455669>

When error "javax.servlet.http.HttpServlet" was not found was fired.

It actually requires to import thre runtime server .jar file

Dependency scope 是用来限制Dependency的作用范围的, 影响maven项目在各个生命周期时导入的package的状态。

自从2.0.9后，新增了1种，现在有了6种scope:

* **compile**  
  默认的scope，表示 dependency 都可以在生命周期中使用。而且，这些dependencies 会传递到依赖的项目中。
* **provided**  
  跟compile相似，但是表明了dependency 由JDK或者容器提供，例如Servlet AP和一些Java EE APIs。这个scope 只能作用在编译和测试时，同时没有传递性。
* **runtime**  
  表示dependency不作用在编译时，但会作用在运行和测试时
* **test**  
  表示dependency作用在测试时，不作用在运行时。
* **system**  
  跟provided 相似，但是在系统中要以外部JAR包的形式提供，maven不会在repository查找它。 例如：

### **How do I create a JAR and install it in my local repository?**

Making a JAR file is straight forward enough and can be accomplished by executing the following command:

1. mvn package

If you take a look at the POM for your project you will notice the packaging element is set to jar. This is how Maven knows to produce a JAR file from the above command (we'll talk more about this later). You can now take a look in the${basedir}/target directory and you will see the generated JAR file.

Now you'll want to install the artifact you've generated (the JAR file) in your local repository (${user.home}/.m2/repository is the default location). For more information on repositories you can refer to our [Introduction to Repositories](http://maven.apache.org/guides/introduction/introduction-to-repositories.html) but let's move on to installing our artifact! To do so execute the following command:

1. mvn install

## **Guide to installing 3rd party JARs**

Although rarely, but sometimes you will have 3rd party JARs that you need to put in your local repository for use in your builds, since they don't exist in any public repository like [Maven Central](http://search.maven.org/). The JARs must be placed in the local repository in the correct place in order for it to be correctly picked up by Apache Maven. To make this easier, and less error prone, we have provide a goal in the [maven-install-plugin](http://maven.apache.org/plugins/maven-install-plugin/) which should make this relatively painless. To install a JAR in the local repository use the following command:

1. mvn install:install-file -Dfile=<path-to-file> -DgroupId=<group-id> \
2. -DartifactId=<artifact-id> -Dversion=<version> -Dpackaging=<packaging>

If there's a pom-file as well, you can install it with the following command:

1. mvn install:install-file -Dfile=<path-to-file> -DpomFile=<path-to-pomfile>

With version 2.5 of the maven-install-plugin it gets even better. If the JAR was built by Apache Maven, it'll contain a pom.xml in a subfolder of the META-INF directory, which will be read by default. In that case, all you need to do is:

1. mvn install:install-file -Dfile=<path-to-file>

<http://www.devnote.cn/article/193.html>

Run maven web on tomcat

<http://blog.sina.com.cn/s/blog_8ced01900101a4u7.html>

Does not import serverlet-api from build maven.

mvn tomcat:run -Dmaven.tomcat.port=8081