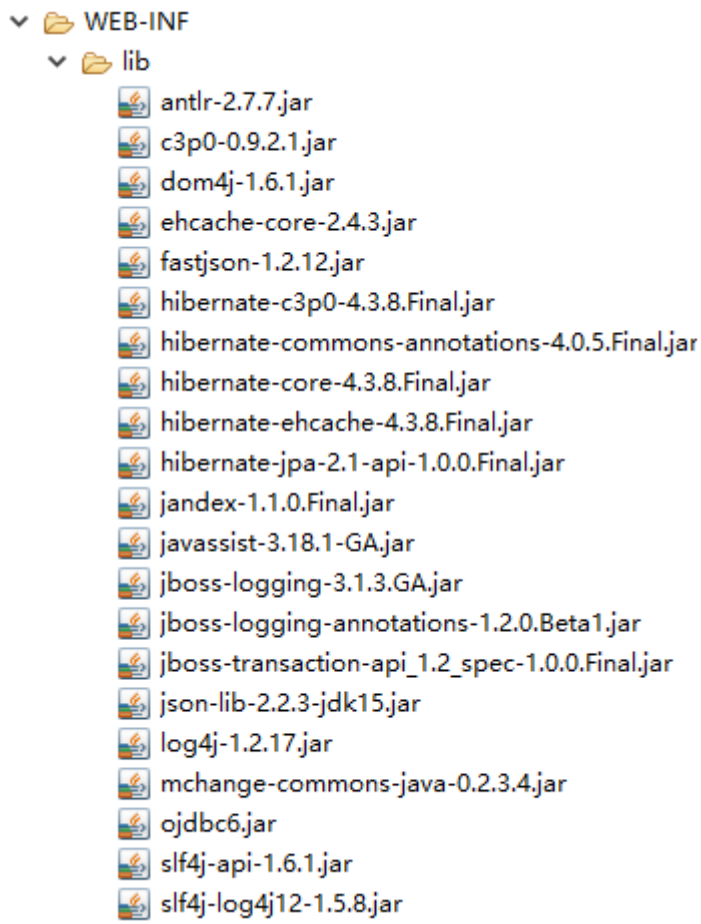
maven

ant -> maven -> gradle

核心

- 强大的构建工具
- 依赖管理[jar包文件]

传统方式



分析弊端:

- 找的jar包比较繁琐
- 造成磁盘容量冗余

- 不利于jar文件的管理[jar的升级,jar的版本]
- 部署的时候,影响了上传的速度.

仓库

- 本地仓库 - repo - 用来存放所有的jar包文件.
- 远程仓库 - 中央仓库[maven默认提供的],第三方平台提供的[阿里].

POM

pom.xml文件

```
<!--依赖的坐标-->
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>4.3.8.Final</version>
</dependency>
```

工作原理:

groupId,artifactId,version - 构成了一个依赖的坐标.

- 优先到本地仓库中进行寻找.
- 如果本地仓库中不存在,则到"远程仓库"中下载[默认的远程的maven的仓库],

  需要在setttings.xml文件中进行镜像的配置[推荐使用阿里的]

环境变量的配置

前提:JAVA_HOME得配置成功.

.../系统变量/... 新建:

变量名:MAVEN_HOME

变量值:D:\apache-maven-3.3.9

再新建

变量名:M2_HOME

变量值:D:\apache-maven-3.3.9

选中Path - 编辑 - 最左边:%MAVEN_HOME%\bin;其余目录

重启cmd,输入:mvn -v

镜像的配置

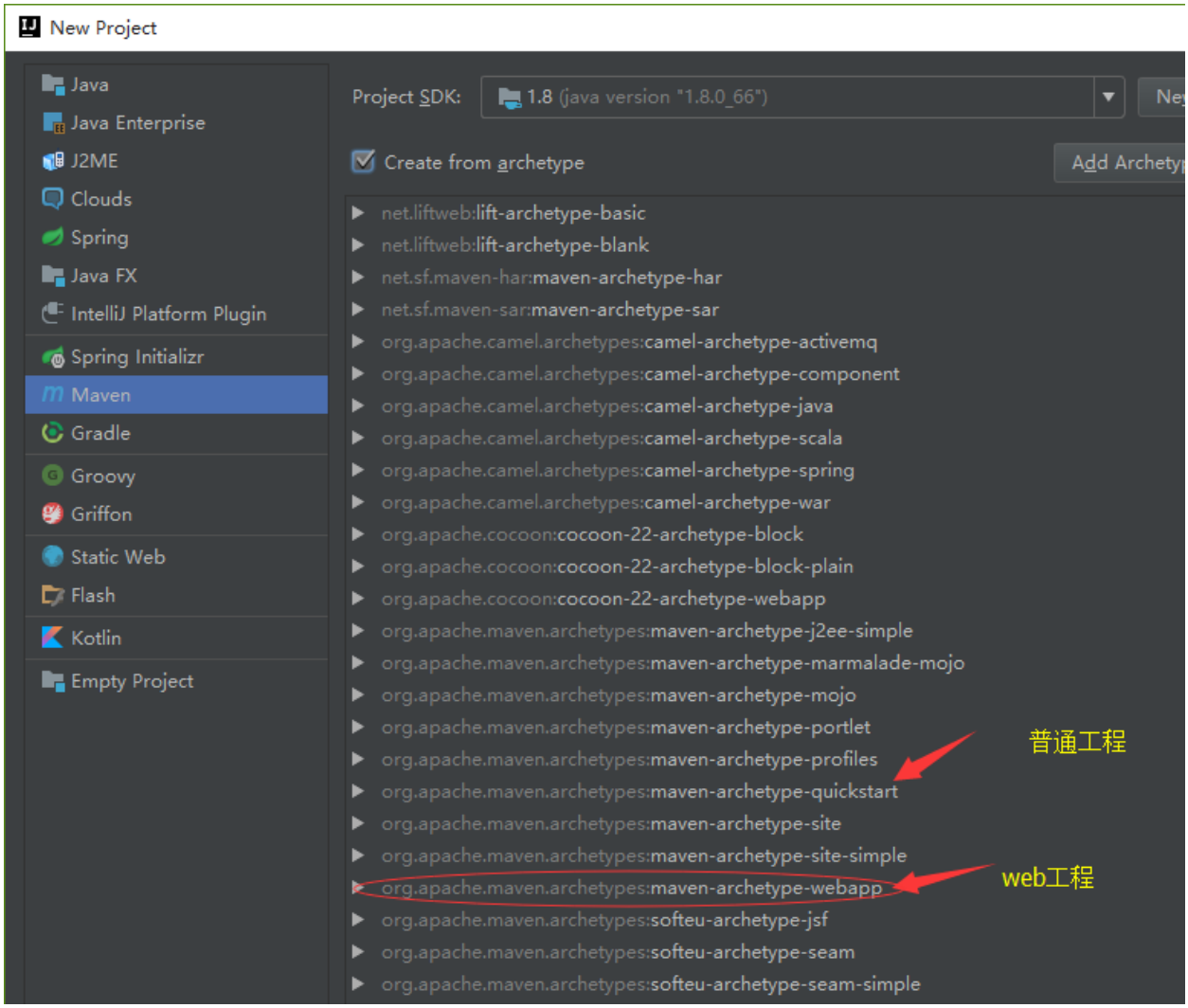
settings.xml

- 通过cmd->进入到C:\Users\Administrator
- 通过dos命令 -> md .m2

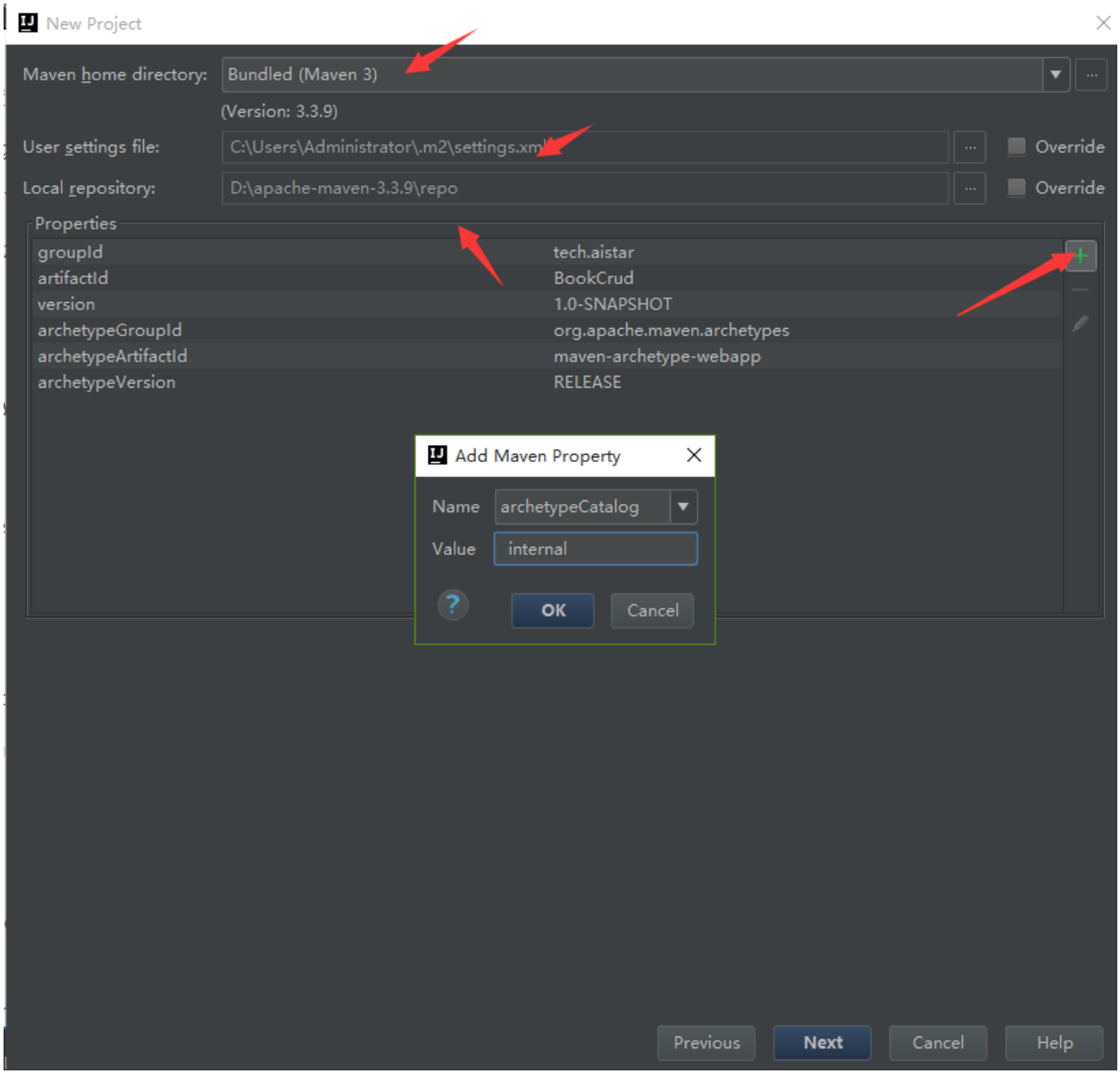
```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
http://maven.apache.org/xsd/settings-1.0.0.xsd">

  <!--指定本地仓库的地址-->
  <localRepository>D:\apache-maven-3.3.9\repo</localRepository>
  <mirrors>
      <!--国内镜像-阿里云服务器-->
      <mirror>
          <id>alimaven</id>
          <name>aliyun maven</name>

 <url>http://maven.aliyun.com/nexus/content/groups/public/</url>
          <mirrorOf>central</mirrorOf>
      </mirror>
  </mirrors>
</settings>
```

IDEA中创建maven的web工程

- 创建maven-web



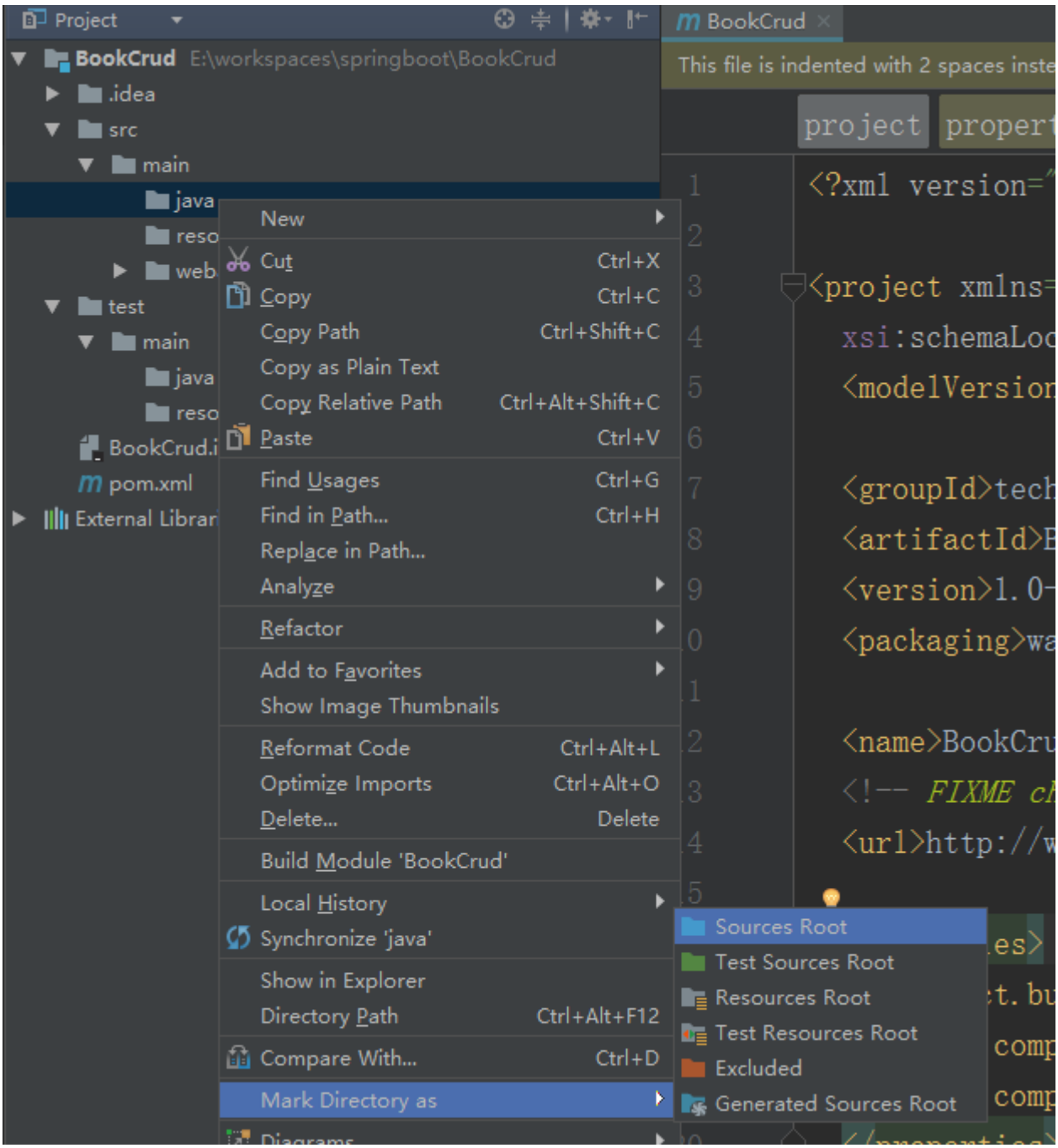
- 确认配置

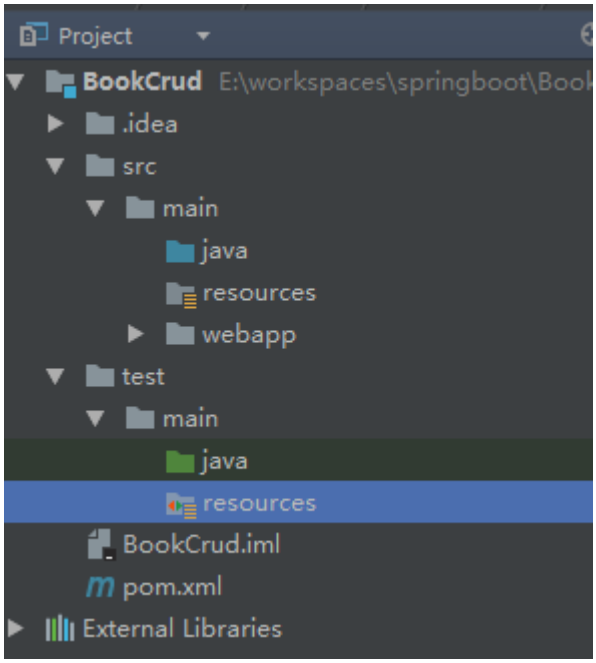


maven的工程结构

约定大于配置

```
工程名
   src/main/java - java的源代码
   src/main/resources - 源代码所需要的配置文件
   src/main/webapp - 存放jsp以及一些静态资源文件[html,css,javascript,图
片...]

   test/main/java - 放测试代码[单元测试]
   test/main/resources - 放测试需要的配置文件
```


Project
BookCrud E:\workspaces\springboot\BookCrud
.idea
src
main
java
reso
web
test
main
java
reso
BookCrud.i
pom.xml
External Librari
BookCrud
This file is indented with 2 spaces inste
project
propert
<?xml version="
<project xmlns=
xsi:schemaLoc
<modelVersion
<groupId>tech
<artifactId>B
<version>1.0-
<packaging>wa
<name>BookCru
<!-- FIXME cl
<url>http://w
New
Cut Ctrl+X
Copy Ctrl+C
Copy Path Ctrl+Shift+C
Copy as Plain Text
Copy Relative Path Ctrl+Alt+Shift+C
Paste Ctrl+V
Find Usages Ctrl+G
Find in Path... Ctrl+H
Replace in Path...
Analyze
Refactor
Add to Favorites
Show Image Thumbnails
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Delete... Delete
Build Module 'BookCrud'
Local History
Synchronize 'java'
Show in Explorer
Directory Path Ctrl+Alt+F12
Compare With... Ctrl+D
Mark Directory as
Diagrams
Sources Root
Test Sources Root
Resources Root
Test Resources Root
Excluded
Generated Sources Root



更改web.xml文件的dtd

```
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID"
version="3.0">

  <display-name>Archetype Created Web Application</display-name>
</web-app>
```

关于pom.xml

- maven的编译的插件

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>3.1</version>
    <configuration>
        <!--推荐maven的编译的版本和当前使用的jdk的版本保持一致-->
        <source>1.8</source>
        <target>1.8</target>
        <encoding>UTF-8</encoding>
    </configuration>
</plugin>
```

- tomcat插件

```
<plugin>
    <groupId>org.apache.tomcat.maven</groupId>
    <artifactId>tomcat7-maven-plugin</artifactId>
    <version>2.2</version>
    <configuration>
        <server>tomcat7</server>
        <port>8888</port>
        <uriEncoding>utf-8</uriEncoding>
        <url>http://localhost:8888/manager/text</url>
        <path>/BookCrud</path>
    </configuration>
</plugin>
```

启动服务器[进入项目所在的根目录中执行]:mvn tomcat7:run

解析url

浏览器:http://localhost:8888/BookCrud/index.jsp

http://192.168.5.150:8888/BookCrud/index.jsp

url - 统一资源[存放在服务器上的东西]定位器 - 资源的唯一标识.

1. http:超文本传输协议.

2. localhost[127.0.0.1] - 服务器所在主机的ip地址
3. 端口号8888 - 端口号和应用程序之间是1:1的关系.并且1000的端口号不推荐使用[系统占用了]

   为了确保找哪个服务器.每个服务器也是应用程序[每个应用程序的端口是不一样的]

4. BookCrud - 项目的名称 - 项目`的上下文路径
5. http://localhost:8888/BookCrud - 自动进入到项目的webapp目录