Maven介绍安装和命令创建

项目存在问题 jar包

▶ aspectjweaver-1.8.13.jar commons-dbcp-1.4.jar ▶ ■ commons-fileupload-1.3.jar commons-io-2.2.jar commons-logging-1.1.3.jar commons-pool-1.6.jar cxf-core-3.1.15.jar cxf-rt-bindings-soap-3.1.15.jar cxf-rt-bindings-xml-3.1.15.jar cxf-rt-databinding-jaxb-3.1.15.jar cxf-rt-frontend-jaxws-3.1.15.jar cxf-rt-frontend-simple-3.1.15.jar cxf-rt-transports-http-3.1.15.jar cxf-rt-ws-addr-3.1.15.jar cxf-rt-wsdl-3.1.15.jar ▶ javax.servlet.jsp.jstl.jar ▶ jstl-impl.jar log4j-1.2.17.jar mybatis-3.2.6.jar mybatis-spring-1.2.2.jar ▶ neethi-3.1.1.jar ojdbc14.jar ▶ spring-aop-4.3.0.RELEASE.jar ▶ ■ spring-beans-4.3.0.RELEASE.jar ▶ | spring-context-4.3.0.RELEASE.jar spring-core-4.3.0.RELEASE.jar ▶ | spring-expression-4.3.0.RELEASE.jar ▶ | spring-jdbc-4.3.0.RELEASE.jar spring-tx-4.3.0.RELEASE.jar spring-web-4.3.0.RELEASE.jar ▶ ■ spring-webmvc-4.3.0.RELEASE.jar stax2-api-3.1.4.jar ▶ | woodstox-core-asl-4.4.1.jar ▶ wsdl4j-1.6.3.jar xmlschema-core-2.2.3.jar

不熟悉jar包 iar包管理

Maven介绍

tomcat 工具 项目部署发布

maven工具 负责项目管理和构建工具 (项目创建 管理jar包)

Maven是Apache开源组织提供的一个基于POM(Project Object Model)的项目构建(Build)工具,所谓的构建指的是代码清除(clean)、编译(compile)、测试(test)、打包(package)、部署(deploy)等一系列流程。

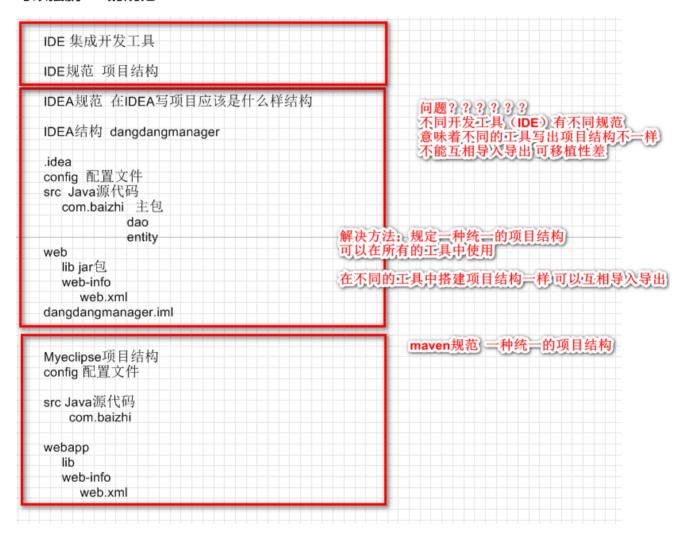
POM 项目对象模型 把项目当做对象来管理

代码清除(clean)删除编译后的项目(out文件夹)

部署 (deploy) 远程部署

Maven的优点

可以摆脱IDE的规范



通过统一的结构就摆脱了不同结构带来的不方便

通过命令快速构建项目

现在:IDEA中按钮

命令: java javac

Maven可以使用maven命令创建 通过黑窗口执行命令可以生成一个空项目

自动管理依赖!!!

依赖dependency 实际上指的jar包

管理jar包 主要是管理jar包关系(互相依赖)

struts-core.jar ----> xwork-core.jar struts-core需要使用 xwork-core中的方法和类 a---->b---->c-----d 需要使用a 必须导入bcd 自己找没法找

maven可以实现 程序员导入a maven自动寻找下载导入bcd 自动管理依赖

如何去寻找下载?

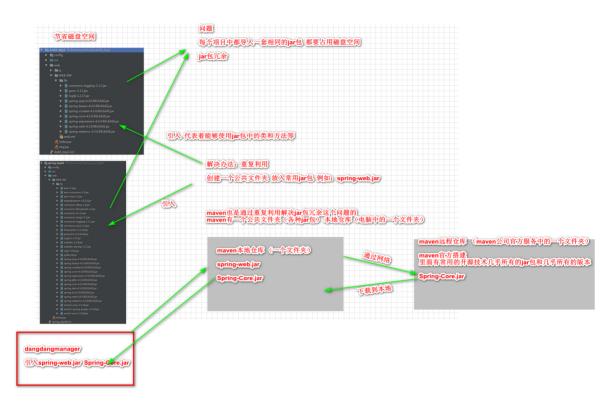
节省磁盘空间!!!



注意:maven使用的时候不导入引用jar包(意味着jar不需要复制到本项目中,但是可以使用jar包中的类和方法等)

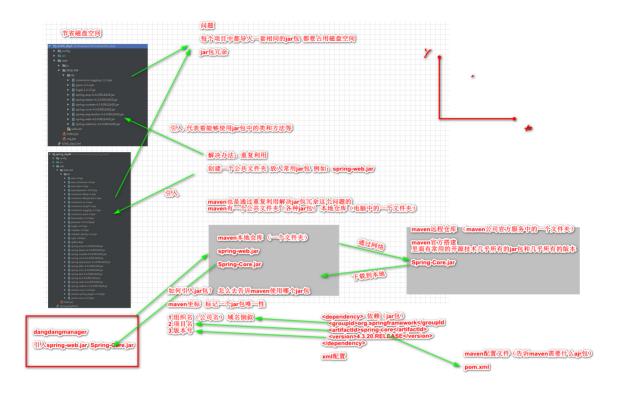
Maven相关使用原理

maven如何去寻找下载jar包?



maven工具通过网络去远程仓库中寻找并且下载jar包 下载到本地仓库

我们的项目如何引入maven中jar包?



在项目中的pom.xml中配置jar包的坐标信息可以让maven去下载引入相关的jar包

maven项目规范 项目结构 !!!

dangdangmaneger

src/main/java Java源代码

src/main/resources 配置文件

src/main/webapp web相关

src/test/java 测试源代码

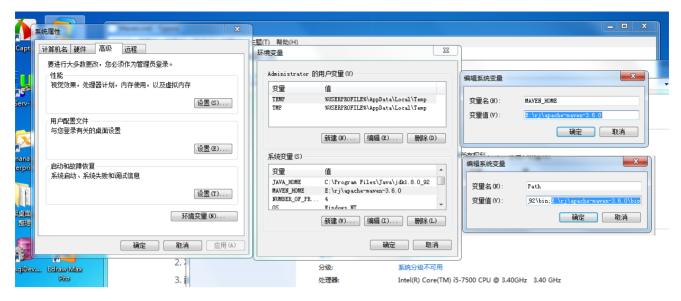
src/test/resources 测试相关配置文件

pom.xml maven核心配置文件

targer 源代码输入目录 编译后的代码在这儿

下载安装maven

- 1. 找到maven官网
- 2. 本地解压
- 3. 配置环境变量
 - 1. MAVEN_HOME 安装目录 E:\rj\apache-maven-3.6.0
 - 2. path bin目录 E:\rj\apache-maven-3.6.0\bin



4. 验证有没有安装配置成功

黑窗口中输入 mvn -version

```
E:\>mun -uersion
E:\
Apache Maven 3.6.0 (97c98ec64a1fdfee7767ce5ffb20918da4f719f3; 2018-10-25T02:41:4
7+08:00)
Maven home: E:\rj\apache-maven-3.6.0\bin\..
Java version: 1.8.0_92, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk1.8.0_92\jre
Default locale: zh_CN, platform encoding: GBK
OS name: "windows 7", version: "6.1", arch: "amd64", family: "windows"
E:\>
```

创建一个maven项目

通过命令创建

archetype 原型 骨架 项目结构(空项目) maven创建了很多常用的项目结构 Java项目 web项目 通过命令可以下载对应的空项目

maven-archetype-quickstart java空项目

maven-archetype-webapp web项目

```
mvn archetype:generate -DgroupId=组织名称 -DartifactId=项目名称 -
DarchetypeArtifactId=maven-archetype-quickstart

mvn archetype:generate -DgroupId=com.baizhi -DartifactId=dangdang -
DarchetypeArtifactId=maven-archetype-webapp
```

注意:骨架并不完善需要手动补全

作业

- 1. 回顾梳理maven优点
- 2. 记忆maven项目结构
- 3. 安装maven 配置环境变量
- 4. 使用maven命令创建项目 并且补全项目结构 包含test相关结构