刘昱江:527161772+班级号+中心+名字

# 什么是互联网架构

## 互联网中架构的问题

### 高并发

概念描述:在单位时间内用户的大量的访问请求

描述:

2017年淘宝的交易额1682亿元,5分22秒支付宝的支付峰值25.6万笔/s

数据库的处理速度 4200万/s

总结:如何抗击高并发,是电商行业中首先要考虑的问题.解决了高并发问题.才是电商的主要问题.

### 分布式

#### 分布式计算

概念:一项任务,交给多个服务器完成

例子:一项任务10小时.如果交给10个人共同完成.1个小时完成.

学科:大数据的应用. 数据的分析.为指导者提供数据的支持.

应用:医疗/就医诊断

#### 分布式系统

概念:根据**功能模块**的不同,将项目进行拆分,拆分为不同的**项目**

说明:传统的项目设计,将全部的模块写到一起,可能会出现一个模块出了问题.直接影响整个项目服务.

总结:**对外统一,对内独立**

### 集群搭建

概念:多台服务器,处理的是**相同的业务**.集群的搭建是抗击高兵发的有效的手段.

例子:如果面临了高并发,则开启多台服务器共同面对并发压力.如果集群设计不好很容易引发”**雪崩效应**”

解决:实现集群的高可用(HA)

### 海量数据

说明:随着行业的发展,数据量增速较快.如何实现快速的检索是当下面临的主要问题

问题:在大量的数据中如何快速的获取数据.是主要的问题

解决方案:

1. 数据库的优化:分库分表,优化sql,建立良好的索引,定期数据转存

## 京淘项目概述

### 说明

说明:京淘项目学习的是当下最为流行的技术,主要是解决如何抗击高并发,实现服务的集群的部署,以及服务与服务之间的架构优化问题.

### 技能点

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 知识点 | 重要程度 |  | 掌握程度 |
| 1 | 业务：商品分类、商品、前台、权限单点登录、购物车、订单、商品全文检索 | ★★★★★ | 1 | 熟练 |
| 2 | Spring、SpringMVC、MyBatis主流框架 | ★★★ | 1 | 熟练 |
| 3 | Maven 一键构建继承和聚合 | ★★ | 1 | 熟练 |
| 4 | PowerDesinger表设计及优化 | ★★★★ | 3 | 会用 |
| 5 | 富客户端EasyUI、KindEditor图文控件 | ★ | 1 | 会用 |
| 6 | RESTFul 访问方式 | ★★★★★ | 1 | 熟练 |
| 7 | 通用Mapper插件，自动生成调用代码 | ★ | 1 | 会用 |
| 8 | Nginx 负载均衡/反响代理 | ★★★★★ | 1 | 精通 |
| 9 | Tomcat集群 | ★★★ | 1 | 熟练 |
| 10 | Linux/CentOS 操作系统 | ★★★ | 1 | 熟练 |
| 11 | MySQL主从复制，Amoeba/mycat读写分离 | ★★★★★ | 3 | 精通 |
| 12 | Redis 缓存服务器，分片，哨兵，集群高可用 | ★★★★★ | 1 | 精通 |
| 13 | JSONP浏览器跨域 | ★★ | 2 | 会用 |
| 14 | HttpClient系统间调用 | ★★★ | 1 | 熟练 |
| 15 | 石英钟定时任务(timer) | ★ | 1 | 会用 |
| 16 | RabbitMQ 消息队列 | ★★★ | 3 | 熟练 |
| 17 | Lucene、Solr搜索 | ★★★★ | 3 | 熟练 |
| 18 | Jsoup爬虫技术 | ★★★★ | 3 | 熟练 |
| 19 | Dubbo框架 | ★★★★ | 4 | 熟练 |
| 20 | Docker容器技术 | ★★★★ | 4 | 熟练 |
| 21 | Druid阿里数据库连接池 | ★★★ | 4 | 熟练 |

# Jt环境搭建

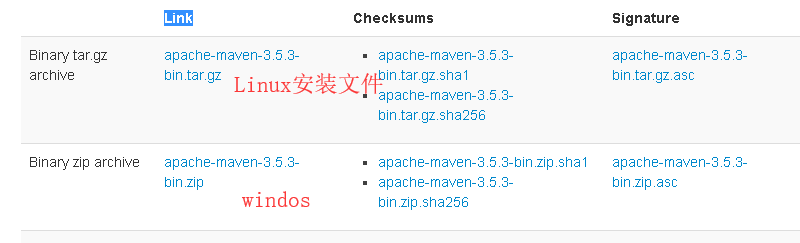
## Maven搭建

### Maven的主要功能

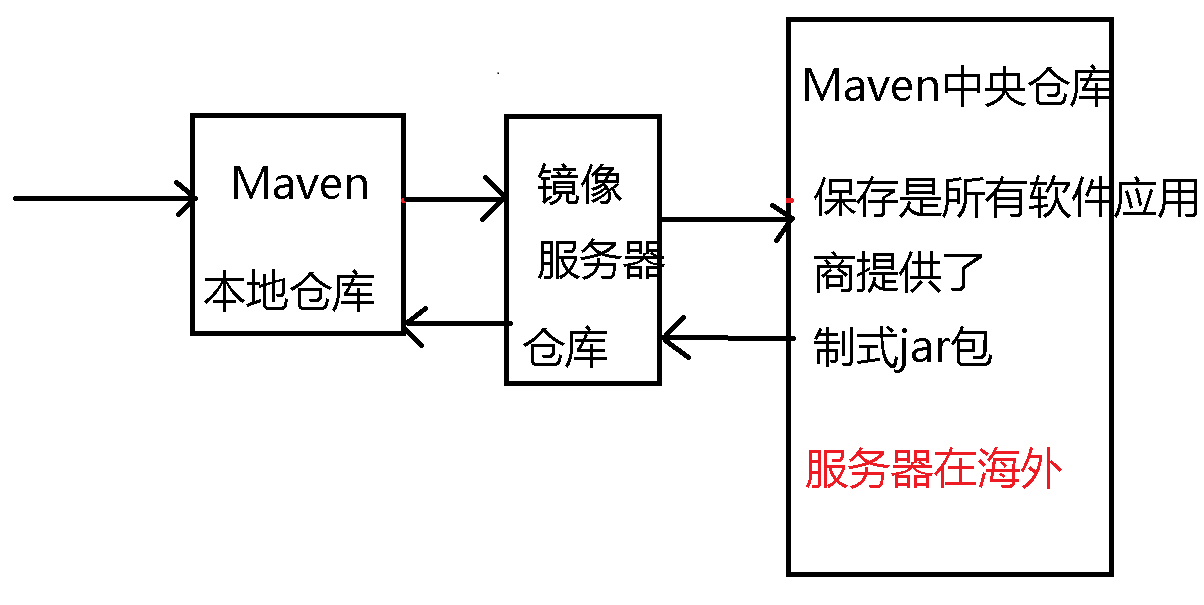
说明:Maven可以实现一站式的项目管理,jar包的依赖/jar包的管理/项目的发布和部署

### Maven的下载

说明:网址http://maven.apache.org/download.cgi#



### Maven的使用的原理



调用的原理:

1. 当用户需要jar包时,首先去本地仓库中查找jar包文件
2. 如果本地仓库中没有jar包,则连接镜像服务器进行jar包的下载.
3. 如果镜像服务器中没有改jar包,则连接中央仓库进行下载.
4. 镜像下载完成之后将文件保存到镜像仓库中,为后续用户提供服务.
5. 将jar包文件保存到本地仓库中,方便日后的使用.

### 镜像的配置

1. 达内学习:

<!-- 达内内网 Maven 镜像服务器配置 -->

<mirror>

<id>nexus</id>

<name>Tedu Maven</name>

<mirrorOf>\*</mirrorOf>

<url>http://maven.tedu.cn/nexus/content/groups/public/</url>

</mirror>

1. 阿里云镜像

<!--阿里云镜像-->

<mirror>

<id>nexus-aliyun</id>

<mirrorOf>\*</mirrorOf>

<name>Nexus aliyun</name>

<url>http://maven.aliyun.com/nexus/content/groups/public</url>

</mirror>

总结:使用maven时必须配置镜像文件,否则将来会自动的校验jar包时,如果没有联网.则会报校验失败.导致pom.xml报错

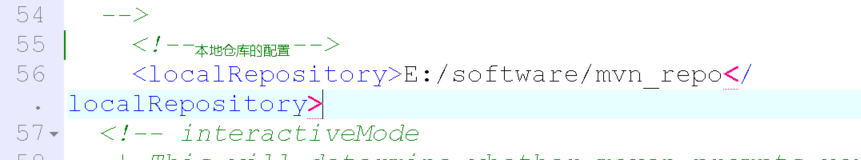
### Maven的配置

说明:Maven的级别介绍

1).用户级别

说明:用户级别是Maven的默认的使用级别,该级别会扫描特定的配置文件,进行加载.加载的路径${user.home}/.m2/**settings.xml**.

2).修改本地仓库的路径



3).修改镜像文件

达内的同学由于没有外网,需要通过达内的镜像实现jar包的现在.所以连接达内镜像

<mirror>

<id>nexus</id>

<name>Tedu Maven</name>

<mirrorOf>\*</mirrorOf>

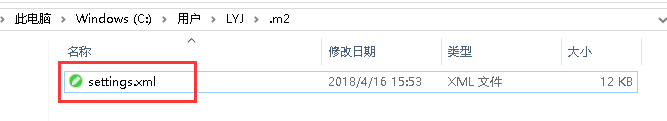
<url>http://maven.tedu.cn/nexus/content/groups/public/</url>

</mirror>

4).修改maven默认的JDK版本



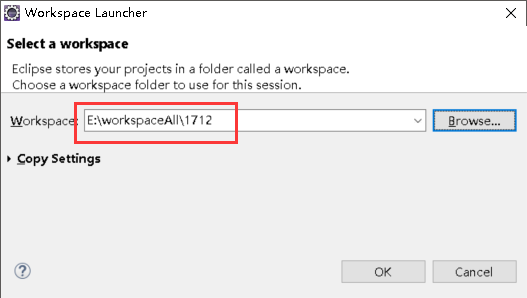
5).**将settings文件拷贝导入.m2文件夹下**



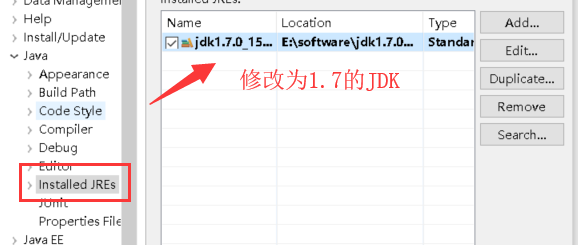
## Eclipse整合Maven

### 切换工作空间

在本地磁盘中创建文件1712,之后将eclipse中的工作空间进行切换

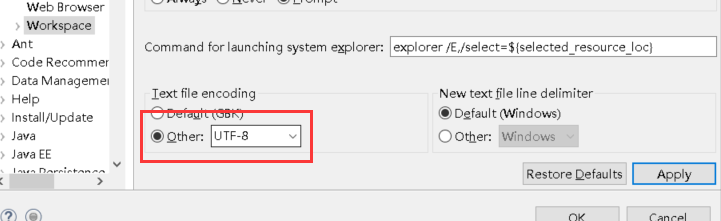


### 添加JDK



### 修改字符集

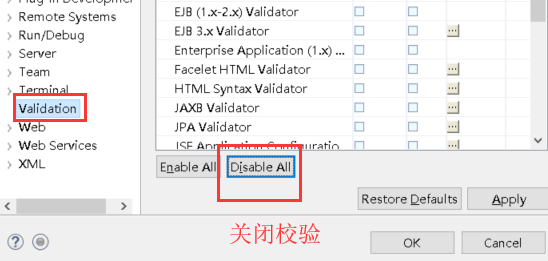
说明:方便同学导入教师代码没有乱码问题



### 关闭校验

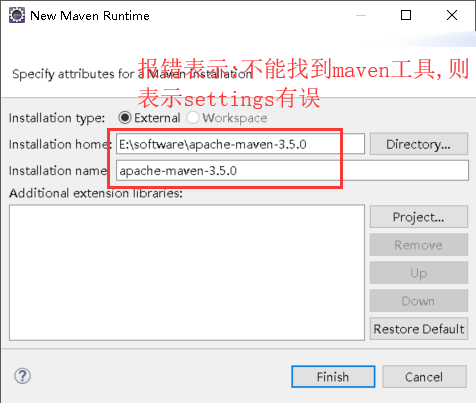
说明:如果不关闭校验,如果是分布式的项目,则在开发时可能会出现项目依赖的问题.

这时校验的时间会很长.



### 关联maven

1.引入Maven



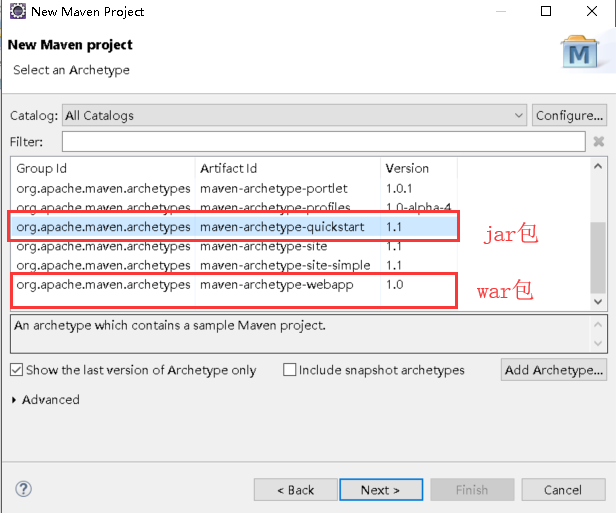
2.配置settings文件



## 骨架创建Maven项目

### Java的骨架

说明:通过骨架创建项目,可以极大的减少程序员添加项目中文件的过程.通过骨架其实就是创建项目的模板,在模板中定义的jdk的使用以及默认的jar包依赖的关系.自己可以根据项目中的依赖,手动添加.

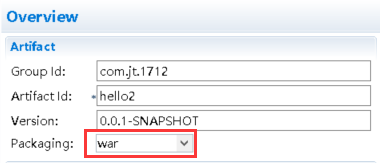


通过java项目的创建可以快速的构建一个jar包项目

### Web项目骨架

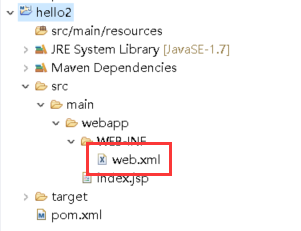
说明:通过webapp骨架创建出来的项目中,包含了web应用的一切的配置

1).打包的类型

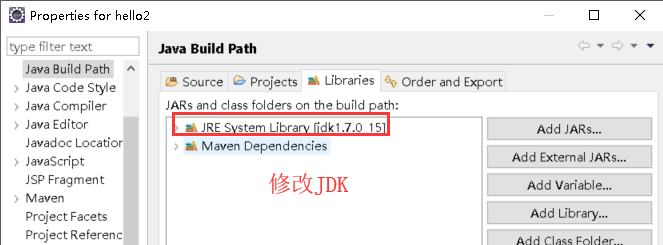


2).web的配置文件

能够自动的添加web的配置



3)生成java包/test包的方法



## Maven的中央仓库如何管理jar包

说明:通过坐标的形式实现jar包的管理.

构成坐标的要素:

组Id:一般都是以公司的域名定义的

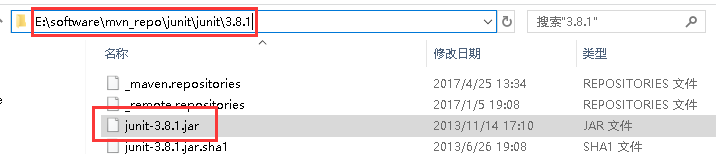
项目Id:标识当前项目的名称/项目的名称不能重复

版本号:唯一确定项目标识

<groupId>com.jt.1712</groupId>

<artifactId>hello2</artifactId>

<version>0.0.1-SNAPSHOT</version>



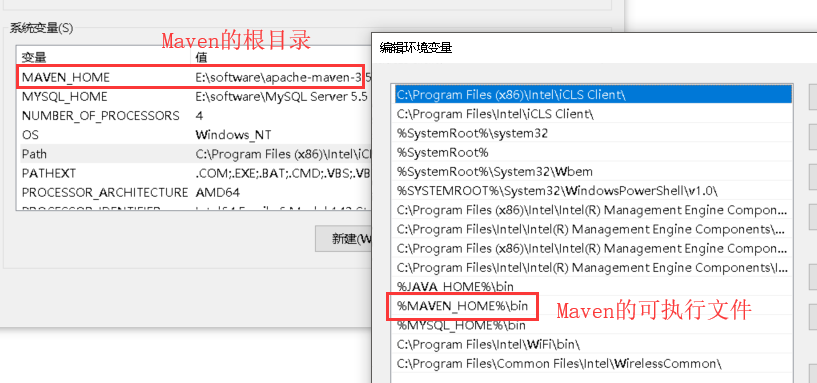
总结:Maven中使用了坐标的机制,可以非常方便的管理众多的jar包文件.方便进行下载.

## Maven项目打包/发布

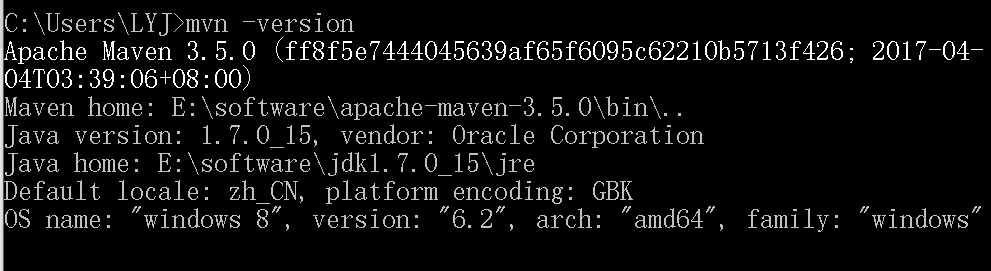
### 打包的命令

说明:如果通过maven命令实现打包则必须配置maven的环境变量.否则程序不能识别命令.

1).环境变量的配置



2).检测环境变量

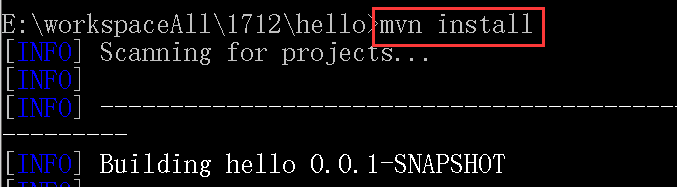


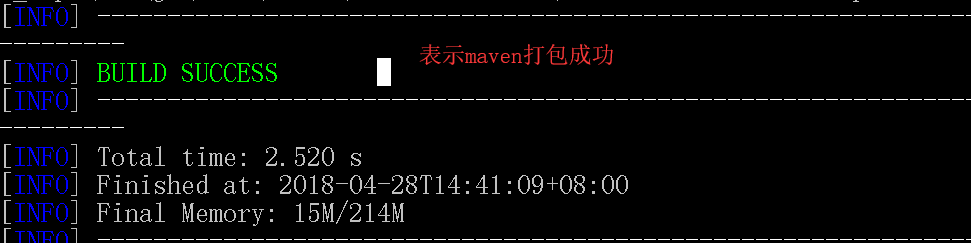
3).执行maven命令

mvn install 将项目打包

说明:如果需要打包maven项目,则必须在**maven项目的根目录**执行该命令(pom.xml)

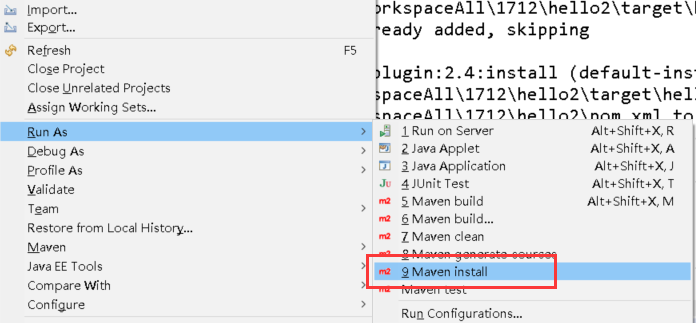
4).命令的执行

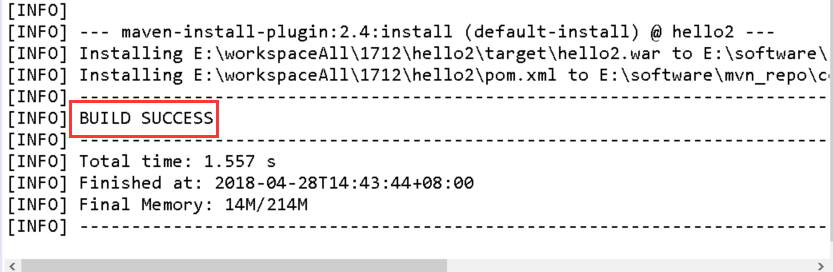




### Eclipse打包

说明:通过eclipse中的插件,可以实现项目的打包和部署.





打包成功后.INFO信息中会显示jar包/war包的路径.其路劲就是本地仓库的路径

### 使用坐标的意义

1. 极大的方便了jar包引用
2. 公司内部如果需要jar包引用,通过坐标快速定位.
3. 方便管理jar包的及版本控制

## Maven中jar包的依赖问题

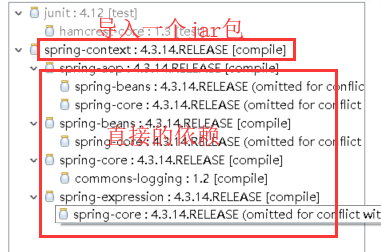
### Jar包依赖的意义

传统项目中jar包依赖的问题:

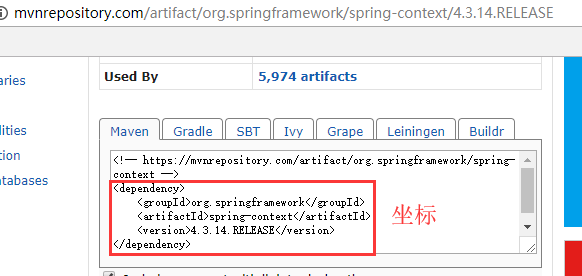
1. 如果构建项目则必须找别人拷贝jar包
2. Jar包冲突问题 (不好排查/springBoot/SpringCloud/)

Maven的优点:

1. 帮助管理jar包
2. 实现jar包的自动的依赖

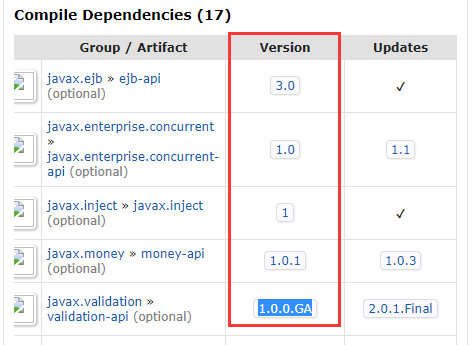


3).jar包的下载依赖的网站



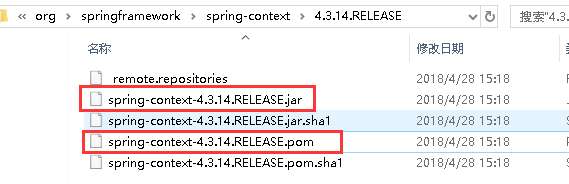
4).相关的jar包的版本号

说明:该网站提供了jar包的全部的依赖关系,如果需要引入jar包,则查询版本通过坐标进行导入.

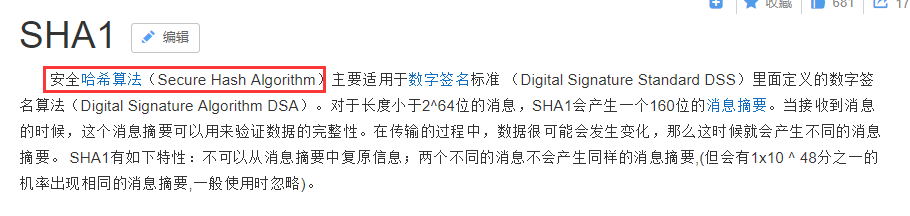


### Maven如何实现jar的依赖???

说明:当maven下载jar包时,首先获取需要的jar包文件.之后根据jar包的pom.xml文件中所规定的依赖,实现动态的依赖.



2).sha1加密



# 京淘的项目分布式部署

## 京淘项目的拆分

### 项目如何拆分

说明:根据功能模块的不同,实现项目的拆分.京淘项目中 前台/后台{商品的后台管理/购物车的后台管理/订单的后台管理…}.根据分布式的思想,将功能模块实现了拆分.最终拆分为不同的项目(工程).

传统项目中存在的问题:

1.传统项目中全部的功能模块都写在一个项目中,如果其中某个模块出现了问题.则可能会导致整个项目不能正常运行.如果项目越大,则出现问题的几率就越高.

所以采用传统的项目构建,不适用于大型项目.

大型项目的构建:

按照功能模块的不同,实现了项目的拆分.最后将京淘项目拆分为

**7大业务系统.2大项目的支撑系统.**

|  |  |  |
| --- | --- | --- |
| 序号 | 项目名称 | 主要作用 |
| 1 | jt-manage | 京淘后台管理系统 负责商品的更新和维护 |
| 2 | jt-web | 京淘前台系统，主要负责商品的展现 |
| 3 | jt-cart | 京淘购物车系统，负责用户购物车信息维护 |
| 4 | jt-sso | 京淘单点登陆系统，实现session数据共享 |
| 5 | jt-order | 京淘订单系统，负责订单维护 |
| 6 | jt-search | 京淘项目的全文检索 |
| 7 | jt-rabbitMQ | 消息队列系统 |
| 8 | jt-parent | 负责jar包的管理和依赖 |
| 9 | jt-common | 管理工具类文件 |

## 项目拆分遇到的问题

### 分布式项目中如何管理jar包

需求:分布式的项目中,如果每个项目自己管理jar包.如果遇到jar包升级和版本冲突时,修改的项目数量会很多.

解决:使用Maven中的继承实现

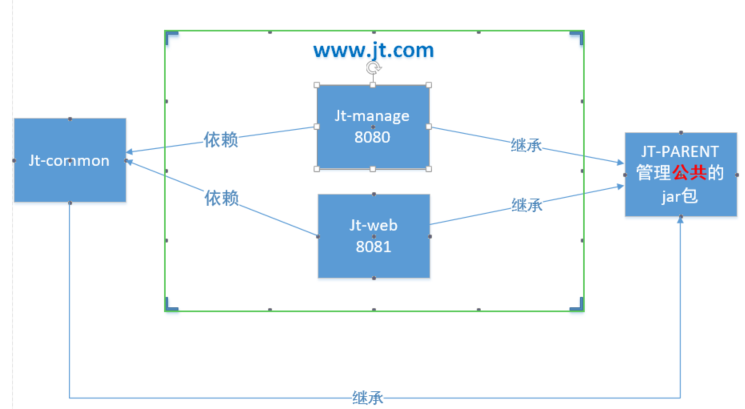
说明:构建一个项目.命名为JT-parent.在该项目中保存是整个架构设计中的公共jar包,如果将来需要修改jar包文件,只需要修改jt-parent即可.

重点:如果对jt-parent做了修改,则必须打包,否则依赖项目失效报错

### 分布式项目中如何管理工具类

问题:分布式项目中,需要使用工具类,如果每个项目都自己单独添加工具类这时会造成项目的维护不易.如果工具类发生了修改,则每个项目都需要修改!!

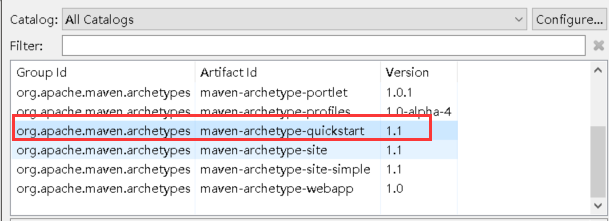
解决:采用jt-common项目定义工具类.由于工具类是java文件构成的,需要被别人直接的调用,所采用依赖的形式,进行注入.



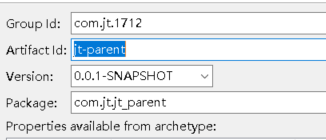
# 京淘项目构建

## jt-parent构建

### 选择骨架

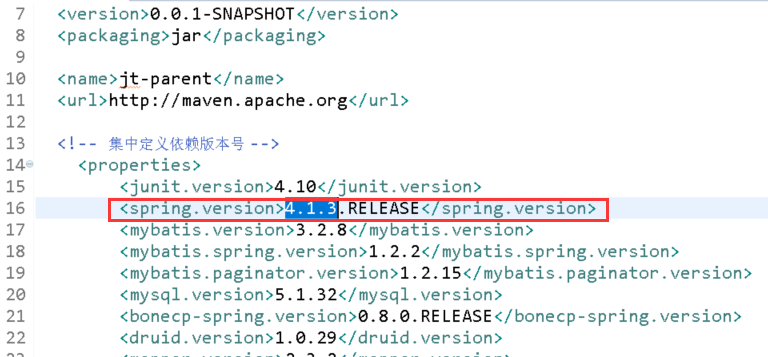


### 定义坐标



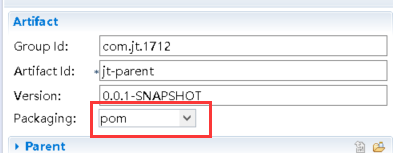
### 导入jt的jar包

说明:根据课前资料中的pom.xml导入jt-parent

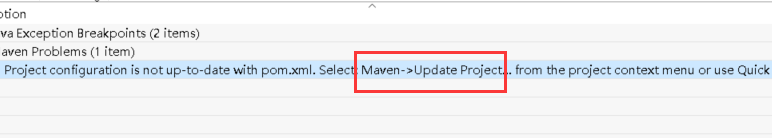


其中使用properties的作用是可以动态的引入该配置,例如spring中引入的版本号为4.1.3.以后可以动态的修改,方便jar包的升级.

### 修改Pom文件类型



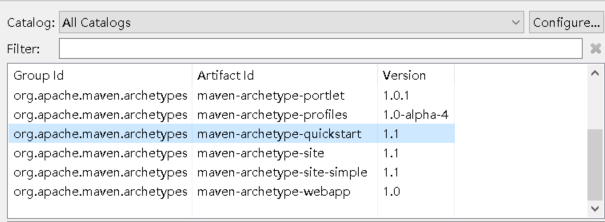
说明:该项目可以被继承.



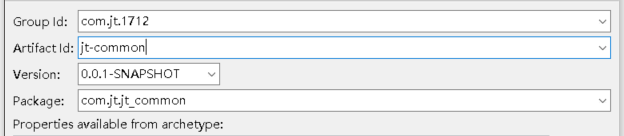
如果修改完项目后,出现了项目报错现象,则更新maven项目

## 构建jt-common项目

### 选择骨架

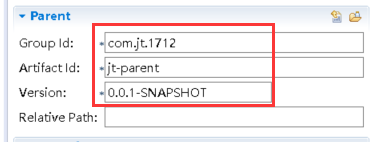


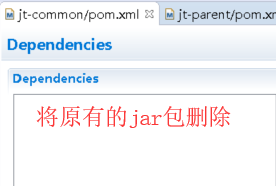
### 定义坐标



### 继承jt-parent

说明:添加jar包的继承





将原有的junit-3.8.1删除,使用父级的junit jar包

注意:jar包依赖的原则.如果引入父级的项目中的jar包与自己的依赖jar包冲突,则以自己依赖的jar包为准.

### 导入工具类文件

说明:将课前资料中的工具类文件的src文件复制.导入到jt-common项目中.

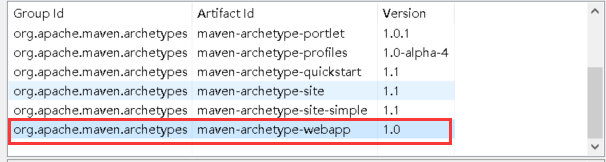
## 分布式系统中支持系统的总结

说明:通过业务支撑系统jt-parent和jt-common共同为项目提供了jar包的支持和工具类的支持.如果需要修改jar包文件则修改jt-parent.如果添加了工具类则修改jt-common.其中体现了分布式思想中**松耦合的思想**.

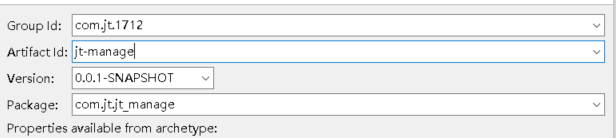
# 搭建京淘后台管理

## 京淘后台搭建

### 选择骨架

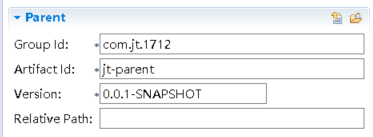


### 定义坐标

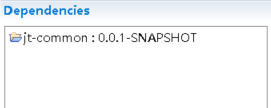


### 添加继承和依赖

1.表示继承jt-parent



2.依赖jt-common



## Tomcat插件使用

### Tomcat插件的使用

使用tomcat软件的问题:

1).tomcat使用的时间越久,tomcat启动的时间就越长.需要定时清理webapps文件夹

2).如果在一台机器中使用多台tomcat,需要进行复制,之后修改tomcat的端口号(需要修改3个端口号)

3).使用eclipse整合tomcat时比较繁琐.

## Tomcat插件添加

### 导入tomcat插件

<build>

<plugins>

<plugin>

<groupId>org.apache.tomcat.maven</groupId>

<artifactId>tomcat7-maven-plugin</artifactId>

<version>2.2</version>

<configuration>

<port>8091</port>

<path>/</path>

</configuration>

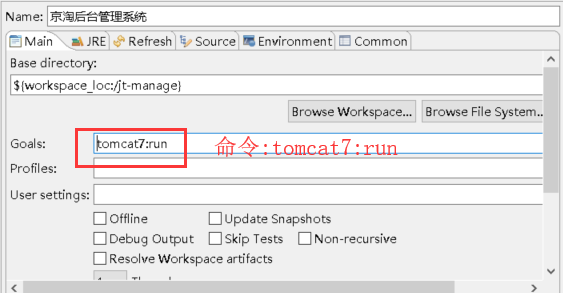
</plugin>

</plugins>

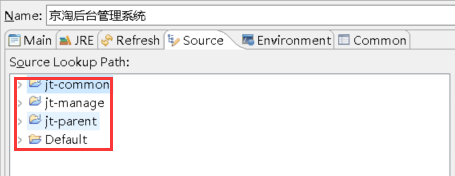
</build>

### 部署tomcat

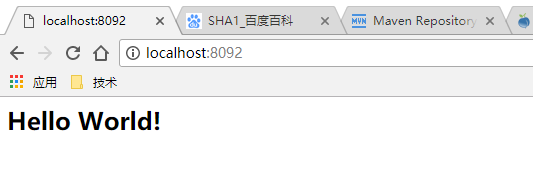
1).编辑启动命令:



2).添加源文件

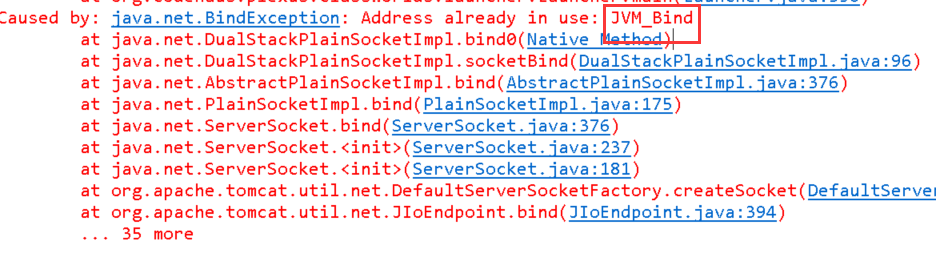


3).访问测试



### 常见错误

1. 端口被占用



说明:如果出现该错误.表示端口被占用,需要将原有的tomcat进程关闭.之后重启

