**密级：**

**java后台框架开发说明**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 文档编号 |  | | | 版本号 | V1.0 |
| 分册名称 | Java编码规范说明书 | | | 第1册/共1册 | |
| 总页数 |  | 正文 |  | 附录 |  |
| 编制 |  | 审批 |  | 生效日期 |  |

修改记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **版本号** | **变更控制报告编号** | **更改条款及内容** | **更改人** | **审批人** | **更改日期** |
| V1.0 |  | 创建 | 常勇 |  | 2017-4-10 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**目 录**

[1.简介 1](#_Toc479675587)

[1.1 目的 1](#_Toc479675588)

[1.2 范围 1](#_Toc479675589)

[1.3 术语 1](#_Toc479675590)

[1.4 参考文档 1](#_Toc479675591)

[3.项目工程 1](#_Toc479675592)

[3.1 项目介绍 1](#_Toc479675593)

[3.1.1 项目名称 1](#_Toc479675594)

[3.1.2 源码地址 1](#_Toc479675595)

[3.1.3 工程结构介绍 1](#_Toc479675596)

[3.1.4 开发环境搭建 2](#_Toc479675597)

[3.1.5 编译打包 6](#_Toc479675598)

# 1.简介

## 1.1 目的

统一开发规范，团队人员协作方便，互相开发手段以及代码互相了解，方便接手其他人员的事务。

## 1.2 范围

Maven做项目管理，eclipse为ide工具，spring以及dubbo等相关框架

使用mysql关系型数据库，redis非关系型数据库，zookeeper为注册中心，rabbitmq为消息队列服务器

供后台开发及相关人员进行阅读

## 1.3 术语

## 1.4 参考文档

《1\_Java编码规范说明书.docx》

《2\_Java注释API导出说明书.docx》

# 3.项目工程

## 3.1 项目介绍

### 3.1.1 项目名称

生意宝

### 3.1.2 源码地址

新整理的Svn服务器地址：

http://192.168.1.100:8080/svn/dianda/java/shengyibao/boot/20170410/shengyibao

老工程代码svn服务器地址：

http://192.168.1.100:8080/svn/dianda/java/shengyibao/branches

### 3.1.3 工程结构介绍

旧工程代码不做介绍，这里只介绍新结构，由于需要保证旧代码实现的服务要正常运行，所以服务的旧代码是整体迁移到新结构中的，后期整个开发按照新结构实现，已经实现业务的代码根据时间做相应的改造

先做如下规定：不可随意创建修改maven的module、包、源代码等会改动工程结构的文件夹以及文件，统一协商做决定，开发人员开发过程只对文件内容做编辑（紧急情况除外）

如图：

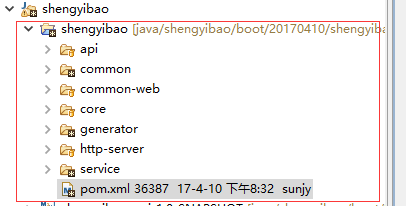


图 3.1.3.1

为svn下载的代码结构

shengyibao：为整个工程最高节点，作为父工程，为管理项目的聚合，继承以及依赖相关作用

shengyibao工程七个子模块module如下：

api：主要包含所有业务接口以及业务所需实体POJO

common：存放工具等各种公用组件

common-web：存放http网络接口的一些公用组件  
 core：存放连接数据库的dao、redis以及消息队列等的一些公用连接组件

generator：存放一些反向生成代码的工具

http-server：存放实现http协议的网络接口代码，主要把后台服务以http协议抛出，供其他方调去

service：存放实现api中业务接口的业务代码，再注册到注册中心，供http-server等网络接模块调去

### 3.1.4 开发环境搭建

使用eclipse从svn导入源代码（结构如：图 3.1.3.1）之后，把每个子模块导入到eclipse中作为eclipse工程结构项目开发

以api为例介绍如何导入

如图 3.1.4.1 鼠标右键该api模块，点击【import】

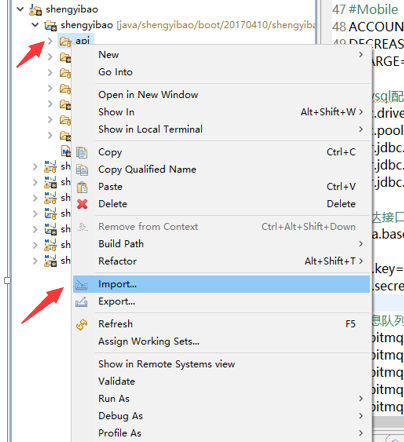


图 3.1.4.1

之后出现如图3.1.4.2 弹出框，选择Maven模块中的Existing Maven Projects模块后，点击【Next】

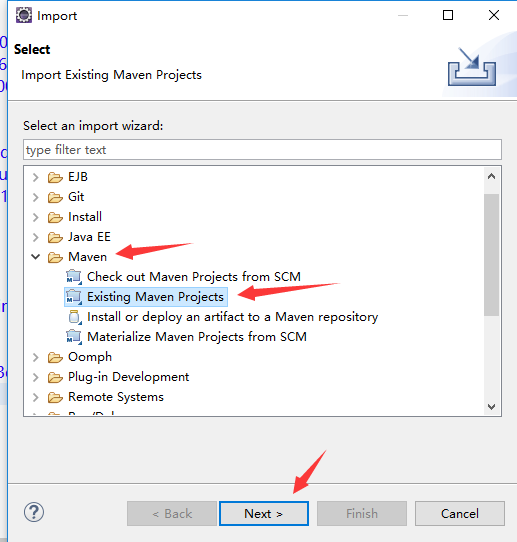


图3.1.4.2

之后按照图3.1.4.3 所示，做勾选，选择等之后点击【Finish】即可

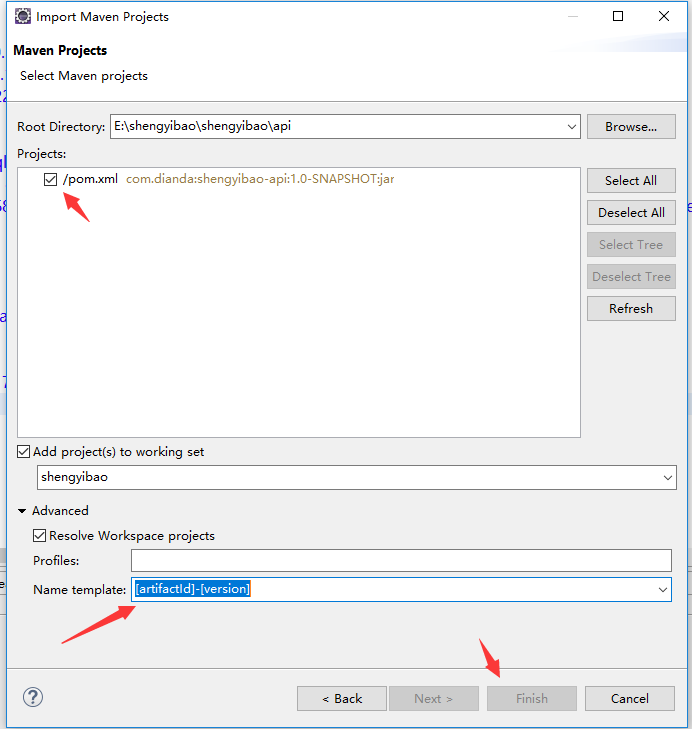


图3.1.4.3

之后会看到eclipse中出现工程，如图3.1.4.4

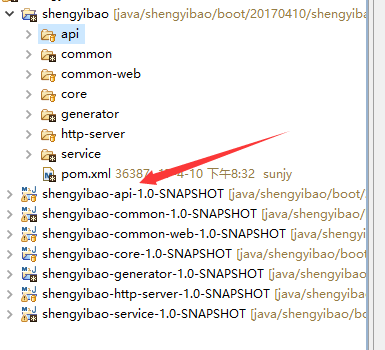


图3.1.4.4

至此把所有Maven结构项目转为eclipse结构项目，可以在eclipse做编辑编译打包操作

### 3.1.5 编译打包

如图3.1.5.1 后台现只给服务器管理人员交付shengyibao-http-server和shengyibao-service两个包做部署，其中api、common等模块只是作为jar被其他工程依赖使用，下边以shengyibao-http-server开发编译打包为例介绍开发流程

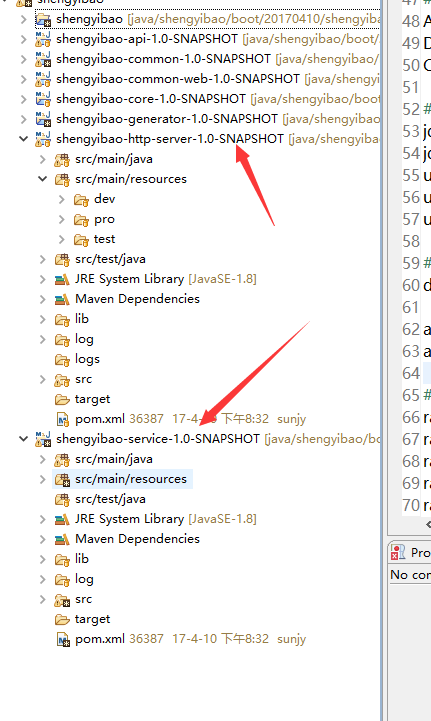


图3.1.5.1

Maven结构不做复述，这里讲解内容依照开发人员已经熟悉Maven和eclipse等相关工具的使用，如需请自行查阅相关资料学习，如图3.1.5.1 两个需要交付运维的包结构相同，以http-server为例介绍

项目采用springboot作为容器启动项目，项目打包为jar，直接使用java启动

如图3.1.5.1 资源文件夹resouces下有3个目录dev、pro和test，分别为开发、生产和测试环境配置变量，采用Maven打包入参方式进行各环境配置分派，具体命令为：mvn clean package -DskipTests -P dev 该为打包dev即开发配置的包，工程中默认为dev开发配置，所以开发人员不必进行切换

可以使用eclipse的Maven插件打包相应的包，该情况为开发人员在紧急情况下打包，直接上传服务器服务，一般不建议使用，采取交付仓库供运维人员下载部署

如图

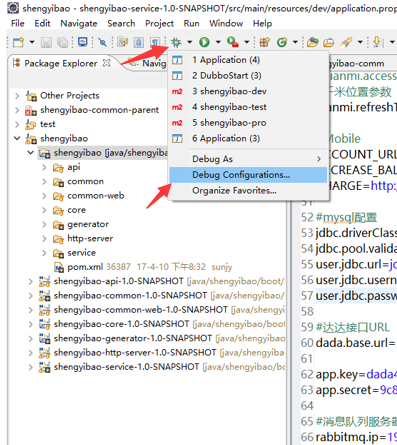


图3.1.5.2

如图创建编译打包入口，打开如图图3.1.5.2编辑界面，勾选编辑如图图3.1.5.3，保存之后在如图图3.1.5.4所示有开发，测试以及上产环境打包快捷方式

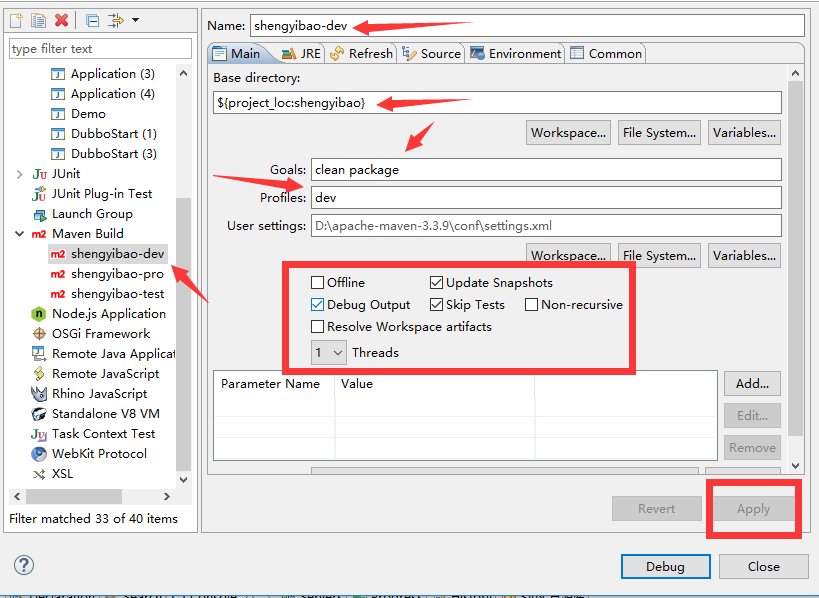


图3.1.5.3

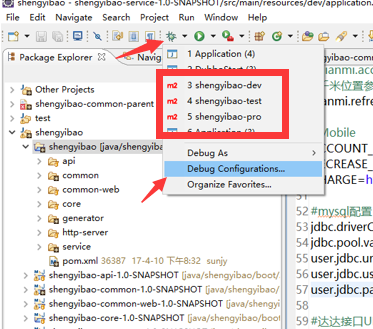


图3.1.5.4