**后台基础框架示例设计说明**

**版本号1.0**

**编写人：贺康**

**编写时间：2016年1月4日**

目录

[1 概述 3](#_Toc440975573)

[1.1 简述 3](#_Toc440975574)

[1.2 目标 3](#_Toc440975575)

[1.3 参考资料 3](#_Toc440975576)

[2 术语表 3](#_Toc440975577)

[3 技术架构 3](#_Toc440975578)

[4 项目结构 4](#_Toc440975579)

[5 工程目录 5](#_Toc440975580)

[5.1 分层 6](#_Toc440975581)

[5.2 开发环境 6](#_Toc440975582)

[5.3 Eclipse开发环境配置 8](#_Toc440975583)

[5.4 Idea开发环境配置 13](#_Toc440975584)

[5.5 项目工具 20](#_Toc440975585)

[5.5.1 Mybatis Generator代码生成工具 20](#_Toc440975586)

[6 需要调整的问题 21](#_Toc440975587)

# 概述

## 简述

目前技术团队没有统一的基础项目，现在根据中台的项目结构形成1套java开发的基础demo，约定项目框架标准，以供以后的新项目快速使用，提高工作效率。

## 目标

1. 不重复造轮子。
2. 统一技术框架。
3. 操作风格统一。
4. 业务需要单独的后台时在此基础上能快速创建。
5. 由于前端资源紧张，尽可能的少占或不占前端资源。

## 参考资料

[技术部java开发规范](http://wiki.hitao.top/pages/viewpage.action?pageId=6292152)

# 术语表

无

# 技术架构

分层MVC模式

View 前端：

Js库：jQuery 1.12.0

Ui控件：Easyui 1.4.4

模板引擎：Freemarker 2.3.20

Controller 业务逻辑：

请求控制、响应：Spring MVC 4.0.4.RELEASE

对象管理：Spring 4.0.4.RELEASE

Modal 数据访问层：

Mybatis 3.2.7

# 项目结构

Hitao大团队核心框架

hitao-pom pom工程

└─ hitao-framework 基于spring-webmvc的框架性的支持

Hitao ERP框架

hitao-pom pom工程

└─ hitao-support ERP团队一些的通用支持框架

后续可能拆分或添加成更细的组件框架

Demo 一般项目的工程结构初稿（Maven项目，源自营销中心）

分子工程的目的是提供jar给其它系统引用，如果没有引用，可以不分

hitao-demo 父工程（聚合项目，父级为hitao-pom）

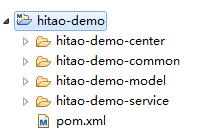
├─ hitao-demo-common 公共包

├─ hitao-demo-model 模型

├─ hitao-demo-service 服务接口

├─ hitao-demo-center 应用服务中心（必需）

└─ pom.xml pom文件



common公共包工程：

项目内部不依赖其它工程的业务公共类、工具类。

模型工程：

对外部应用提供对象模型类

服务接口工程：

对外部应用提供接口类和验证通信实现（公司内部应用服务间调用统一使用dubbo框架）。

工程分两种情况

1. 调用方使用dubbo框架的，工程只需提供接口类。
2. 调用方非dubbo框架的，工程提供接口类和验证通信实现。

如果有对应多个不同协议类型的外部应用就分成多个工程。例如:提供对内dubbo调用和对外(第三方)http api，则建两个接口工程。

应用服务中心工程：

web应用工程，包括业务、服务的实现、数据库操作、web应用、提供远程调用等。

# 工程目录

Demo的Maven工程目录初稿（源自营销中心）

├─ pom.xml

└─ src

├─ main

│ ├─ **java**

│ │ └─ com.hitao.demo 项目名称

│ │ ├─ **common**  **公共包**

│ │ ├─ **center** **应用服务中心**

│ │ ├─ **model**  **模型**

│ │ └─ **service**  **服务接口**

│ ├─ **resources** **配置文件**

│ ├─ **script** **sql脚本文件**

│ └─ **webapp**

│ ├─ index.html

│ ├─ META-INF

│ │ └─ MANIFEST.MF

│ ├─ **statics** **静态资源**

│ │ ├─ css 样式

│ │ ├─ images 图片

│ │ ├─ js 页面的js文件

│ │ └─ plugin js的插件、类库(jquery、easyui等)

│ └─ WEB-INF

│ ├─ template  **模板文件**

│ └─ web.xml

└─ **test** **单元测试**

├─ java

│ └─ com.hitao.demo

└─ resources **单元测试资源文件**

## 分层

PageView

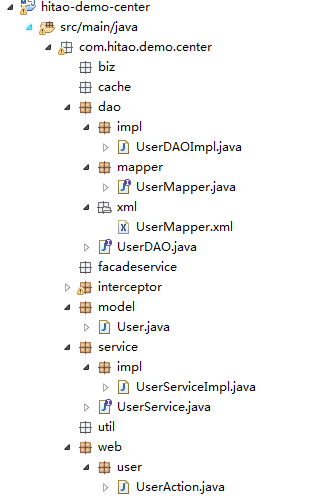
Action

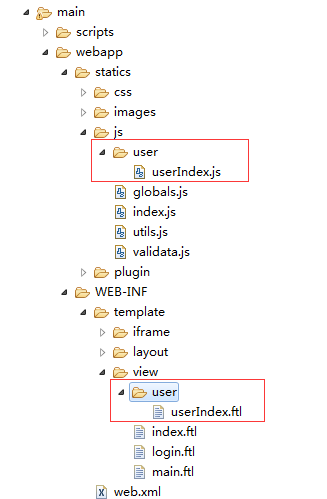
Service

DAO

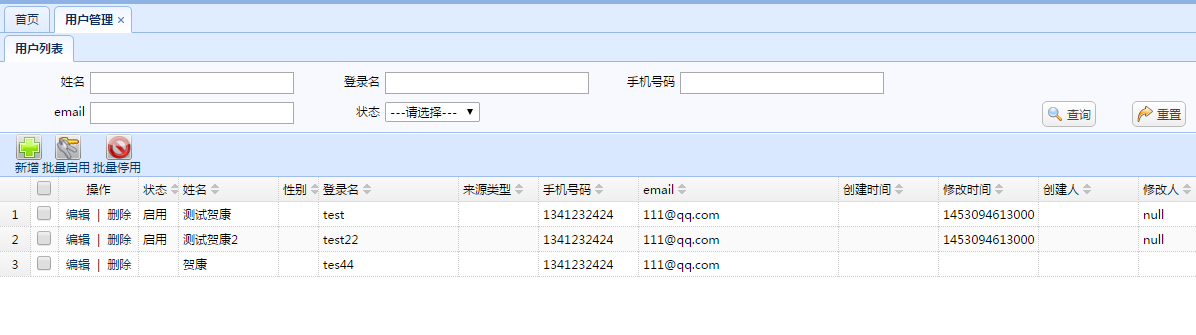
Mapper

## 用户管理示例代码结构





## 用户管理CRUD示例界面



## 开发环境

Jdk1.7.79+

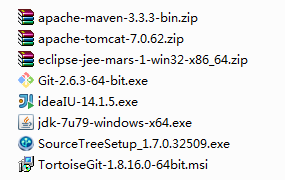
Tomcat7.0.62+

Maven3.3.3+

Eclipse4.5.1+或Idea14.1.5+

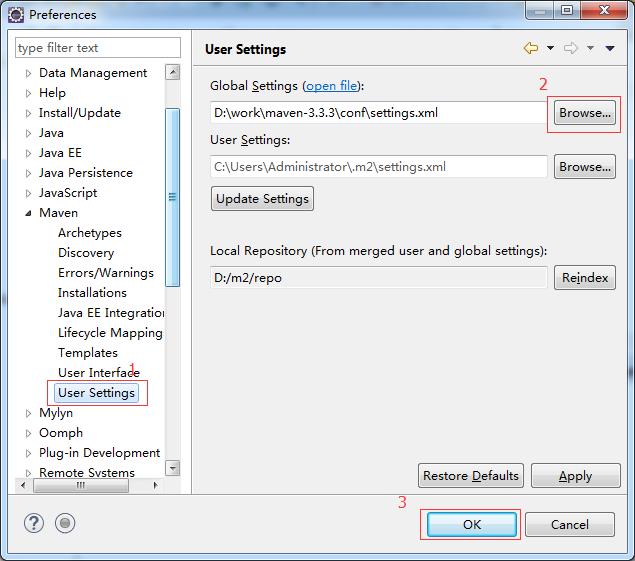
Git-2.6.3+

TortoiseGit1.8.16+或SourceTreeSetup1.7.0

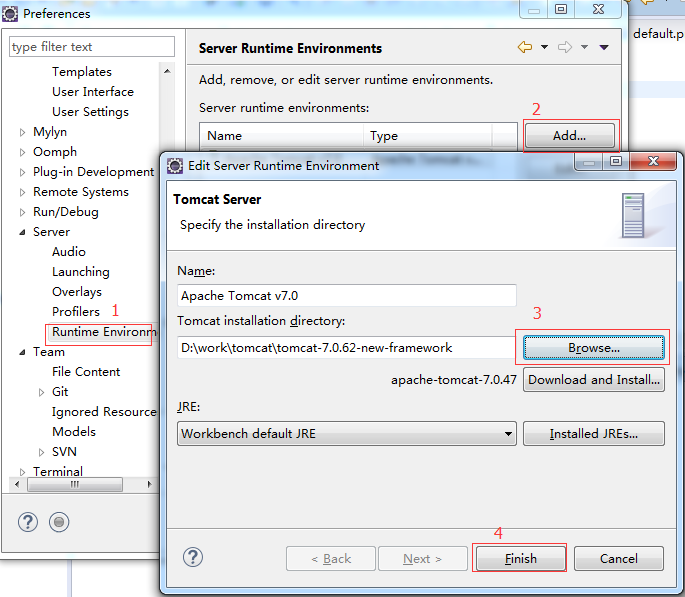


## Eclipse开发环境配置

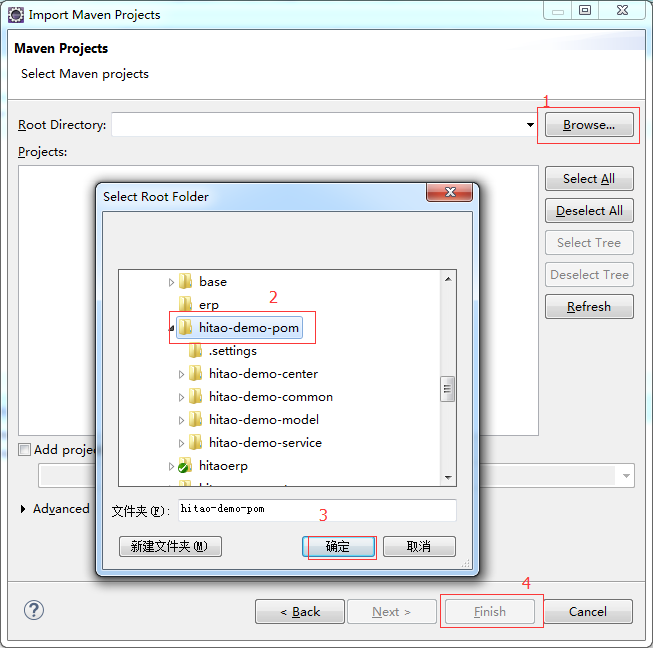
1. 安装jdk，设置环境变量java\_home和path
2. 解压Maven，比如：d:\work\maven，设置环境变量maven\_home和path
3. 解压tomcat，比如：d:\work\tomcat
4. 解压eclipse，比如：d:\work\eclipse
5. maven设置，Project -> Preference -> Maven

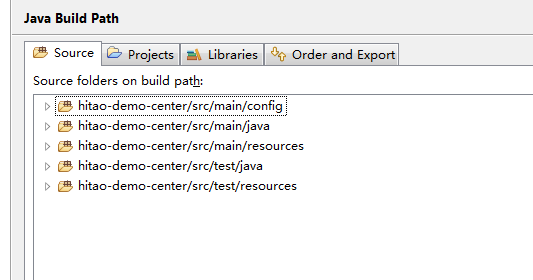


1. tomcat设置，Project -> Preference -> Server

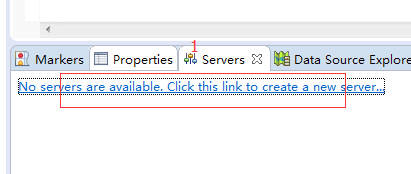


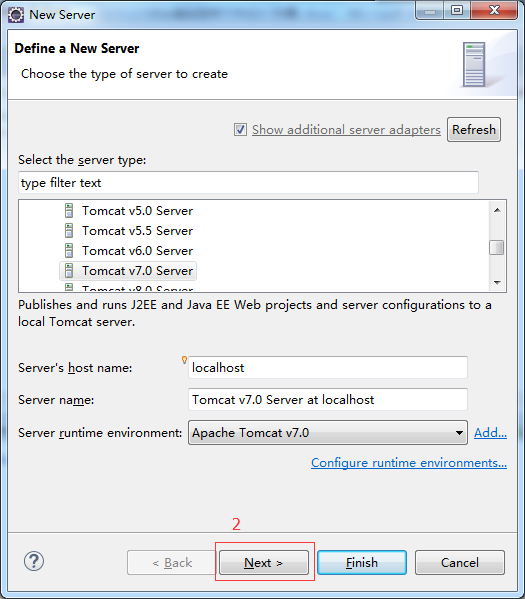
1. 导入Maven示例工程，File -> import -> Maven -> Existing Maven Projects

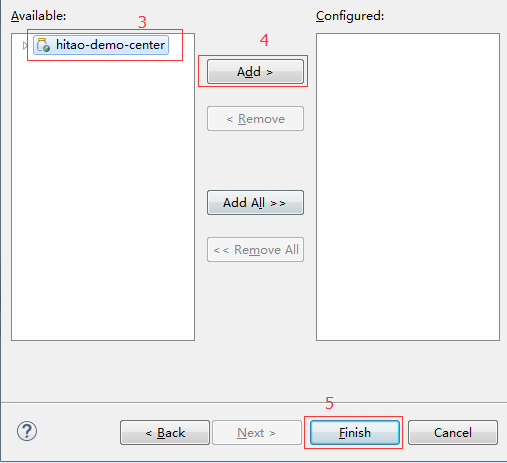


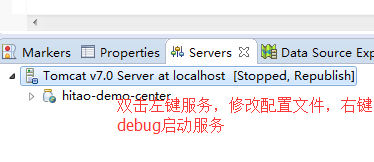


1. 新增启动示例的tomcat服务

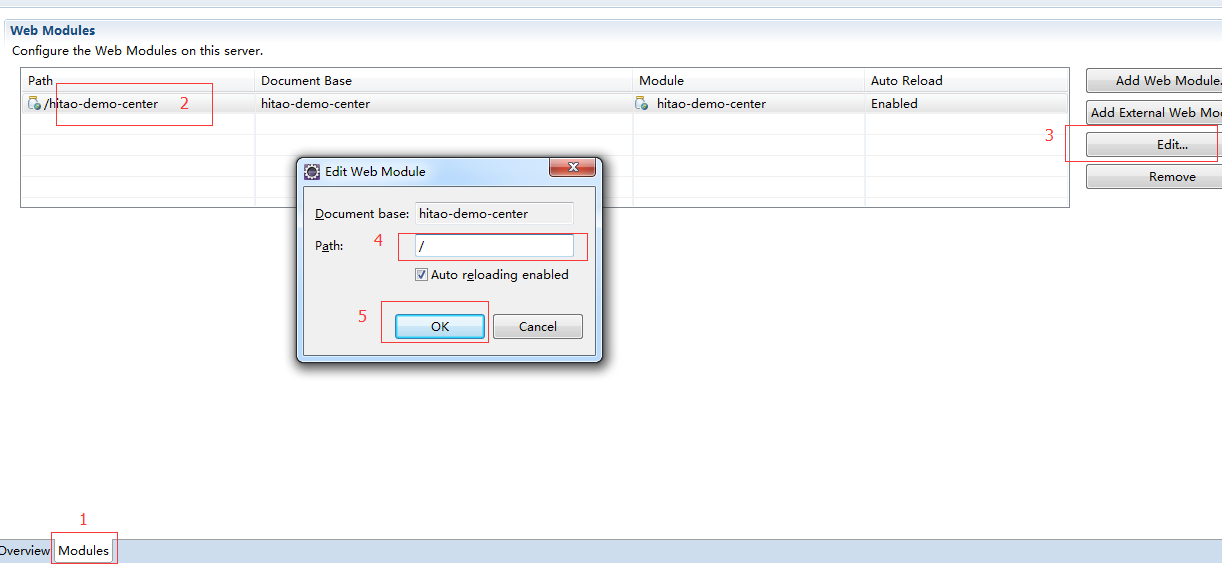








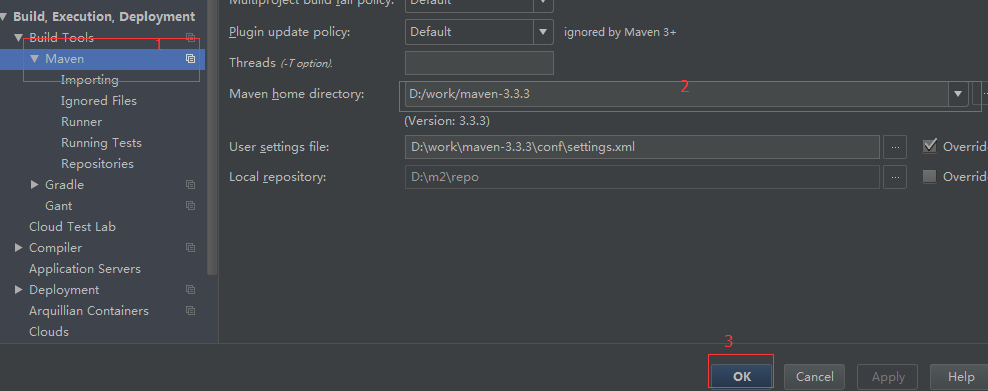
修改webapp访问路径



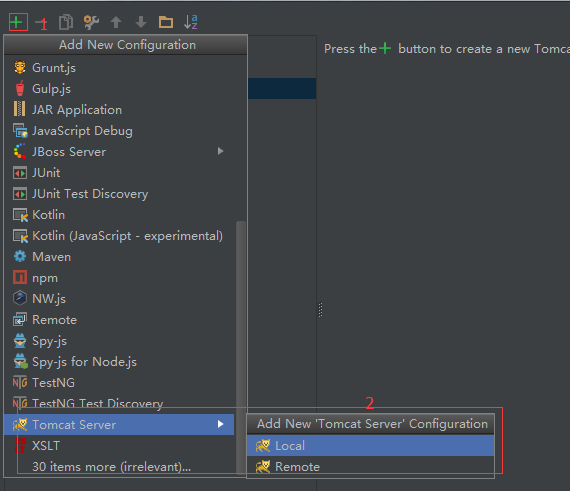
## Idea开发环境配置

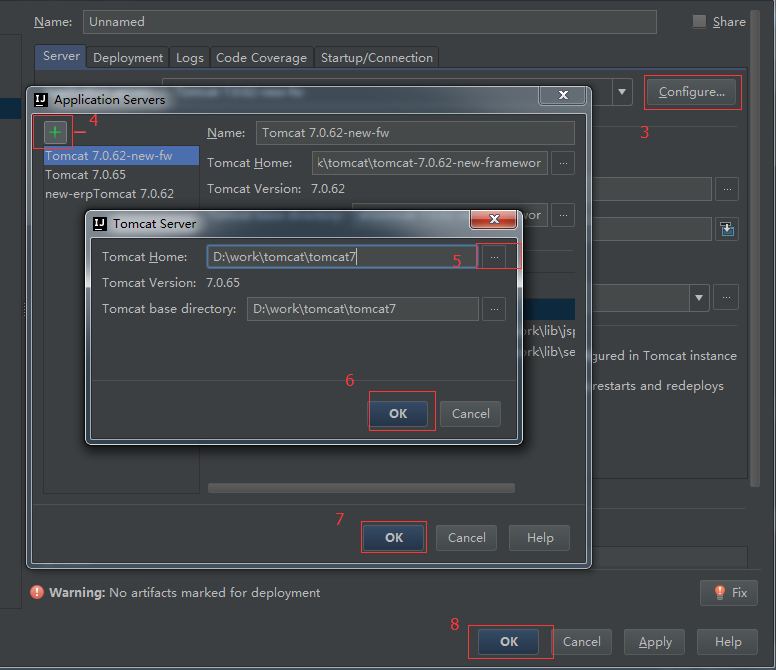
安装jdk、tomcat、maven参考eclipse开发环境配置

1. 安装idea(略)
2. Maven设置，File -> Setting -> Build, Execution, Deployment -> Build Tools -> Maven

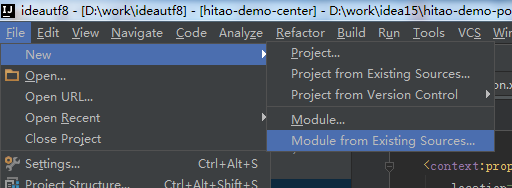


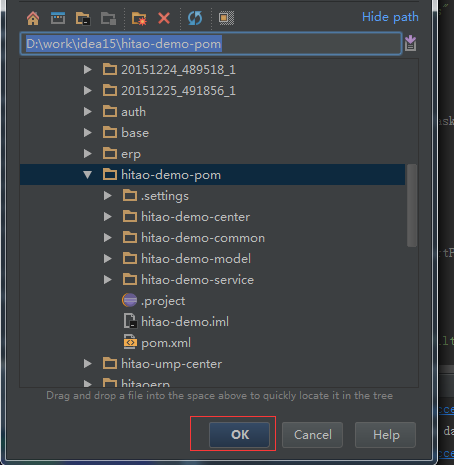
1. Tomcat设置，Run -> Edit Configurations…

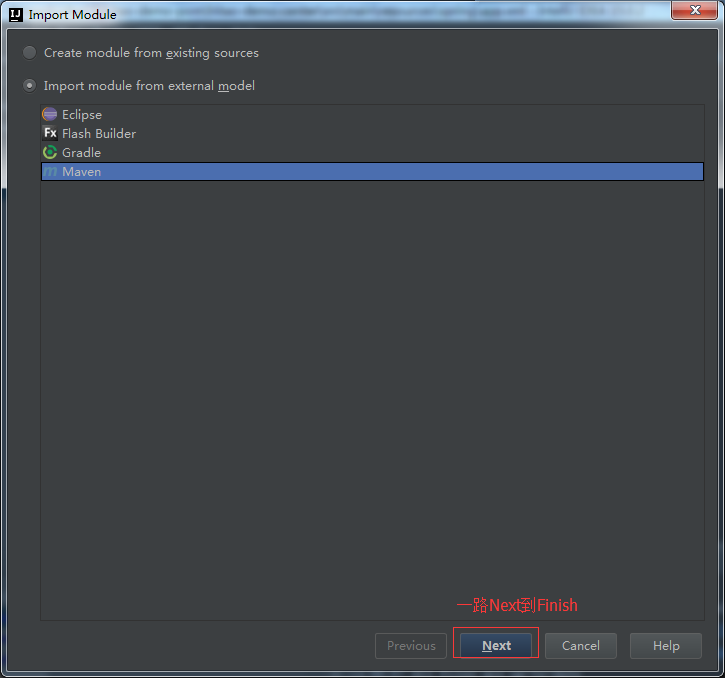


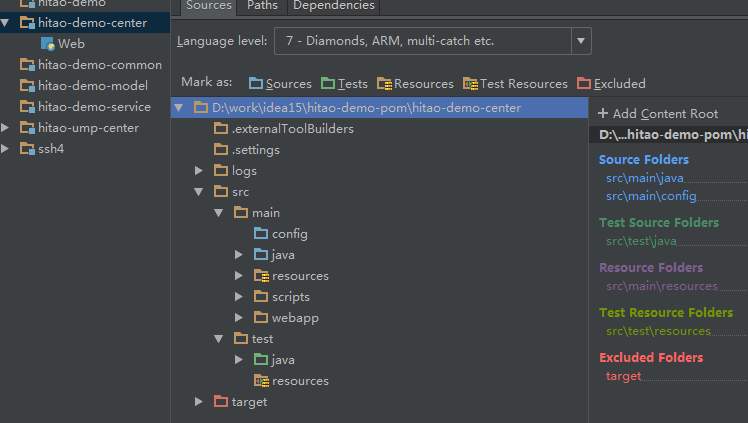


1. 导入Maven示例工程



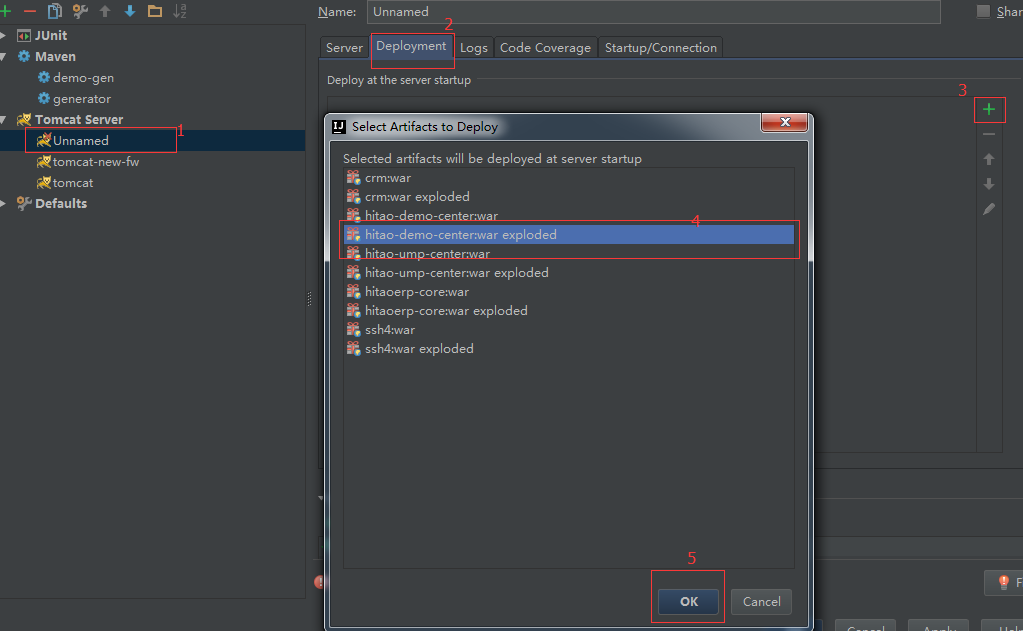


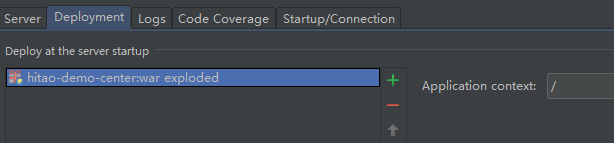




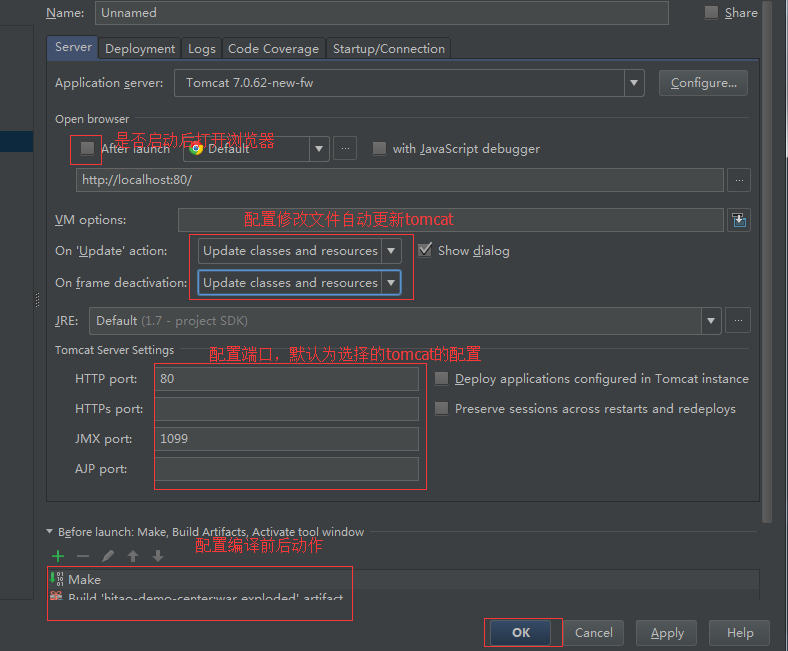
1. 新增启动示例的tomcat服务，Run -> Edit Configurations…

选择部署服务

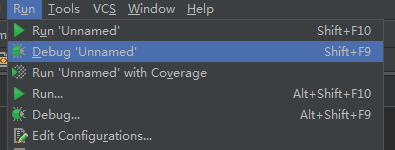




修改启动、运行配置项



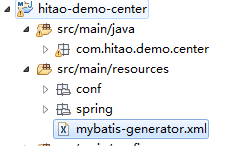
启动



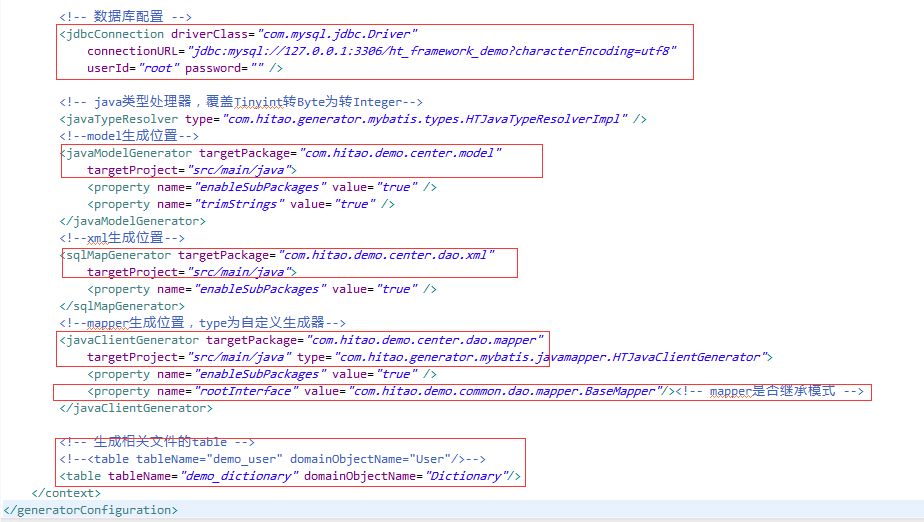
## 项目工具

### CURD代码生成工具

1. 打开Center项目下的mybatis-generator.xml，修改里面的配置信息

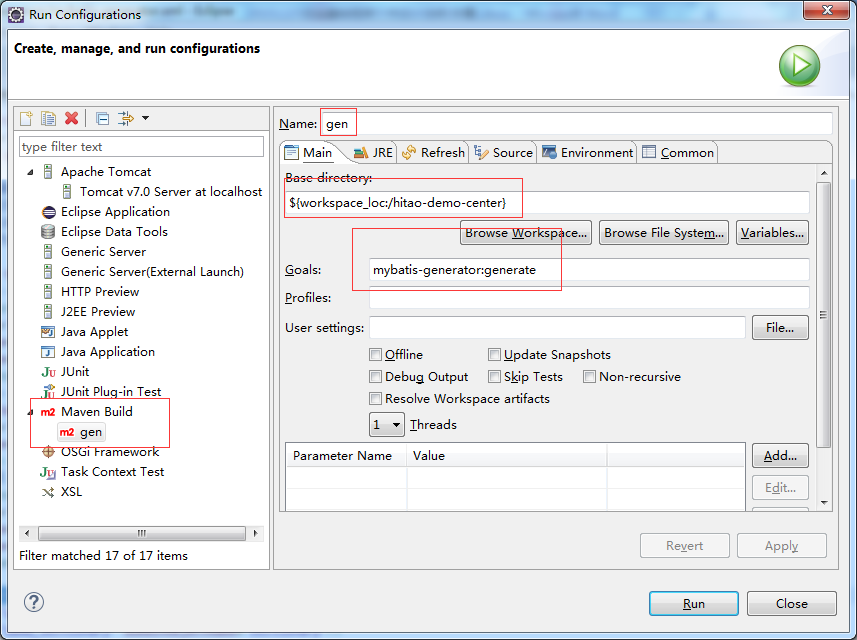




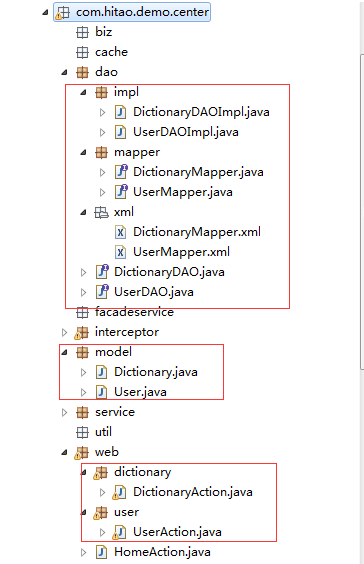


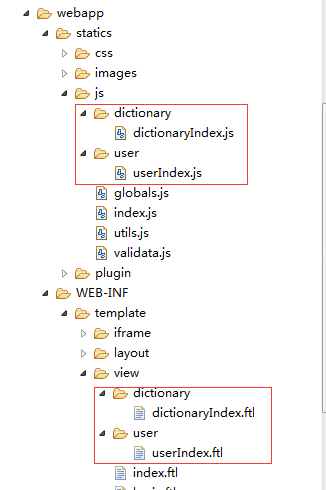
注意红框的地方，最重要的是数据库配置和最下面需要生成代码的table。

1. 项目下运行mvn mybatis-generator:generate或者添加运行Maven Build， run -> run configurations… -> Maven Build



1. 运行Maven，会在配置的位置自动生成CURD代码





PS1：由于代码生成工具工程没有加入公司私服，要用时需要运行一下mvn clean install命令在本地Repository中安装。

PS2：注意多次生成mapper.xml是合并代码的，需要把重复的代码删除掉，不然会出错

### 多配置项打包

开发环境maven打包命令 mvn clean package -P dev

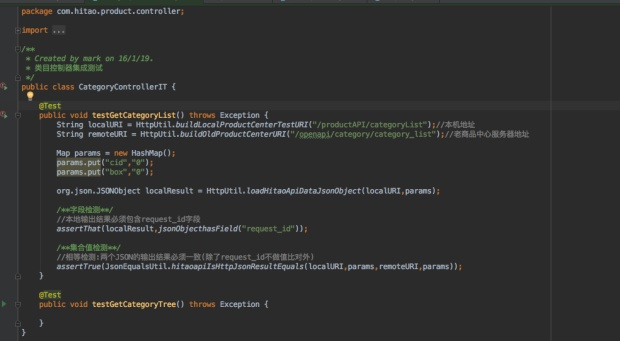
测试环境maven打包命令 mvn clean package -P test

预发环境maven打包命令 mvn clean package -P pre

生产环境maven打包命令 mvn clean package -P prod

### 集成测试

新建单元测试类(类名以IT结尾)，直接比对两个接口的JSON输出结果：



运行mvn integration-test

### Findbugs插件

hitao-framework框架已加Findbugs插件，运行mvn findbugs:findbugs或mvn findbugs:gui（gui界面）查看findbugs的report

# 需要调整的问题

1. mavn分环境配置项(dev、test、pre、prod对应开发、测试、预发、生产) 已加
2. pom.xml加测试、ci、findbugs、checkstyle（可先用插件）的 jar包 已加
3. 基类加批量插入 已加
4. 完善自动生成工具 已加
5. 生成方法可配置 暂时不加
6. 未来改成服务的方式，配置数据库、参数生成代码zip包下载