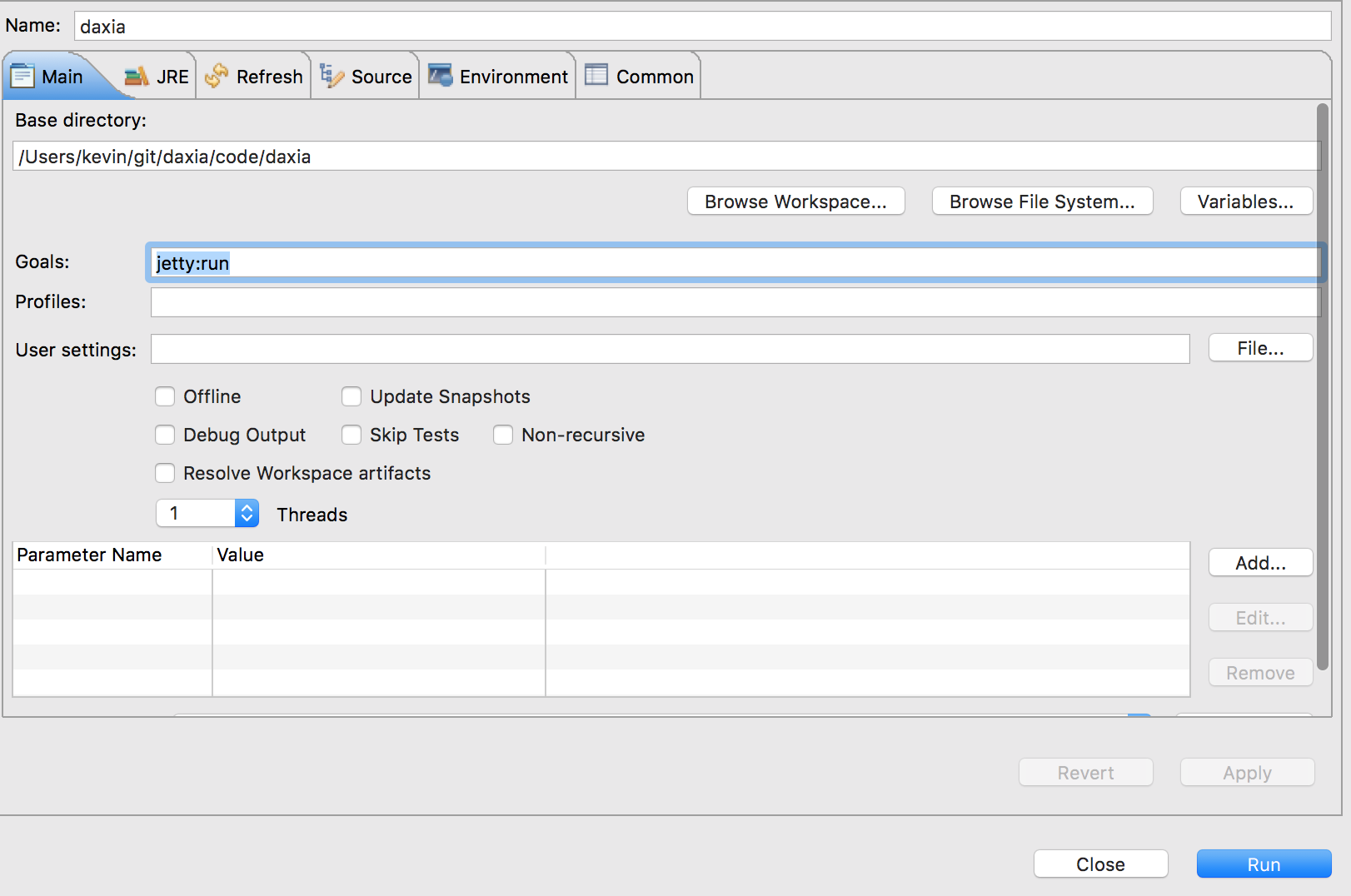
# 框架说明

1. 总体架构
   1. Hibernate + spring mvc + jsp
   2. Dao + service + controller三层
2. 项目结构
   1. 这是一个maven项目
3. 运行项目
   1. 导入数据库，在项目下的doc/db.sql
   2. 修改数据库连接参数，在src/main/resources/app.properties，改成本地的
   3. 推荐使用maven的jetty插件进行运行，即项目右键--> run as --> mvn build ，然后配置 如 图  
      第一次会比较慢，因为要从网上下一些东西，以后就可以把offline选上，就会快很多。但是如果pom.xml里增加了新的依赖，就需要联网，这时就要选取消offline
   4. 也可以用tomcat的形式跑，方法自行百度
   5. 跑起来成功后，访问http://localhost:8080即可，如果是tomcat跑的，还要加上应用名http://localhost:8080/xxx (xxx就是项目的名字）
4. 命令规范
   1. 数据库表名全小写，字段名小写开头，驼峰式写法。如果引用了其它的表的id，写成xxx\_id的形式
   2. 数据库id都是bigint(20)的形式，自增
   3. 数据库规范只是建议，不做强制，有个性化的需求可以以自己的为准
   4. 日期都是datetime
   5. 如果表名是example，对应的model名是Example，dao是ExampleDAO， service是ExampleService， 后台管理的控制层是AdminExampleDAO（跟后台管理相关的controller都要以Admin开头，访问url都是admin/开头，比如admin/example/list），前端网页比如门户之类的不要以Admin开头，就写成ExampleController。列表页是example\_list.jsp，详情页是example\_detail.jsp
5. 开发步骤，以开发一个example模块为例
   1. 在数据库建立一张example表，如果已经有表，就跳过这一步
   2. 写实体类Example.java
   3. 写DAO层ExampleDAO
   4. 写Service层ExampleService
   5. 如果要做后台管理的模块，就写AdminExampleController，然后写example\_list.jsp和example\_detail.jsp
   6. 如果是用在前端站点，就写ExampleController
   7. 配置菜单，菜单名为示例管理，url为/admin/example/list
   8. 重启项目，登陆后台管理，应该就可以看到菜单管理了
6. 权限管理
   1. 权限是针对角色来进行的
   2. 每个角色拥有一系列指定的权限
   3. 每个用户都有一个或多个角色
   4. 相关的表是role, authority, userrole, roleauthority
   5. 所以如果要为example增加权限控制的话，步骤如下
      1. 在权限管理，增加示例管理的权限
      2. 在角色管理，为已有角色或创建一个新角色，并为这个角色配置“示例管理”的权限。权限名是中文，编码是英文
      3. 在用户管理，为用户配置该角色
   6. 在菜单管理里，为“示例管理”这个菜单，配置“示例管理”这个权限。只有拥有这个权限的用户才可以看到这个菜单
   7. 重新登陆后权限就会生效。
   8. 权限管理做细点的化，可以再增加“新增示例”，“删除示例”等权限
   9. 在页面上，如果某个按钮，希望只有某个权限的人才可以看到，用以下标签  
      <sec:authorize ifAnyGranted="example.delete">  
       <button ...  
      </sec:authorize>
   10. 只在前端做权限控制是不安全的，一定要在后台对应的controller也做控制，比如在AdminExampleController的delete方法，增加如下注解  
       @PreAuthorize("hasRole('example.delete')")