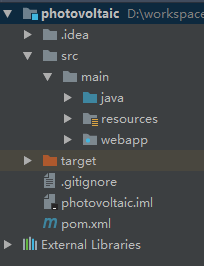
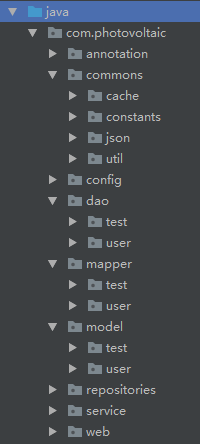
**光伏项目java框架说明**

1. 技术说明
2. maven项目，使用pom管理依赖
3. 采用oracle数据库，单一数据源(GFTW)
4. jdk1.8、tomcat8、springMVC+Mybatis框架
5. guava-cache作为本地缓存(tomcat缓存)工具，用于提升性能(缓存只用于一些不会经常变化的数据存储，变化大的不能用缓存)，主要缓存接口数据，提供了清除界面。重启tomcat会清除
6. 使用redis，主要作用是为了减少对数据库的操作，弥补数据库在高I/O访问时并发的不足，配合使用提高性能。如：存储用户的一些数据
7. 针对4、5两点，大家有高见可以指点并斧正下
8. 使用git作为项目版本控制工具
9. 目录结构说明及规范
10. 结构说明(如右图)
    1. src/mian目录为项目，分别为java、resources、

webapp三大目录

* 1. pom.xml文件为项目依赖管理及默认编译配置中心
  2. 开发过程中需要注意的目录是java目录，不同类似的接口，接下来详细讲解java的目录结构

1. src/main/java目录说明(如右图)
   1. annotation目录为自定义注解目录，如Permission.java
   2. commons目录为一些通用的配置目录，如缓存配置、

通用接口等

* 1. config目录为一些配置目录，如redis集群、缓存定义
  2. dao目录为数据控制层接口目录，根据功能新建子目录，

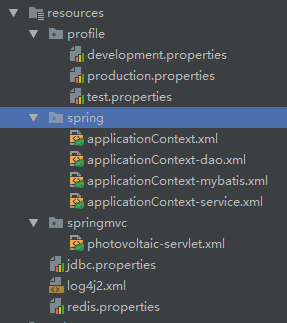
相对应的，mapper目录存放于dao层一一对应的xml

* 1. model目录，为数据库表对应的java类文件
  2. repositories目录暂时不用，因为项目暂时使用单一

数据源

* 1. service目录，业务逻辑层，通常是通过实现interface

详情查看项目，实现类需要加入@Service注解，这样，项目启动时会自动扫描并加载相关bean

* 1. web目录(controller目录)，现在根据不同场景(app/web/admin等)进行了划分，ios/android接口的controller置于appcontroller目录下，admin接口的controller置于admincontroller目录下，依此类推
  2. web/interceptor 目录为拦截器所在目录，不同的拦截器对不同的controller进行拦截，通过在photovoltaic-servlet.xml中配置拦截器拦截的目录，对不同接口的请求进行统一的管理
  3. 

1. 配置文件目录为resources目录，主要是

springMVC、数据源(Oracle)的配置，由

于配置比较复杂，不再进行书面描述，

有疑问可以去百度或者问我

1. 目前前后端完全分离后，webapp目录

暂时搁置不用

1. 示例说明

本示例主要是通过test.html(需要在testcase.js中添加测试样例)页面，请求restful风格的接口，并将返回的数据显示在页面



1. 启动工程后在浏览器输入链接: http://localhost:8080/test.html
2. 选择请求的测试样例(接口URL为: http://localhost:8080/servlet/test/getTestDataList)，请参考TestController.java类，为restful风格的接口，请求相应都是JSON格式数据



1. 接口相应顺序为ControllerMethodInterceptor.java -> TestController. getTestDataList -> TestService. getTestDataList -> TestDAO.getTestDataList
   1. ControllerMethodInterceptor.java为拦截器，会 拦截配置目录的所有方法
   2. TestController为控制层，不实现具体的业务，具体的业务在Service中实现。控制层的一个主要的作用是分离业务、配置接口路径等信息
   3. TestService为业务逻辑层，通过处理查询到的数据库中的数据，并进行处理，返回给Controller并响应给接口调用
   4. TestDAO为持久化层接口，可以通过mabatis的注解在该interface中实现对数据库的操作，也可以结合TestDAOMapper.xml(可以看成是TestDAO.java 接口的implements)实现数据持久化层。注意:本项目使用Oracle数据库，sql方面必要的时候要使用oracle sql语法
2. 图表说明

注意:commons或util 通常代表通用的方法或者存放方法的目录

|  |  |  |
| --- | --- | --- |
| 目录 | 子目录 | 说明 |
| src/main/java/  (com/photovoltaic) | annotation | 自定义注解，如权限验证 |
| commons | cache：缓存管理(web用)及redis操作类(使用redis使用RedisOperator类中的通用方法)  constants:通用的数据类，如返回码  … |
| config | 缓存配置(名称及存活时间)、redis集群配置，后期可能会加入其他的配置项 |
| dao | 数据持久化层，注意分功能分模块进行目录定制，方便管理(如user有关的放入user文件夹，设备有关放入device文件夹下) |
| mapper | dao的xml实现 |
| model | 数据库表的java模型，字段一一对应，分功能分模块 |
| repositories | 废弃 |
| service | 业务逻辑层，分功能分模块 |
|  | web |  |
| web | admincontroller | 管理后台调用的controller置于此目录，同样分功能分模块 |
| appcontroller | app调用的controller置于此目录，分功能分模块 |
| dto | 通常用于存放接口返回的数据模型类，分功能分模块，此folder需要新增 |
| controller | 此目录现在有测试例子 |
| interceptor | 拦截器所在目录 |
| resources | profile | 不同环境下的配置文件(本地启动默认(pom.xml配置)使用development.properties)，不同环境下通过脚本指定不同的文件 |
| spring | 此目录配置文件，配置扫描目录及其他信息，用于扫描并注册bean、…… |
| springmvc | servlet.xml文件 |
| webapp | WEB-INF | web.xml，配置项目启动的一些设置，very important |
| pom.xml |  | 此文件为maven项目的核心文件，用于配置依赖文件、项目默认编译环境以及项目的基本信息 |