光伏项目 java 框架说明

一、技术说明

1、 maven 项目，使用 pom 管理依赖

2、 采用 oracle 数据库，单一数据源(GFTW)

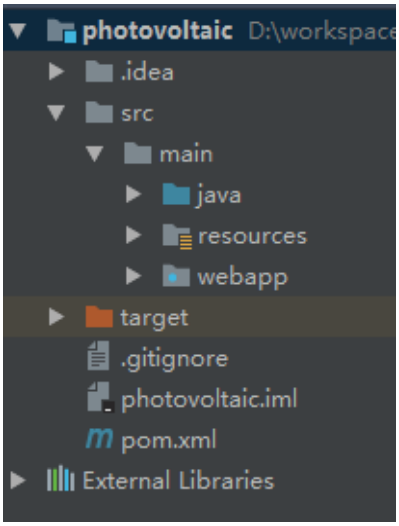
3、 jdk1.8、tomcat8、springMVC+Mybatis 框架

4、 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

5、 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

6、 针对 4、5 两点，大家有高见可以指点并斧正下

7、 使用 git 作为项目版本控制工具



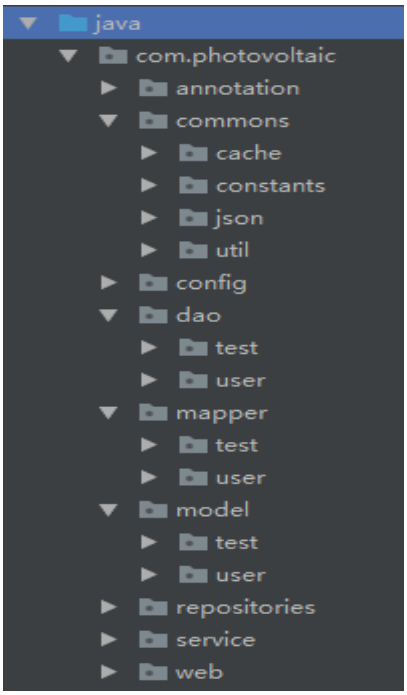
二、目录结构说明及规范

1、 结构说明(如右图)

a) src/mian 目录为项目，分别为 java、resources、webapp 三大目录

b) pom.xml 文件为项目依赖管理及默认编译配置中心

c) 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



2、 src/main/java 目录说明(如右图)

a) annotation 目录为自定义注解目录，如 Permission.java

b) commons 目录为一些通用的配置目录，如缓存配置、通用接口等

c) config 目录为一些配置目录，如 redis 集群、缓存定义

d) 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

e) model 目录，为数据库表对应的 java 类文件

f) repositories 目录暂时不用，因为项目暂时使用单一数据源

g) 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

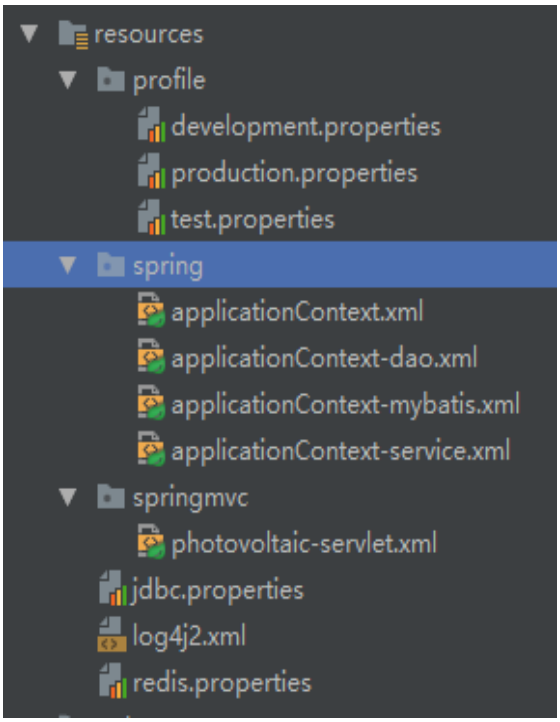
h) web 目录(controller 目录), 现在根据不同场景(app/web/admin 等)进行了划分，ios/android 接口的 controller 置于 appcontroller 目录下，admin 接口的 controller 置于 admincontroller 目录下，依此类推

i) 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

j)

3、 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

4、 目前前后端完全分离后，webapp 目录暂时搁置不用

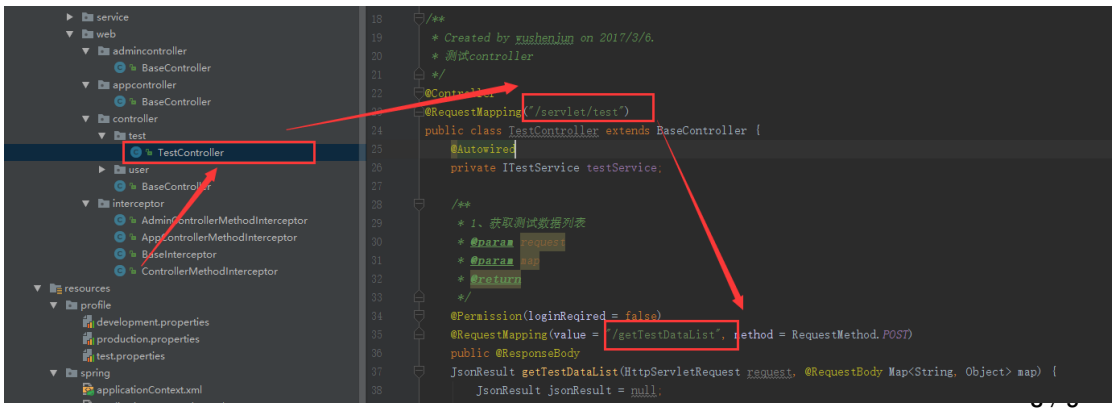


三、示例说明

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



3、 接口相应顺序为 ControllerMethodInterceptor.java -> TestController. getTestDataList -> TestService. getTestDataList -> TestDAO.getTestDataList

a) ControllerMethodInterceptor.java 为拦截器，会 拦截配置目录的所有方法

b) 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

c) 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

d) 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

4、

四、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