框架介绍、使用

1. 框架简介
   1. 来源：<https://gitee.com/wanhaha555/jww>
   2. 技术选型：
      1. ● 核心框架：Spring Boot + Dubbo
      2. ● 分布式协调服务：ZooKeeper
      3. ● 校验框架：Hibernate Validator
      4. ● 安全框架：Apache Shiro
      5. ● 代码生成：MyBatis Plus Generator
      6. ● 持久层框架：MyBatis + MyBatis-Plus
      7. ● 数据库连接池：Alibaba Druid
      8. ● 缓存框架：Redis.clients:jedis
      9. ● 队列框架：Apache ActiveMQ
      10. ● 会话管理：Spring-Session
      11. ● 日志管理：SLF4J
      12. ● 前端框架：Layui
      13. ● 公用工具集：Hutool
      14. ● 支付组件：Egan pay-java-parent
      15. ● 代码简化：Lombok
      16. ● 序列化框架：Alibaba Fastjson
      17. ● HTTP客户端：Hutool-http
      18. ● 接口测试框架：Swagger2
      19. ● 字体图标：Alibaba Iconfont
      20. ● JWT：jsonwebtoken
   3. 扩展内容：
      1. 增加quartz、rabbitmq、七牛云、ELK、shardingjdbc、elasticsearch等功能组件；
      2. 优化代码生成器，增加html字典项标签
      3. 集成docker
2. 工程目录简介
   1. jww-common：公共模块
      1. jww-common-core：基础核心模块，提供MVC基类、异常类
      2. jww-common-db：mybatis-plus配置工程
      3. jww-common-dsession：httpSession缓存配置
      4. jww-common-es：elasticsearch基础配置
      5. jww-common-http：暂时无用
      6. jww-common-jwt：jwt框架的基础配置
      7. jww-common-log：日志配置，提供logback连接elk功能
      8. jww-common-mdb：数据库读写分离配置
      9. jww-common-mq：提供rabbitmq和activemq的基础配置
      10. jww-common-oss：提供阿里云、腾讯云、七牛云对象存储配置
      11. jww-common-pay：集成支付宝，微信支付基础配置
      12. jww-common-redis：集成redis基础配置
      13. jww-common-shardingjdbc：集成shardingjdbc基础配置
      14. jww-common-web：提供web框架安全、异常处理等实现
   2. jww-component：第三方组件安装包
   3. jww-ui：前端UI，基于layui
   4. jww-upm：业务工程
      1. jww-api-server：提供api服务
      2. jww-job-rpc：quartz服务
      3. jww-ump-common：业务公共模块
      4. jww-ump-generator：代码生成器
      5. jww-ump-rpc：后台基础服务模块
      6. jww-ump-server：后端web服务模块
      7. sqls：mysql数据库建库脚本
3. 工程组件介绍
   1. 代码生成器
      1. 模板位于工程目录下resources/templates
      2. 生成内容：列表页html/js，表单页html/js，controller，service，serviceimpl，model，dao，dubbo配置文件，mapper配置文件。
      3. 配置generator.properties：

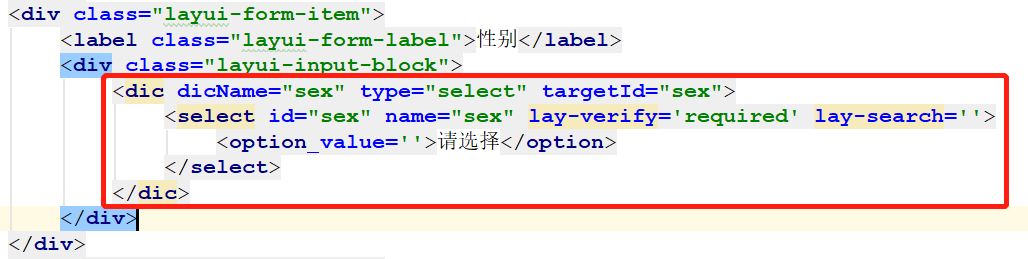
strategy.tableprefix=t\_ #表前缀，该配置由表t\_test生成Test

strategy.include=t\_develop\_customer #本次生成的表名

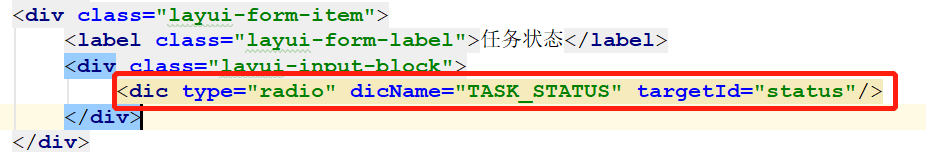
package.parent=com.jww #包配置

project.service=customer #服务名，父模块名

package.modulename=developcustomer #模块名，子模块名

* + 1. 运行CodeGenerator，生成好的文件位于jww-ump-generator工程目录里，复制到对应的工程目录即可。
  1. 页面字典项标签使用
     1. 用于前端字典项字段的展示
     2. 支持select、radio、label，暂不支持checkbox；实现逻辑见base.js。
     3. 使用方法：
        1. select：字典项下拉框

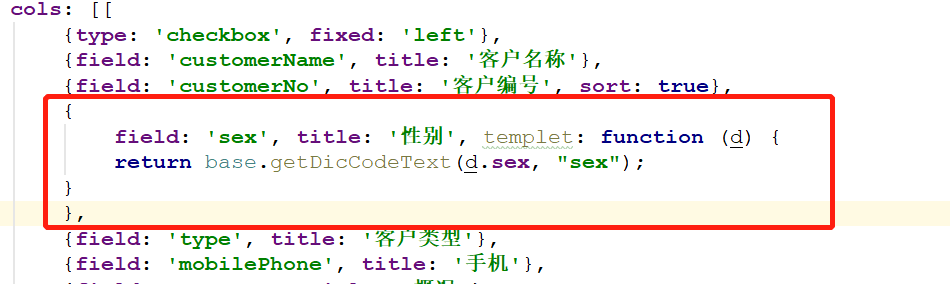
使用标签<dic>，type对应drop，dicName对应系统字典类型，targetId对应dic标签内select标签的id。

* + - 1. radio：单选框

使用标签<dic>，type对应radio，dicName对应系统字典类型。该标签会生成<input type='radio'/>标签，targetId对应其id和name。

* + - 1. label：用于只读页面只显示字典项文本

使用标签<dic>，type对应label，dicName对应系统字典类型，code对应该字典项code（即字典项的key）。

* + - 1. teble中使用字典项

使用templet自定义该cel的模板，调用base.getDicCodeText方法，第一参数是该列的属性值，第二列是对应系统字典类型。