Gamma-user，用户中心服务层

应用框架：Spring-boot，

RPC服务：Dubbo，

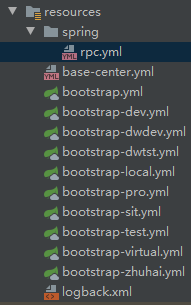
持久层：mybatis, (redis)

Web服务器：tomcat

**Base-center（启动）注册服务**

Package: springboot, dubbo, redis, guava（工具类）, zookeeper

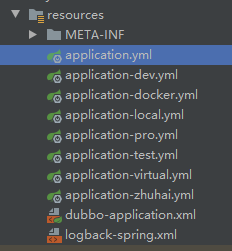
1. 配置文件：



**Gamma-mdm（启动）主数据**

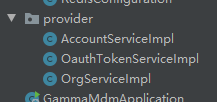
Package:springboot, dubbo, guava（工具类）, zookeeper

1. 配置资源文件



1. Config配置包：@Configuration
2. Provider：服务层

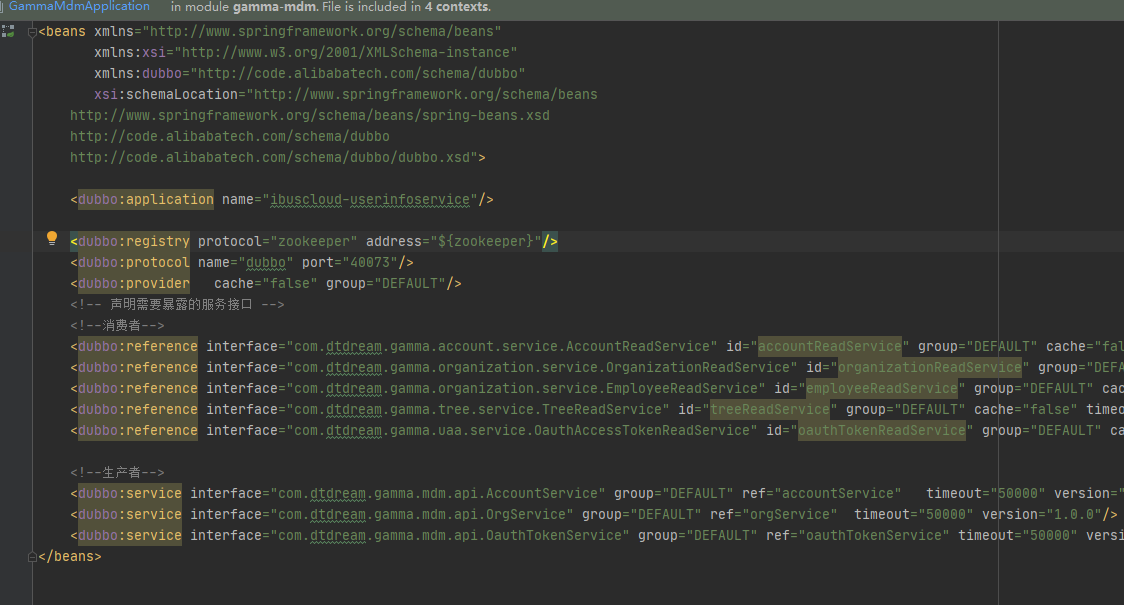
@Service



1. 启动类：里面的注解

@SpringBootApplication(scanBasePackages = {  
 "com.dtdream.gamma.mdm",  
 "com.dtdream.gamma.organization",  
 "com.dtdream.gamma.tree",  
 "com.dtdream.gamma.privilege",  
 "com.dtdream.gamma.account"})

@ImportResource(value = {"classpath:/dubbo-application.xml"}) 。dubbo暴露接口服务xml文件如下：



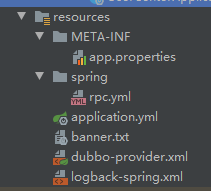
**User-center(启动)用户中心**

Package: springweb

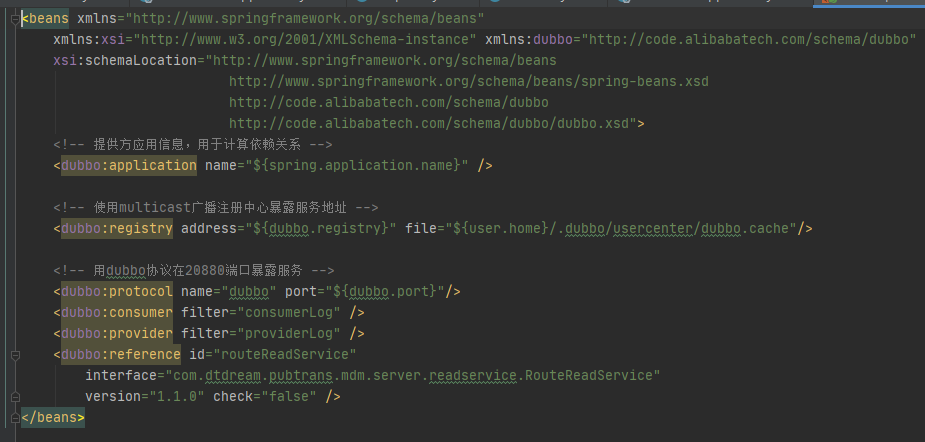
1. Docker：发布工具



1. 配置资源文件：



Dubbo暴露服务dubbo-provider.xml



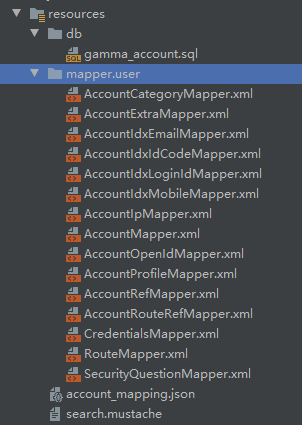
模块举例-账户相关功能

**Gamma-account**（此为实现包，另外有接口包Gamma-account-api）

Package:mybatis, springboot, mysql, redis

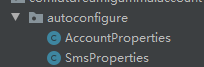
1. 配置资源文件：

Mybatis配置 mapper

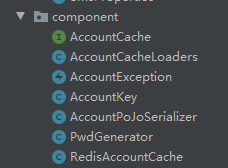


1. 自动配置类：

@ConfigurationProperties(  
 prefix = "user",  
 ignoreInvalidFields = true,  
 ignoreUnknownFields = true  
)



1. Component组件，@Component



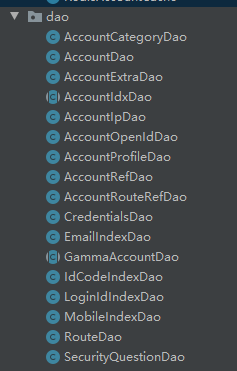
1. Dao层:对数据库的操作，通过mybatis配置SQL执行语句

@Repository

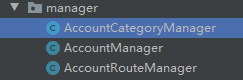
=====================

@Resource(name = sqlSessionFactoryName)

public void setSqlSessionFactory(SqlSessionFactory sqlSessionFactory) {  
 super.init(sqlSessionFactory);  
}



1. 管理：事务管理（原子性），数据读写服务的事务



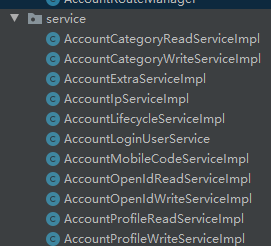
@Component

@Transactional(transactionManager = "defaultUserDataSourceTransactionManager")

1. 服务层：主要是接口的实现类

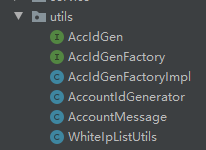
@Service

主要有response方法



1. 工具包：

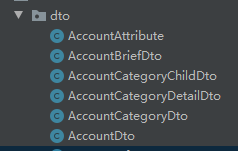
@Component



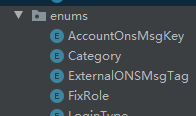
**Gamma-account-api**

1. Dto:??类似于提供一个bean，均实现接口implements Serializable

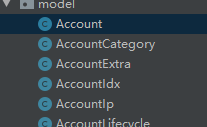
@Data



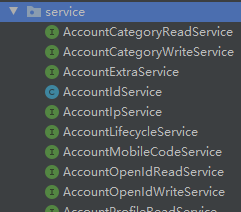
1. 枚举类：提供相关枚举变量



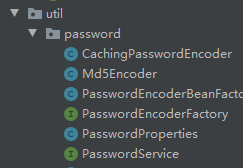
1. Model：对象，提供完整的Bean



1. 接口包：提供服务的接口，（public)



1. 工具包：



**User-notify-client 客户端**

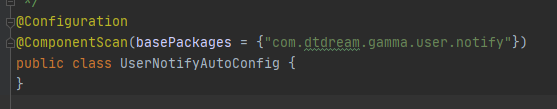
Package：redis, http-request

配置资源文件：这里是配置自动配置类

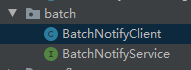


自动配置类



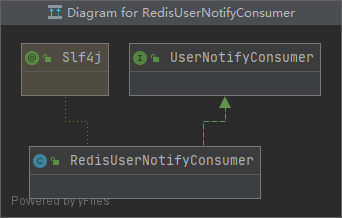


Batch:???批处理消息？？



消费者接口：侦听，实现在下面

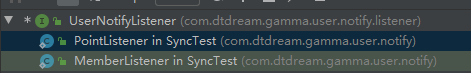




侦听类：



有两个实现，这两个实现在服务器的测试方法中，似乎不是主程序内容



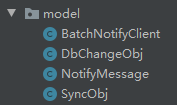
生产者接口：实现在下面



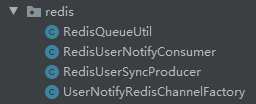
工厂接口：创建生产者和消费者

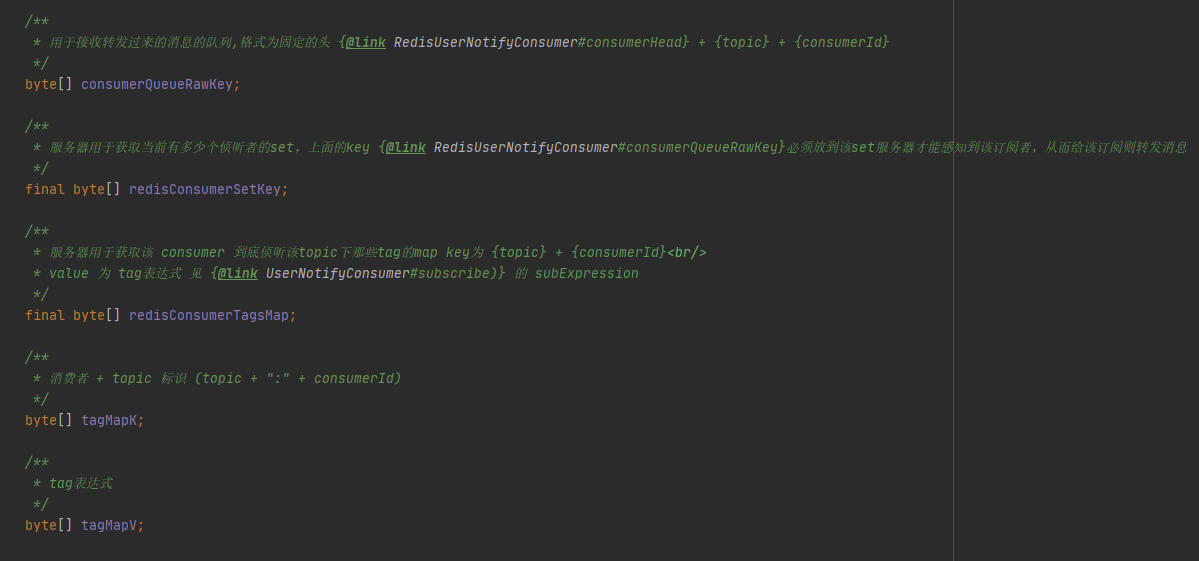


Model:对象，bean



Redis:是上面接口生产者消费和工厂的实现类

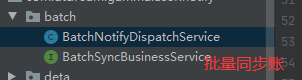




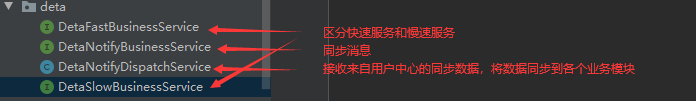
**User-notify-server 服务器**

Package:springboot, redis

Batch:



Deta:???



服务：包含一些过滤操作



同步：

工具包：

