

# 广告系统功能说明

## 一、概述：

1. 首先，广告主在我们的投放系统中投放广告，才会有接下来的广告检索和广告索引，所以广告主是我们系统的顶层设计，即用户设计。
2. 广告投放系统和广告检索系统是该项目的核心，基于 SpringCloud 的技术进行开发。其中，使用 Eureka 进行服务注册和服务发现；使用 Zuul 实现服务网关，作为广告系统总的入口，给我们的广告系统提供服务；使用 Feign 去调用不同的微服务，等等。
3. 广告投放系统的广告数据索引由两个部分去组成，第一个是全量索引，第二个是增量索引。其中，全量索引指的是广告主在我们的广告投放系统中投放的广告，在广告检索系统还没构建完成，读取的一些静态的索引数据。
4. 我们先定义一个位置，到这个位置之后的内容成为动态数据，之前的内容为固定好的数据，称为静态数据。我们先把这部分静态数据导出到静态文件里面，这份文件是从 MySQL 中读取，比如广告主预先投放的一些数据，叫做全量广告数据文件。广告检索系统会读取这份数据文件，建立一份全量索引，或者叫静态索引。之后，广告检索系统“伪装”成一个 slave，绑定到 MySQL 上面，监听到 MySQL 的 blob，然后去建立增量索引。这个增量索引是指广告主在监测系统的过程中，对广告数据进行修改，删除，增加，等等。这部分数据就是一份动态数据，索引称为增量索引。

5. 然后用这份数据去实现广告检索服务, 建立索引的目的是为了加快检索服务。

媒体方会去对接我们的检索系统, 然后对我们的检索系统发送请求。广告检索系统收到请求之后, 对请求进行解析, 实现检索服务, 对照着请求信息去索引数据里面检索数据, 最终获得媒体方想要的广告数据。

6. 广告主投放的数据本身具有广告计费方式, 比如 CPM, CPC, CPT 等等。

广告系统的核心功能就是广告投递服务(ad-sponsor)和广告检索服务(ad-search), 主要就重点分析这两个模块的实现与功能, 对于其他非核心功能就只进行一个简单的分析和描述说明。

主要功能模块参照下图:



图 1 工程结构图

## 二、非核心功能、模块介绍：

### 1. ad-eureka、ad-gateway、ad-dashboard

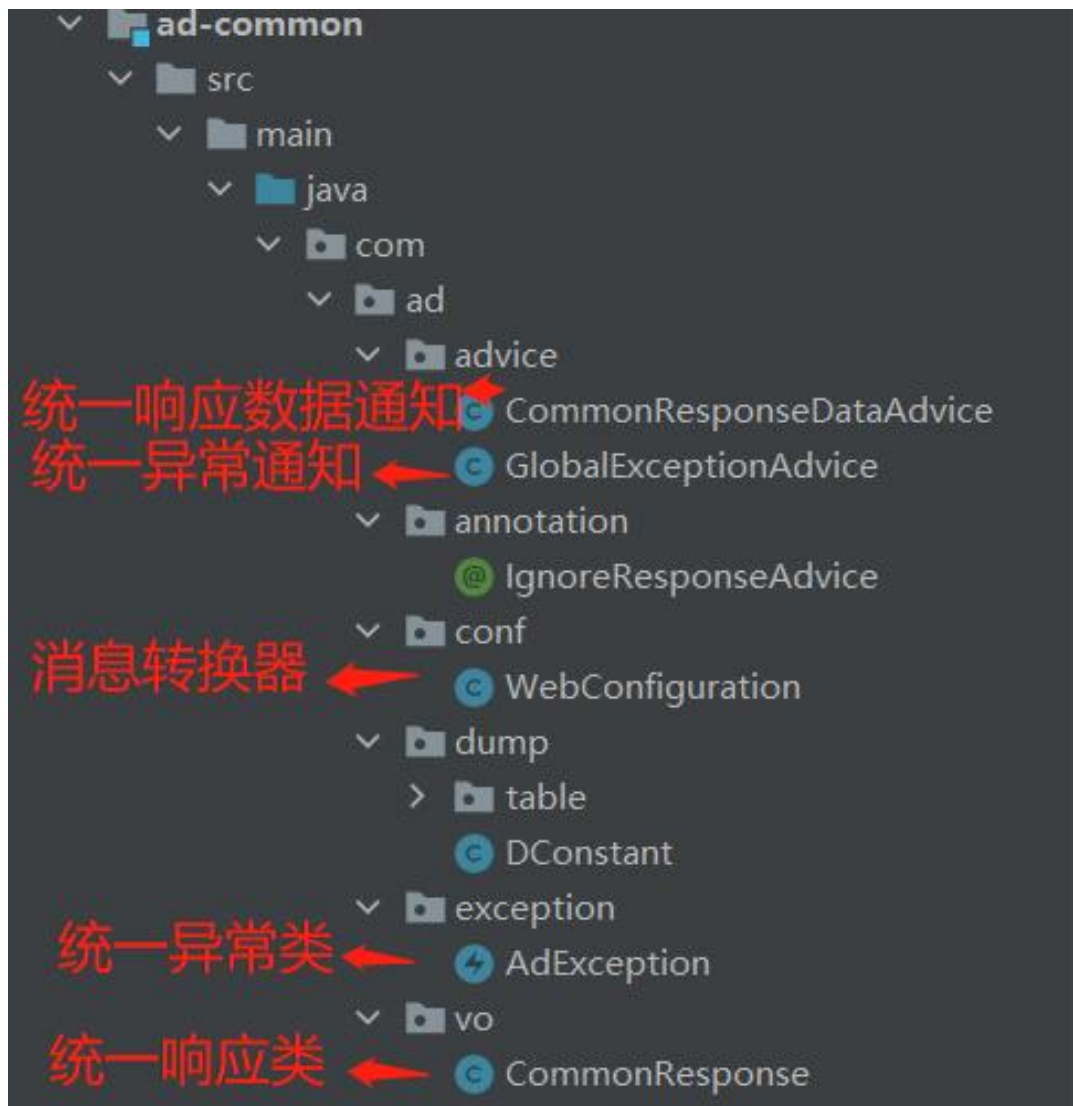
ad-eureka 主要是用于启动 eureka，提供服务注册和服务发现功能。

ad-gateway 主要用于启动网关 zuul，网关通常的用途有用户鉴权、限流、过滤等，但是本项目并没有涉及到用户注册登录的模块，因此这里的网关仅仅被我用来统计一个请求经过网关前后过滤器的耗时。

ad-dashboard 主要用于监控服务之间的调用，发生调用请求时，打印相关的日志供我们使用。

### 2. ad-common

ad-common 是通用模块服务，在该模块下，我们会定义统一请求响应处理、统一全局异常处理、消息转换器、数据库表持久化为后缀为 .data 的磁盘文件（ad-search 模块会详细说明）等通用功能。

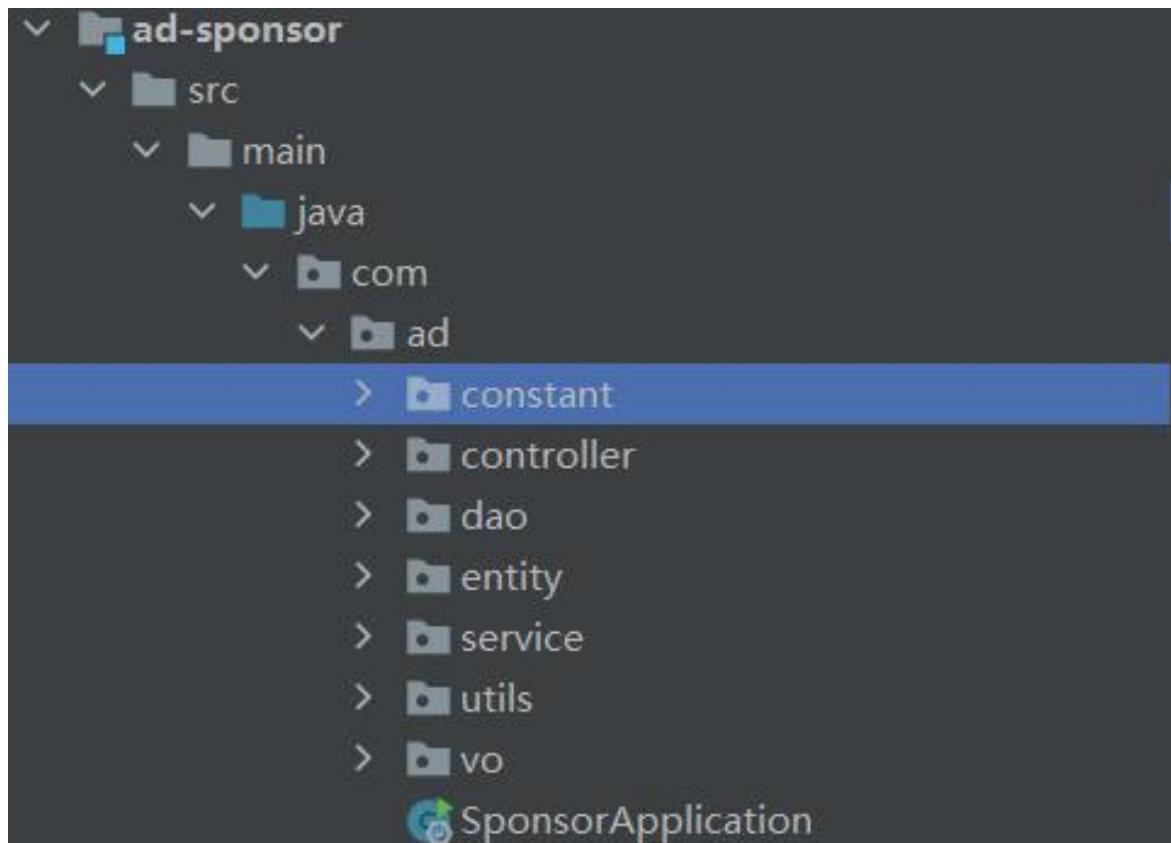


### 三、核心功能、模块介绍：

#### 1. ad-sponsor

该模块的功能主要是提供给广告商的，用于投放广告到系统中，投放的广告数据将被存储到 MySQL 中，因此该功能主要实现关于广告数据的增删改查功能。

工程结构：



## 数据库表结构:

用户账户 ( ad_user )	含义
username	账户名称
token	账户 token
user_status	账号状态
create_time	创建时间
update_time	更新时间

推广单元 ( ad_unit )	含义
plan_id	关联推广计划id
unit_name	推广单元名称
unit_status	推广单元状态
position_type	广告位类型 ( 开屏、贴片等 )
budget	预算
create_time	创建时间
update_time	更新时间

推广计划 ( ad_plan )	含义
user_id	标记当前记录所属用户
plan_name	推广计划名称
plan_status	推广计划状态
start_date	推广计划开始时间
end_date	推广计划结束时间
create_time	创建时间
update_time	更新时间

关键词限制 ( ad_unit_keyword )	含义
unit_id	关联推广单元id
keyword	关键词

地域限制 ( ad_unit_district )	含义
unit_id	关联推广单元id
province	省
city	市

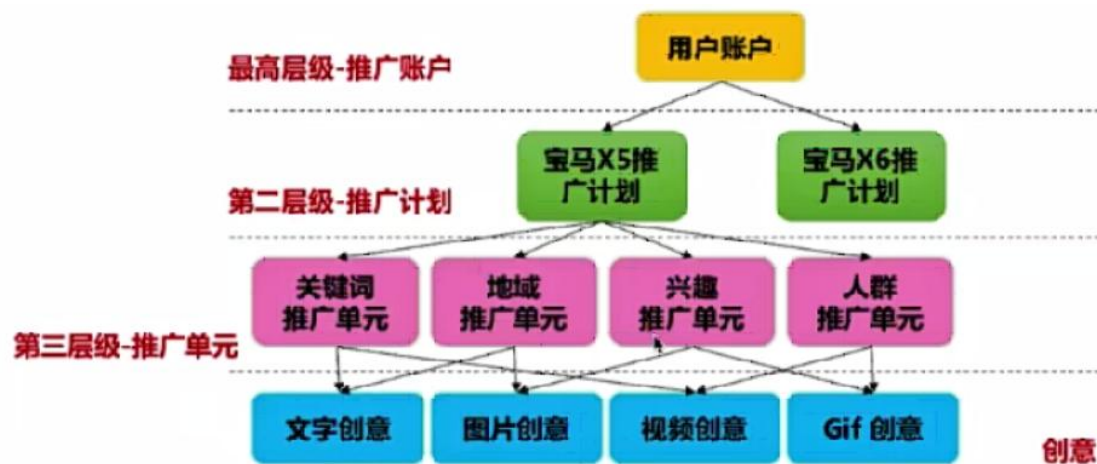
兴趣限制 ( ad_unit_it )	含义
unit_id	关联推广单元id
it_tag	兴趣标签

创意 ( ad_creative )	含义
name	创意名称
type	物料类型 ( 图片、视频 )
material_type	物料子类型 ( bmp、avi )
height	高度
width	宽度
size	物料大小, 单位 KB
duration	持续时长, 视频不为 0
audit_status	审核状态
user_id	标记当前所属用户
url	物料地址
create_time	创建时间
update_time	更新时间

创意与推广单元关联 ( creative_unit )	含义
creative_id	关联创意id
unit_id	关联推广单元id

## 广告详细数据说明：

根据上面的表和字段我们详细说明一下投放的广告包含了哪些数据。



### (1) 广告的推广账户

推广账户用来唯一标识广告商，能够表示出相应的广告属于哪些广告商投递的。

### (2) 广告的推广计划

推广计划就是一个广告的总体方向，比如上图中推广计划可以是宝马x5 推广计划，也可以是奔驰推广计划等等。

### (3) 广告的推广单元

推广单元是一个推广计划的下一级划分，用来说明一个推广计划的对

象和地域等等。比如上图的宝马 x5 推广计划，可以设置为只在北京、上海、广州推广，面向的人群是爱车人士。可以很明显看出一个推广计划可以对应多个推广单元。

#### （4）广告的创意

广告创意的作用就是用来吸引他人点击查看广告的，每个推广单元可以对应到多个广告创意。

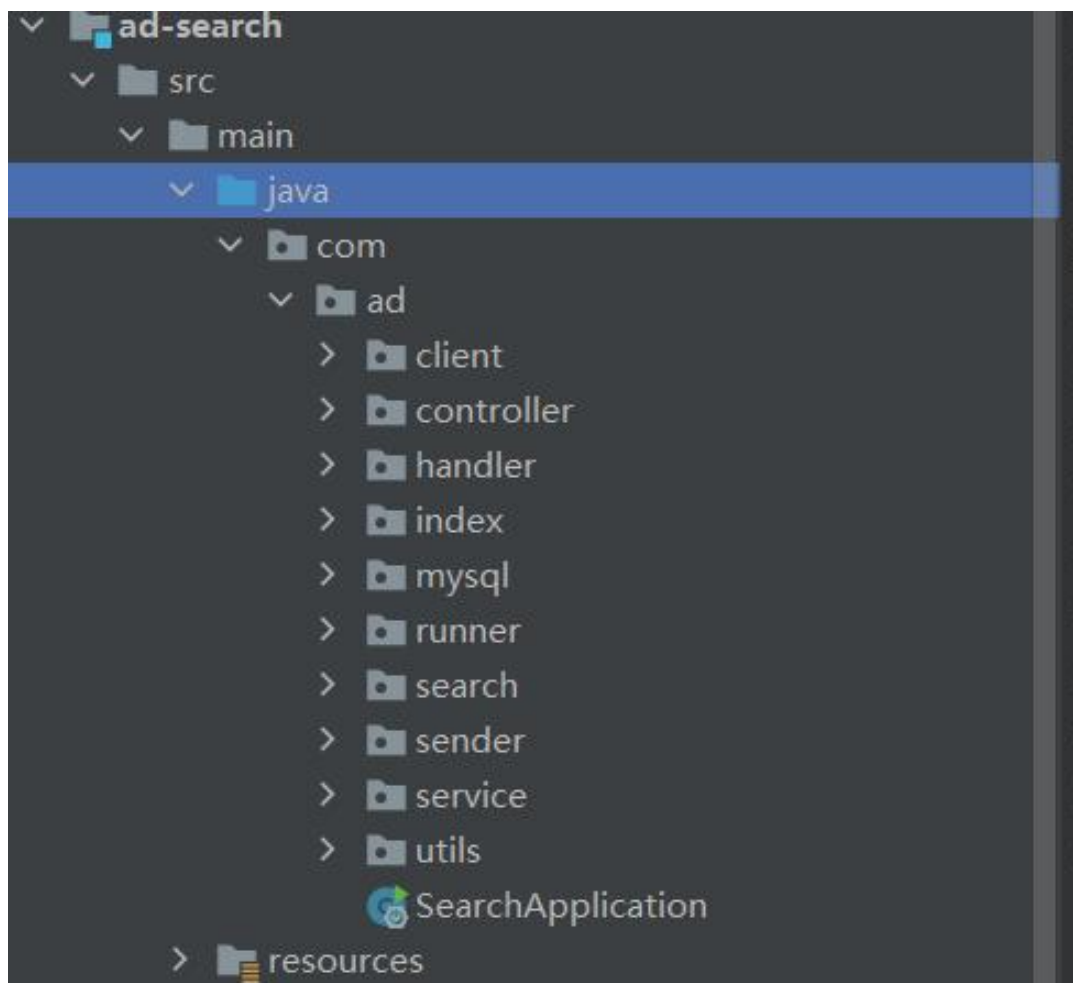
详细代码主要是增删改查多，就不加以说明了。

### 3. ad-search\*

ad-search 模块是广告的检索系统，不仅非常重要，编码难度也比较高，其中**最重要的就是对于全量索引的构建、还有对 MySQL Binlog 的监听来构建增量索引。**

#### （1）全量索引和增量索引的存储设计

工程目录：



从工程目录中可以看到有非常多个包，下面详细说明核心的功能的实现和作用：

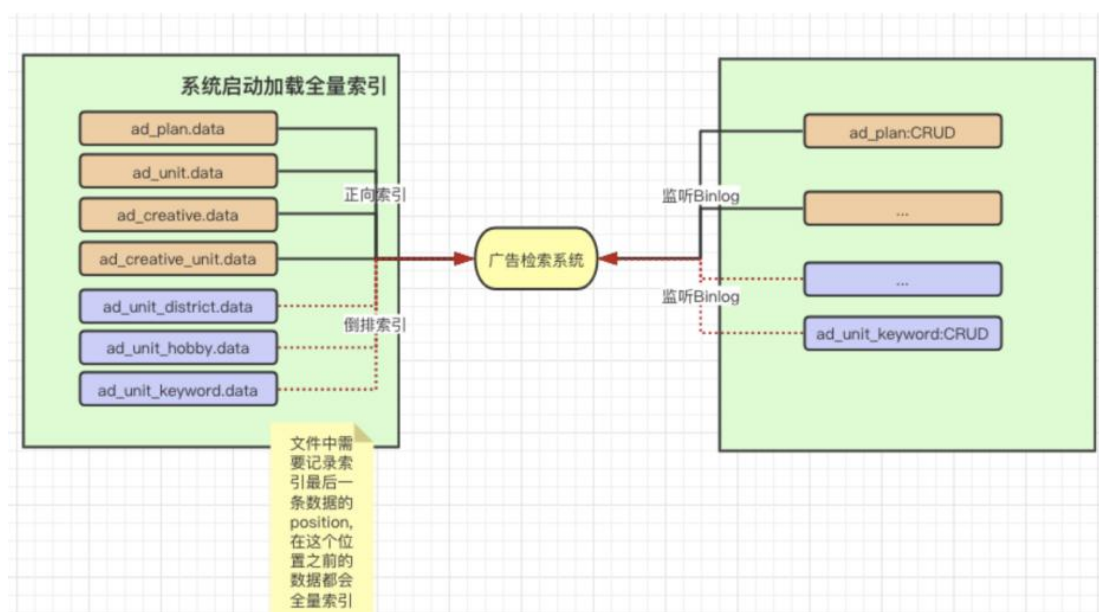


图 2 图示为正向索引、倒排索引



对于广告检索系统，需要构建**正向索引和倒排索引**，而且广告数据被修改的可能性很大，而检索系统又没有对应的数据库（为了实现与投放系统的解耦），因此我们可以监听 MySQL Binlog，然后通过编码的方式加载到检索系统中，我们的整体思路如下：

当广告商投放广告数据后，广告数据存储到 MySQL，同时投放系统会把 MySQL 中的数据以 json 形式 dump 到指定目录文件下，以 .data 后缀结尾，检索系统通过编码加载这些 json 数据，用 concurrentHashMap 结构构造出正排索引（为了防止多线程安全问题），因为检索系统需要速度快，因此直接编码把索引实现加载到 jvm 内存中，通过 key-value 的方式就可以像“MySQL”一样查询数据，其中 id 作为 key（对应到数据库一行记录的 id），而字符串值作为 value（对应到数据库一行记录具体字段的值）。

很明显，我们的正向索引适用于**推广计划**、**推广单元**和**创意**这几张表的数据上，因为广告检索的请求信息，不可能是请求具体的**计划**或**推广单元**，它的检索请求一定是限制条件。

对于推广单元，通常需要通过推广单元的关键词限制去检索相应的推广单元，这就需要我们自己设计编码实现倒排索引。



#### 全量索引

检索系统在每次启动的时候，需要一次性读取当前数据库中的所有数据，建立索引。

**TIPS：**（这里有一个问题，如果我们系统流量高的时候，需要部署多实例，每一个实例在系统启动的时候，都直接从数据库中加载全量索引，会对数据库造成极大的压力。）

因此，通常我们实现全量索引都是通过读取文件中预先存储的广告数据来实现全量索引导入。

#### 增量索引

顾名思义就是后来新增的数据索引。系统在运行过程中，会对数据库的变动进行监控，实现加载更新，构建索引，我们系统中通过对数据库表的CRUD操作的变动，通过监听Binlog来实现增量索引的更新。

### 编码实现：

我们分析了广告索引的维护有 2 种，**全量索引加载**和**增量索引维护**。

#### （2）binlog 的监听和增量索引的构建

包括解析模板、监听器、投递增量数据的发送器

后续慢慢更新....