

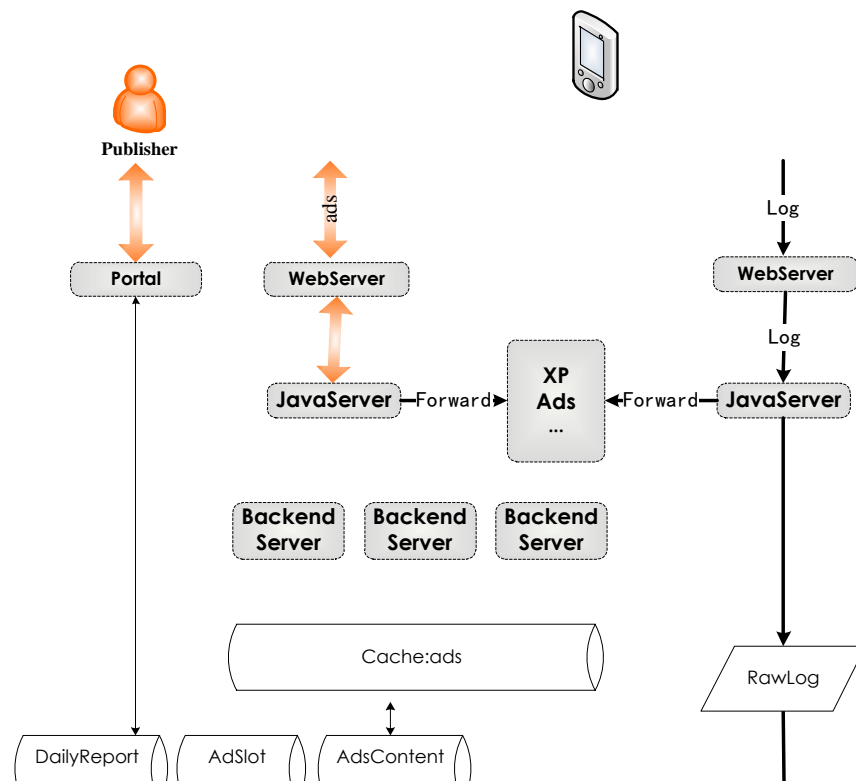
# UFP 设计文档

## 1. 系统设计目标:

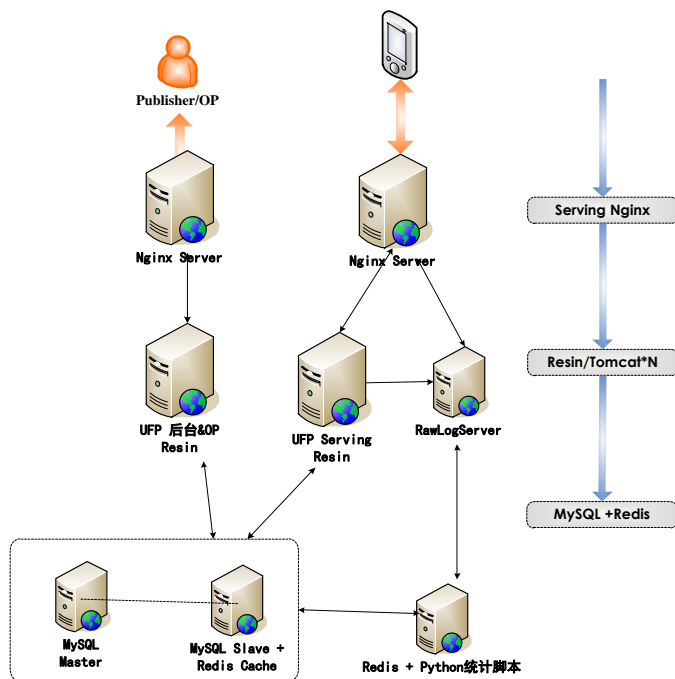
支撑每天 1000 万 IMEI 用户的访问，每天 3000 万次请求  
支持每天 6000 万的 log 日志

## 2. 系统架构

部署结构如下，初期需要 4 台服务器，后续压力增大时可添加更多 JavaServer。



部署结构图



典型四层部署结构：

Nginx  
Resin  
Mysql5  
Redis  
JDK6  
Ubuntu

改进点：

- a) 使用 Nginx，更快更稳定
- b) 使用 Resin Java server
  - i. 更新稳定，速度是 Tomcat 的 3 倍以上
  - ii. 并发处理、启动停止机制优于 Tomcat
- c) 加入 Redis Cache server，尽量减少 Mysql 的压力，保证数据处理速度。

### 3. SDK 前后端交互协议

参照 google docs 文档：

<https://docs.google.com/a/umeng.com/document/d/1stj-fKwE-IO2k0QlpS440U2zPZvR08M0DKnof4L-kGc/edit>

改进：

- (1) 目前协议传输使用 POST 请求，考虑改为 Get 请求（URL 最长支持为 1024Byte，可满足需求）。

好处：便于 Nginx/Apache 记录日志到 Raw Log；便于调试分析；Post 请求并不能避免 Spam。

待讨论：如何将数组扁平化，多 promoters 中合并 action

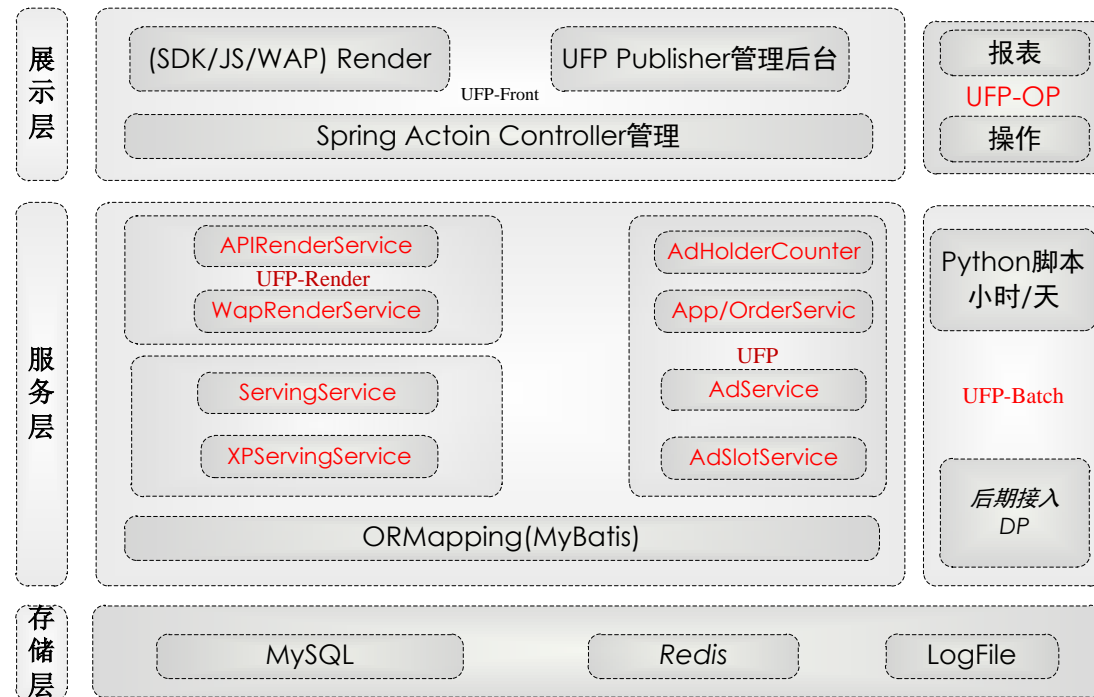
- (2) 服务端返回数据格式中加入更多参数，目标是将配置后台化。

#### 4. 后台系统模块设计

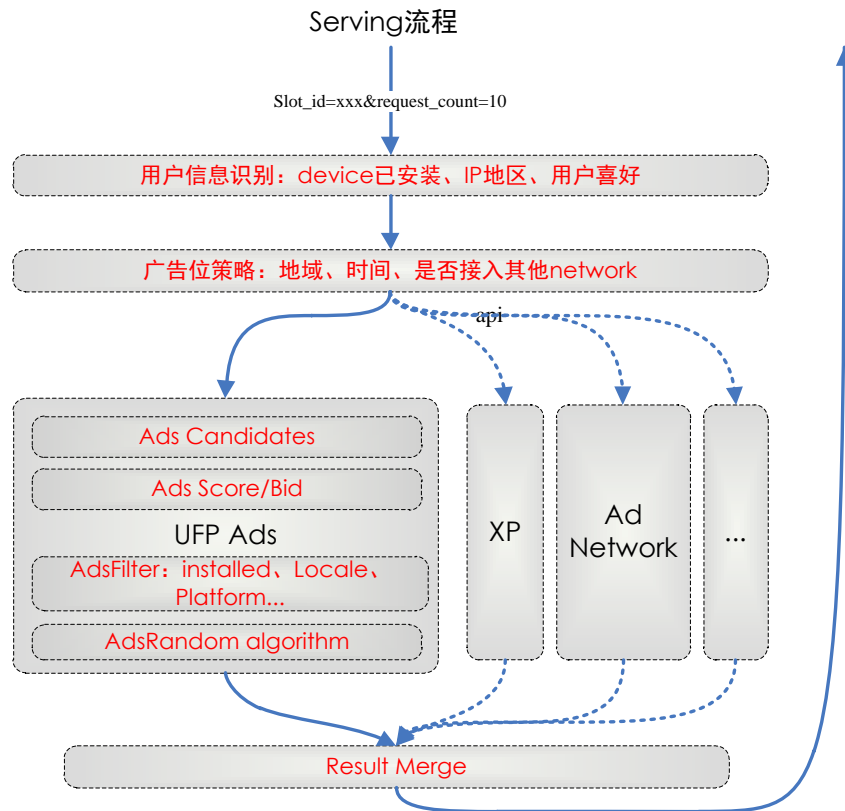
典型的 Spring MVC 三层结构，系统并不复杂，拿过来直接可以用。

和我们目前基于 rails 交换架构类似，但由于 Java 项目模块划分更容易也更清晰一些。

##### a) 系统模块设计



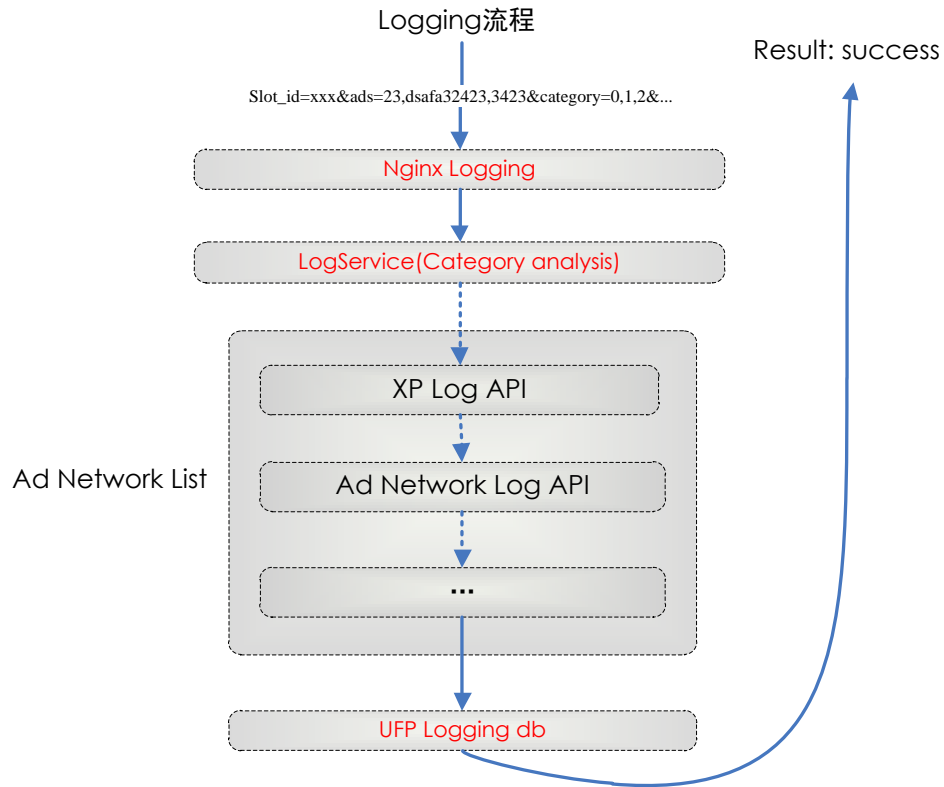
##### b) UFP Serving 流程设计



接入第三方广告网络流程:

1. 服务器端根据广告位的设置策略，如果有接入广告网络，将 `ad_network_key`、`request_count`、`layout` 和用户设备信息等通过 API 调用。
2. 将广告网络返回的结果字段 mapping 到 ufp 协议，同时对每条广告追加一个 `category`(0: 交换、1 交换自主、2 交换付费、21 UFP、31 umeng ads)，然后将结果返回给 SDK
3. SDK 正常展示所有广告。

c) UFP Logging 流程设计



#### Logging 流程

1. ufp Nginx 打印 log 到文件中。(ufp 的 batch 根据此文件定时分析文件入库到 mysql)
2. UFP LogService 根据 Log 中的 category 识别是否含有非 UFP 广告，如果有将 log（只包含广告网络的 ads）转发给各广告网络。
3. 返回 200 成功结果给 SDK。

#### TODO:

计划 Q2 开始，Logging 转发一份给数据平台保存。后续再作分析。

#### d) 技术路线

JDK6

JSTL+HTML+jQuery

Spring Action

Spring Core + Transaction

MyBatis (ORM)

Mysql 5

Redis

Python

#### 未来改进:

- i. 考虑使用 Google Guice 替换 Spring，超轻量级的依赖注入容器、速度快代码少，很容易进行单元测试。(liqun)
- ii. 前端可以考虑 Google GWT，Java 代码来替代 javascript，便于实现和维护复杂

的前端逻辑。(hai)

## 5. 数据统计模块

- Raw Log 记录在日志文件，以小时进行切割，文件名如 request\_2011101610.log。
- 初期使用 python 或 ruby 脚本进行日志解析入库，每小时一次进行累加。先计算 AdReportSummary、AdReportPromoterSummary、AdReportDetail，再汇聚到 AdSlot、AdOrder、App、Account。
- 每日对 RawLog 进行重新计算一次，覆盖当天计算的数据。
- Raw Log 入库，并追加附加信息。(MyISAM)
- 

Issues:

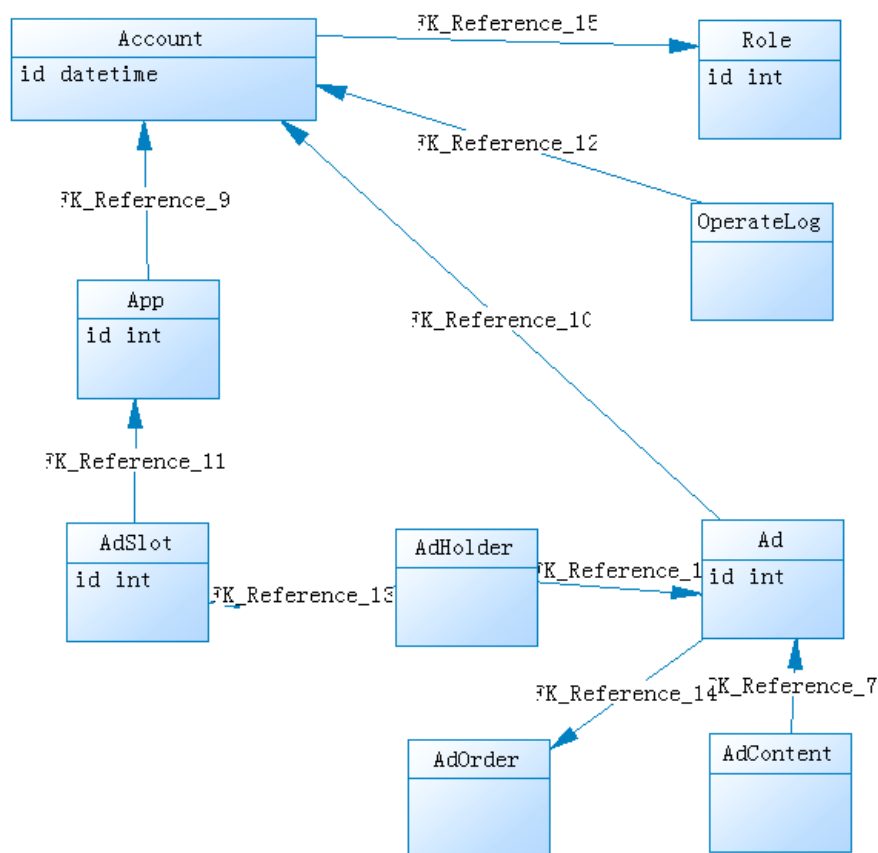
- 测试期间如何快速看到数据

## 6. 数据库详细设计

数据库我们使用 MySQL，因为除日志外数据规模不大，稳定性和性能更可靠，运维和优化技术相对成熟。存储引擎使用支持事务的 InnoDB，保证广告系统数据的一致性。

### a) 数据库 ER 关系设计

账户和广告信息相关表



a) 数据库表字段设计

由于 UFP 目前只面对 Publisher，用户角色较少，因此权限系统设计相对简单。

```
Create database umeng_ufp;
Use umeng_ufp;

create table account
(
  id int(11) NOT NULL auto_increment comment '自增的 ID',
  created_at datetime,
  updated_at datetime,

  email varchar(64) not null comment '账户地址',
  password varchar(64) not null comment '密码，MD5 加密后',
  type enum('admin','admin_view','admin_op','publisher') not null comment '用户组',

  name varchar(32) comment '账户名称',
  phone varchar(32) comment '电话或手机',
  info varchar(128) comment '更多信息',
  logo_url varchar(128) comment '账户定制的 Logo 地址',

  last_login_time datetime,
  login_times int,

  PRIMARY KEY (`id`)
) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

create table role
(
  id int(11) NOT NULL auto_increment comment '自增的 ID',
  created_at datetime,
  updated_at datetime,

  type enum('admin','admin_view','admin_op','publisher') not null comment '用户组',
  permission varchar(128) comment '权限设置',

  PRIMARY KEY (`id`)
) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

create table app
(
  id int(11) NOT NULL auto_increment comment '自增的 ID',
  created_at datetime,
```

```

updated_at datetime,

app_key varchar(32) NOT NULL comment '应用的 key 与 umeng 主站对应或新的',
platform enum('android','ios') NOT NULL comment 'android/ios',
name varchar(32) not null comment '应用名称',

PRIMARY KEY (`id`)
) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

create table ad_slot
(
    id int(11) NOT NULL auto_increment comment '自增的 ID',
    created_at datetime,
    updated_at datetime,

    app_id int NOT NULL comment '所属应用',
    name varchar(32) not null comment '广告位名称',
    size_landing varchar(32) comment '入口尺寸',
    size_content varchar(32) comment '内容尺寸',
    platform enum('android','ios') NOT NULL comment '平台',

    style varchar(32) comment '样式',
    device varchar(32) comment '子平台、适用设备',
    channels varchar(32) comment '适用渠道',
    areas varchar(32) comment '适用地域',
    timeslots varchar(32) comment '适用时段',
    display_rate varchar(32) comment '出现概率',

    enable_xp varchar(32) comment '是否接入交换',
    xp_rate int comment '交换比例',
    xp_id int comment '交换 ID',
    enable_adnet varchar(32) comment '是否接入广告网络',
    adnet_rate int comment '广告网络比例',
    adnet_id int comment '广告网络 ID',

    status varchar(32) NOT NULL comment '状态',

    PRIMARY KEY (`id`)
) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

create table ad
(
    id int(11) NOT NULL auto_increment comment '自增的 ID',
    created_at datetime,

```



updated\_at datetime,

name varchar(32) not null comment '广告名称',

size varchar(32) comment '尺寸',

type varchar(32) comment '类型',

ad\_order\_id int comment '所属订单的 id',

prior varchar(32) comment '优先级',

start\_time datetime,

end\_time datetime,

device varchar(32) comment '子平台、适用设备',

channels varchar(32) comment '适用渠道',

areas varchar(32) comment '地域',

network varchar(32) comment '适用网络',

keywords varchar(32) comment '关键词',

landing\_type varchar(32) comment '打开方式',

timeslots varchar(32) comment '适用时段',

budget varchar(32) comment '预算',

status varchar(32) NOT NULL comment '状态',

PRIMARY KEY (`id`)

) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

create table ad\_holder

(

id int(11) NOT NULL auto\_increment comment '自增的 ID',

created\_at datetime,

updated\_at datetime,

ad\_id int not null comment '广告 ID',

ad\_slot\_id int NOT NULL comment '广告位的 ID',

PRIMARY KEY (`id`)

) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

create table ad\_content

(

id int(11) NOT NULL auto\_increment comment '自增的 ID',

created\_at datetime,

updated\_at datetime,

platform varchar(32) NOT NULL comment '平台',

```
title varchar(32) NOT NULL comment '应用名称',
ad_words varchar(32) comment '广告语',
description varchar(512) comment '描述',
provider varchar(32) comment '开发商',
icon varchar(32) comment '图标地址',
url varchar(32) comment '下载地址',

size int comment '大小',
price varchar(32) comment '在商店的标价, 默认为 0',

app_package_name varchar(32) comment "",
app_version_code varchar(32) comment "",
app_version_name varchar(32) comment "",
app_store_id varchar(32) comment "",

PRIMARY KEY (`id`)
) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

```
create table ad_order
(
    id int(11) NOT NULL auto_increment comment '自增的 ID',
    created_at datetime,
    updated_at datetime,

    start_at datetime,
    end_at datetime,

    name varchar(32) NOT NULL comment '广告公司名称',
    contact varchar(32) NOT NULL comment '广告公司联系人',

    PRIMARY KEY (`id`)
) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

```
create table user_op_log
(
    id int(11) NOT NULL auto_increment comment '自增的 ID',
    created_at datetime,
    op_date date,
    op_time time,

    account_id int comment '账户的编号',

    ad_id int comment '广告的编号',
    ad_order_id int comment '广告订单的编号',
```

```

ad_slot_id int comment '广告位的编号',
app_id int comment '应用的编号',

admin_id int comment '管理员 id, 如果不是管理员操作, 就显示 0',
op_content varchar(32) comment '1、流量 2、推广类型 3、推广样式 4、应用名称/网站名称 5、
推广文字/网站简介 6、详细描述 7、开发商 8、icon 地址/网站 logo 地址/图片地址 9 apk 地址/appstore
地址/网页地址 10、跳转方式 11、优先级 12、是否参接入 13、启用/暂停 14、广告价格 15、付费方式
16、投放网络 17 投放时段 18、投放时间 19、预算 20、入口类型 21、入口尺寸 22、内容尺寸 23、
渠道 24、出现概率 25、广告类型 26、名称 26、所属应用 27、所属订单 28、关键词 29、适用设备
30、打开方式 31、地域',

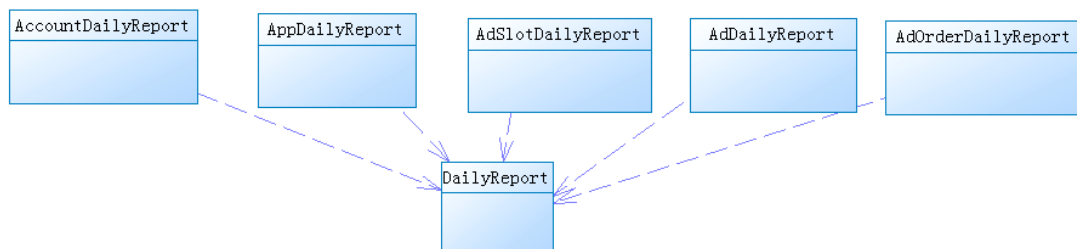
op_type varchar(32) comment '1: 新增 2: 删除 3: 修改 4: 暂停 5: 启用',
ad_type varchar(32) comment '1、交换 2、友盟广告 3、付费应用 4、自销广告',
ad_subtype varchar(32) comment '1: 广告位 2: 广告 3、应用 4、订单 5、账户',
old_value varchar(32) comment '修改前的值',
new_value varchar(32) comment '修改后的值',

PRIMARY KEY (`id`)
) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

```

外键：优点：缺点：

索引：



DailyReport 分三个表，分别为 publisher\_summary, promoter\_summary, detail

```

Report
create table report_publisher_summary
(
    id int(11) NOT NULL auto_increment comment '自增的 ID',
    created_at datetime,
    updated_at datetime,

    publisher_id int NOT NULL comment '媒体 Id',
    date_s date,

    impression int not null default 0,
    click int not null default 0,
    download int not null default 0,

```

```

appstore_click int not null default 0,

PRIMARY KEY (`id`)
) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

create table report_promoter_summary
(
    id int(11) NOT NULL auto_increment comment '自增的 ID',
    created_at datetime,
    updated_at datetime,

    promoter_id int NOT NULL comment '被推广的 ID: 广告、广告位、应用、订单',
    date_s date,

    impression int not null default 0,
    click int not null default 0,
    download int not null default 0,
    appstore_click int not null default 0,

    PRIMARY KEY (`id`)
) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

create table report_detail
(
    id int(11) NOT NULL auto_increment comment '自增的 ID',
    created_at datetime,
    updated_at datetime,

    publisher_id int NOT NULL comment '媒体 Id',
    promoter_id int NOT NULL comment '被推广的 ID: 广告、广告位、应用、订单',
    date_s date,

    impression int not null default 0,
    click int not null default 0,
    download int not null default 0,
    appstore_click int not null default 0,

    PRIMARY KEY (`id`)
) ENGINE= InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

```

## 7. 工作和人员计划

10.10-10.15 一周 系统设计、数据库设计 (ke、hai)

10.17-10.29 二周 搭系统框架、模块、数据读写模块 (ke)  
10.31-11.19 三周 UFP 交互实现, 数据统计实现, 功能实现 (Ke+FTE+Intern)  
11.10-11.25 两周 开始和 SDK 连调, 基于交换的 SDK  
11 月底 一期版本可用

人员计划:

11 月前务必到岗一个 Java 工程师, 主要做 UFP 的前端编程。

11 月初到岗一个 Python/Ruby 实习工程师(maybe 鄢舒元), 主要用于对 UFP 的日志进行统计, 核对数据。(目标是保证数据无误, 后期量起来后会迁移到 Haddop 平台计算)