|  |  |  |
| --- | --- | --- |
| 文档状态 | 文档编号 |  |
| [ ] 草稿 | 当前版本 | 3.2.5 |
| [√] 正式发布 | 作者 | 邓伟鹏，陈晓明，龚永平 |
| [ ] 正在修改 | 完成日期 | 2017年12月4日 |

# Pxene RTB接口协议

|  |  |  |  |
| --- | --- | --- | --- |
| 修改时间 | 作者 | 版本号 | 要点 |
| 2018/1/4 | 李春光 | 3.2.6 | 1. 修改“[投放策略10分钟未完成目标](#_投放策略_10分钟未完成目标(SET))”redis表的field值； 2. 增加“[自动出价计数](#_自动出价计数（HASH）)”表 |
| 2017/12/11 | 邓伟鹏 | 3.2.5 | 1. 增加对“[获取单个创意的详细信息](#_获取单个创意的详细信息（STRING）)”中“w”、“h”两个字段的文字描述； |
| 2017/12/04 | 邓伟鹏 | 3.2.4 | 增加“[投放策略10分钟未完成目标](#_投放策略_10分钟未完成目标(SET))”redis表 |
| 2017/11/17 | 邓伟鹏 | 3.2.3 | 修改自动投放相关部分：   1. “[投放策略10分钟计数](#_投放策略10分钟计数（HASH）)”中增加当前十分钟时段出价次数和出价累计金额； 2. 修改“[投放策略10分钟计数](#_投放策略10分钟计数（HASH）)”中的field为区分定价和竞价； 3. 修改重要功能用法中“[自动出价调整](#_自动出价调整)”部分的传递参数和相关说明； |
| 2017/11/17 | 邓伟鹏 | 3.2.2 | 1. 去掉“允许无设备ID的流量”key； 2. 在“[单个投放策略的定向信息](#_获取单个项目的定向信息)”中增加nodeviceid列表段； 3. 修正了“[单个投放策略的定向信息](#_获取单个项目的定向信息)”例子中scene段的key名。 |
| 2017/11/15 | 邓伟鹏 | 3.2.1 | 修改自动投放相关部分：   1. “[单个投放策略的基本信息](#_单个投放策略的基本信息(STRING))”中增加autobid属性 2. 修改“[投放策略10分钟计数](#_投放策略10分钟计数（HASH）)”的imp和clk域 3. “[自动出价调整](#_自动出价调整)”说明中增加出价计算通知的参数“allowtime”； 4. “[单个创意出价信息](#_单个创意的出价信息（HASH）)”中关于价格单位的描述更正为 ：人民币 分/CPM |
| 2017/11/10 | 邓伟鹏 | 3.2.0 | 增加自动投放相关部分：   1. redis投放计数（集群）中增加”[投放策略10分钟计数](#_投放策略10分钟计数（HASH）)”； 2. redis投放计数（集群）中的”[活动及策略点击展示价格计数](#_订单及策略点击展示价格计数（HASH）)”中去掉“10m\_imp\_p\_(policyid)\_(10分钟段)”和“10m\_clk\_p\_(policyid)\_(10分钟段)”域； 3. 重要功能用法→[匀速投放](#_匀速投放)→清除缓存定时器（JAVA）中去掉“HDEL key field 命令 从每小时的05分起每隔10分钟从“活动及策略点击展示价格计数”中删除前一个10分钟时段计数数据。”并修改计数取数说明； 4. 重要功能用法中增加“[自动出价调整](#_自动出价调整)”功能说明； |
| 2017/11/08 | 邓伟鹏 | 3.1.7 | 修改redis中java写入部分的价格单位为RMB 分/CPM（原文档中是RMB 分/CPM\*100），涉及一下地方：  “[单个活动的投放控制策略](#_单个订单的投放控制策略（hash）)”中的totalbudget、daybudget\_(月日)  “[单个策略的投放控制策略](#_单个策略的投放控制策略（hash）)”中的totalbudget、daybudget\_(月日)  “[ADX定价合同价格信息](#_ADX定价合同价格信息（HASH）)”中的imp、clk（这里是改成rmb 分/点击）  注：“[活动及策略点击展示价格计数](#_订单及策略点击展示价格计数（HASH）)”中的价格单位未做修改，仍然保持RMB 分/CPM\*100 |
| 2017/11/07 | 邓伟鹏 | 3.1.6 | 1.修改“[获取单个创意的详细信息](#_获取单个创意的详细信息（STRING）)”中curl的描述；  2.去掉“[获取单个创意的详细信息](#_获取单个创意的详细信息（STRING）)”中的securecurl字段；  3.在“[投放策略暂停](#_投放策略暂停(STRING))”中增加一种停止投放原因 |
| 2017/10/26 | 邓伟鹏 | 3.1.5 | 1. 增加对“[获取单个创意的详细信息](#_获取单个创意的详细信息（STRING）)”中对ext的说明； 2. “[设置/获取竞价/结算/展现/点击标志位](#_设置/获取竞价/结算/展现/点击标志位（STRING）)”的可以中增加advid； |
| 2017/10/18 | 邓伟鹏 | 3.1.4 | 1.增加竞价程序对deeplink的支持：   1. [Bid Backed→Common Request](#_Common_Request)中增加“support\_deep\_link” 2. [Bid Backed API→Common Response→Seat bid→Bid](#_Bid)中增加deeplink； 3. Redis-投放设置（单机）中“[获取单个创意的详细信息](#_获取单个创意的详细信息（STRING）)”增加deeplink属性。 4. 增加“[deeplink支持](#_deeplink支持)”说明。   2.去除“[单个投放策略的基本信息](#_单个投放策略的基本信息(STRING))”中的“is\_secure”字段  3.修改“[人群包黑白名单及频次计数控制](#_黑白名单及频次控制计数（hash）)”中的enddatetime为linux时间戳。  4.增加[开发备忘录](#_开发备忘)。  5. “[获取单个创意的详细信息](#_获取单个创意的详细信息（STRING）)”的例子中的“monitorcode”值修正 |
| 2017/10/13 | 邓伟鹏 | 3.1.3 | 1. “[人群包黑白名单及频次控制计数](#_人群包黑白名单及频次控制计数（HASH）)”中key的设备id类型增加“other”和“unknow”两种方式 2. “[人群包黑白名单及频次控制计数](#_人群包黑白名单及频次控制计数（HASH）)”的说明中举例的id改为int型的值 |
| 2017/10/12 | 邓伟鹏 | 3.1.2 | 1. 修改“[ADX定价合同价格信息](#_ADX定价合同价格信息（HASH）)”的存储方式； 2. 删除“[获取单个创意的详细信息（STRING）](#_获取单个创意的详细信息（STRING）)”中的” crid”，其值由“creativeid”替代 |
| 2017/10/12 | 邓伟鹏 | 3.1.1 | 去掉用户点击广告行业编号和广告分类相似度的redis存储。在未来开发自动投放时再统一考虑。 |
| 2017/9/28 | 邓伟鹏 | 3.1.0 | 在“[获取单个创意的详细信息（STRING）](#_获取单个创意的详细信息（STRING）)”中   1. 修改" creativeid”类型为int； 2. 为“cid ”(string,Campaign ID)增加说明； 3. 增加“[Redis-投放控制（单机）设计](#_Redis-投放控制（单机）设计)”； 4. 修改“[重要功能用法](#_重要功能用法)”中的“[频次控制](#_频次控制)”、“[投放KPI及成本控制](#_投放KPI及成本控制)”、“[匀速投放](#_匀速投放)”：在此三处加入跟踪及投放模块对“[Redis-投放控制（单机）设计](#_Redis-投放控制（单机）设计)”的使用; 5. 增加竞价刹车机制，包括：“[广告信息设置更新时间标识](#_广告信息设置更新时间标识（string）)”，“[投放控制更新时间标识](#_投放控制更新时间标识（string）)”和重要功能用法中的“[竞价刹车机制](#_竞价刹车机制)”说明 |
| 2017/9/25 | 邓伟鹏 | 3.0.5 | 1.增加“[ADX定价合同价格信息](#_ADX定价合同价格信息（HASH）)”；  2.重要功能用法的“[投放KPI及成本控制](#_投放KPI及成本控制)”中增加了跟踪模块对定价和竞价价格计算的说明；  3.删除“[单个投放策略的定向信息](#_单个投放策略的定向信息(STRING))”中的“广告位展现形式定向”；  4.将campaignid、policyid、mapid等主要对象id的类型修正为了int型； |
| 2017/9/21 | 邓伟鹏 | 3.0.4 | 在“[获取单个创意的详细信息（STRING）](#_获取单个创意的详细信息（STRING）)”中增加creativeid属性 |
| 2017/9/5 | 邓伟鹏 | 3.0.3 | 修改redis单机中[”允许无设备ID的流量”](#_允许无设备ID的流量（STRING）)表的存储格式为string |
| 2017/8/24 | 邓伟鹏 | 3.0.2 | 增加redis单机中”允许无设备ID的流量”表 |
| 2017/8/15 | 龚永平 | 3.0.1 | 更新” 展现/点击/结算” |
| 2017/6/21 | 邓伟鹏  陈晓明 | 3.0.0 | 协议大改版：  修改了redis结构，区分了单机和集群；  添加了“[重要功能用法](#_重要功能用法)” |
| 2016/12/20 | 周菲 | 2.0.12 | 4.12.1节修改gdt，sohu私有属性  3.1，3.2节补充程序中定义字段 |
| 2016/12/19 | 周菲 | 2.0.11 | 4.3节增加pmp的dealid |
| 2016/12/16 | 张政 | 2.0.10 | 4.3节增加is\_secure |
| 2016/12/8 | 周菲 | 2.0.9 | 4.12节增加securecurl |
| 2016/10/27 | 周菲 | 2.0.8 | 4.12.2节增加zplay私有数据 |
| 2016/10/11 | 薛蕊 | 2.0.7 | 4.3节增加effectmonitor，标识是否需要效果监测 |
| 2016/9/27 | 薛蕊 | 2.0.6 | 4.12.2节增加广点通私有数据添加广告样式 |
| 2016/9/19 | 周菲 | 2.0.5 | 4.6节修改app定向，分平台定向app id  4.3.1节广告主审核ID  4.2.2节增加广点通私有数据pmp合同号 |
| 2016/9/1 | 薛蕊 | 2.0.4 | 4.3节添加第三方监测编号  4.12节添加landingurl |
| 2016/8/16 | 周菲 | 2.0.3 | 3.1.1 节添加adpos, 广告位置  3.1.4 节添加用户user信息 |
| 2016/8/3 | 薛蕊 | 2.0.2 | 4.12添加imgs，存储原生多张图信息 |
| 2016/7/4 | 薛蕊 | 2.0.1 | 3.1.1.2节添加is\_limit\_size，是否限制尺寸  7.15节调整video格式取值 |
| 2016/6/21 | 薛蕊 | 2.0.0 | 更新接口文档样式，添加修改记录，添3加搜狐ext字段描述 |

**目录**

[Pxene RTB接口协议 1](#_Toc495578828)

[0.设计目的 7](#_Toc495578829)

[1. 整体架构 7](#_Toc495578830)

[2. 竞价模块架构 0](#_Toc495578831)

[2.1. Request Adapter（竞价请求适配器） 0](#_Toc495578832)

[2.2. Response Adapter（竞价响应适配器） 0](#_Toc495578833)

[2.3. Bid Backend（标准竞价后端模块） 0](#_Toc495578834)

[2.4. Redis数据库 0](#_Toc495578835)

[3. Bid Backend API接口设计 1](#_Toc495578836)

[3.1. Common Request 1](#_Toc495578837)

[3.1.1. Impression 1](#_Toc495578838)

[3.1.2. App 4](#_Toc495578839)

[3.1.3. Device 4](#_Toc495578840)

[3.1.4. User 5](#_Toc495578841)

[Geo 5](#_Toc495578842)

[3.2. Common Response 5](#_Toc495578843)

[3.2.1. Seat bid 5](#_Toc495578844)

[4. Redis-投放设置（单机）设计 8](#_Toc495578845)

[4.1. 所有的投放活动ID(SET) 8](#_Toc495578846)

[4.2. 所有的投放策略ID(SET) 8](#_Toc495578847)

[4.3. 单个投放策略下所有的创意ID(SET) 8](#_Toc495578848)

[4.4. 单个投放策略的基本信息(STRING) 9](#_Toc495578849)

[4.4.1. 广告主审核ID（由adx提供） 10](#_Toc495578850)

[4.5. 单个投放策略的定向信息(STRING) 11](#_Toc495578851)

[4.6. 单个投放策略引用的人群包(HASH) 14](#_Toc495578852)

[4.7. 获取单个创意的详细信息（STRING） 14](#_Toc495578853)

[4.7.1. 创意审核ID（由adx提供） 17](#_Toc495578854)

[4.7.2. ADX相关的私有数据 18](#_Toc495578855)

[4.8. 单个创意的出价信息（HASH） 21](#_Toc495578856)

[4.9. 获取ADX的定向信息（保留？调研一下） 22](#_Toc495578857)

[4.10. 单个活动的投放控制策略（HASH） 24](#_Toc495578858)

[4.11. 单个策略的投放控制策略（HASH） 25](#_Toc495578859)

[4.12. 允许无设备ID的流量（STRING） 26](#_Toc495578860)

[4.13. 获取Adx的信息（STRING） 27](#_Toc495578861)

[4.14. 获取图片展现形式(STRING) 31](#_Toc495578862)

[4.15. 日志输出内容动态配置(STRING) 32](#_Toc495578863)

[4.16. ADX定价合同价格信息（HASH） 33](#_Toc495578864)

[4.17. 广告信息设置更新时间标识（string） 33](#_Toc495578865)

[5. Redis-投放控制（单机）设计 34](#_Toc495578866)

[5.1. 推广活动暂停(STRING) 34](#_Toc495578867)

[5.2. 投放策略暂停(STRING) 34](#_Toc495578868)

[5.3. 推广活动频次上限(SET) 34](#_Toc495578869)

[5.4. 投放策略频次上限(SET) 35](#_Toc495578870)

[5.5. 创意频次上限(SET) 35](#_Toc495578871)

[5.6. 投放控制更新时间标识（string） 35](#_Toc495578872)

[6. Redis-投放计数（集群）设计 36](#_Toc495578873)

[6.1. 人群包黑白名单及频次控制计数（HASH） 36](#_Toc495578874)

[5.1.1. 频次计数清除索引(SET) 38](#_Toc495578881)

[6.2. 活动及策略点击展示价格计数（HASH） 39](#_Toc495578882)

[6.3. 设置/获取竞价/结算/展现/点击标志位（STRING） 40](#_Toc495578883)

[7. 重要功能用法 40](#_Toc495578884)

[7.1. 基本广告投放 40](#_Toc495578885)

[7.2. 频次控制 41](#_Toc495578886)

[7.3. 人群包 42](#_Toc495578887)

[7.4. 投放KPI及成本控制 43](#_Toc495578888)

[7.5. 匀速投放 45](#_Toc495578889)

[7.6. 出价控制 46](#_Toc495578890)

[7.7. 竞价刹车机制 47](#_Toc495578891)

[8. 展现/点击/结算 47](#_Toc495578892)

[8.1. 赢价/展现/点击流程图 51](#_Toc495578893)

[8.1.1. 价格解密 53](#_Toc495578894)

[9. 日志模块（待更新） 54](#_Toc495578895)

[9.1. 日志文件内容的一般格式 54](#_Toc495578896)

[9.2. 日志的配置文件 54](#_Toc495578897)

[9.3. 日志文件的迁移 55](#_Toc495578898)

[9.4. 日志模块对外提供接口 55](#_Toc495578899)

[9.4.1. openlog 55](#_Toc495578900)

[9.4.2. writelog 55](#_Toc495578901)

[9.4.3. closelog 56](#_Toc495578902)

[9.4.4. 竞价与追踪日志 56](#_Toc495578903)

[10. 附录 57](#_Toc495578904)

[10.1. Adx编号 57](#_Toc495578905)

[10.2. 广告行业/APP类型 58](#_Toc495578906)

[10.3. Imp type 58](#_Toc495578907)

[10.4. Native 58](#_Toc495578908)

[10.4.1. Layout 58](#_Toc495578909)

[10.4.2. Asset type 59](#_Toc495578910)

[10.4.3. Asset Image type 59](#_Toc495578911)

[10.4.4. Asset Data type 59](#_Toc495578912)

[10.5. Didtype/Dpidtype 60](#_Toc495578913)

[Didtype 60](#_Toc495578914)

[Dpidtype 60](#_Toc495578915)

[10.6. Carrier运营商 60](#_Toc495578916)

[10.7. Connectiontype连接类型 61](#_Toc495578917)

[10.8. Devicetype设备类型 61](#_Toc495578918)

[10.9. Ctype点击类型 61](#_Toc495578919)

[10.10. Type广告创意类型 61](#_Toc495578920)

[10.11. Ftype广告创意文件类型 62](#_Toc495578921)

[10.12. Attr广告属性 62](#_Toc495578922)

[10.13. OS 操作系统类型 63](#_Toc495578923)

[10.14. Video广告位类型 63](#_Toc495578924)

[10.15. Video格式类型 63](#_Toc495578925)

[10.16. 图片格式类型 63](#_Toc495578926)

[10.17. Impression 扩展 64](#_Toc495578927)

[instl 64](#_Toc495578928)

[splash 64](#_Toc495578929)

[acceptadtype 64](#_Toc495578930)

[10.18. Make设备制作商类型 64](#_Toc495578931)

[10.19. adpos 广告位置 64](#_Toc495578932)

[10.20. gender 用户性别 65](#_Toc495578933)

[10.21. 第三方监测编号 65](#_Toc495578934)

[11. 错误码 65](#_Toc495578935)

[11.1. 公共 65](#_Toc495578936)

[11.2. 竞价 65](#_Toc495578937)

[11.3. 结算/展现/点击 66](#_Toc495578938)

# 0.设计目的

主要目的为了降低ADX接口差异带来的复杂度，且降低后续接入的ADX带来的重复开发任务。

# 整体架构

整体架构包含以下几个模块：

1. ADX adpater：ADX适配器部分，根据不同的ADX编写对应的ADX adpater，主要功能是将不同的ADX request转化为common request，同时将Backend返回的common response根据ADX平台转化为对应的ADX response。

2. Backend：竞价部分，将ADX adapter转化的common request进行处理，然后根据地域定向，黑白名单等进行竞价，然后返回common response。

3.Settle：结算部分，ADX通知DSP该次展现已经竞价成功，通知会调到Settle程序。

4.Imp\_Click：展现/点击部分，ADX会通知APP去Image Server中下载竞价成功的图片或视频（广告），在展现或点击后会调到Imp\_Click程序。

单个ADX的RTB的整体架构图如下图1.1，



图1.1 单个ADX的RTB的整体架构图

图中，1.ADX request传给ADX adpater进行解析；2.ADX adpater将ADX request转化common request，然后传给Backend，其中Backend表示竞价部分；3.Backend将竞价的common response传给ADX adpater；4.ADX adapter根据平台不同转化为对应平台的response。

# 竞价模块架构



竞价模块整体分为4部分：ADX、Request Adapter、Response Adapter、Bid Backend和Redis数据库。

## Request Adapter（竞价请求适配器）

将adx\_request数据分离成common\_request和private\_request，前者传递给Bid Backend的竞价接口进行竞价，后者交给Response Adapter做后续处理。

## Response Adapter（竞价响应适配器）

接收Bid Backend处理之后的竞价common\_response，将其中的广告数据填充到从Redis读取ADX专属的广告代码模板（如iurl、cturl、adm等）中，转换成ADX特有的数据格式。再结合private\_request中的额外信息，构造adx\_response返回给ADX。

## Bid Backend（标准竞价后端模块）

接受common\_request请求，根据Redis存储的项目创意信息，经过复杂的只能竞价流程后，将竞价创意信息填充到common\_response返回给Response Adapter。

## Redis数据库

本地redis slave，实时和远程redis master保持同步，为Bid Program提供项目、创意、定向、频次等信息；为Response Adapter提供ADX广告代码模板信息。其中ADX广告代码等URL的格式被剥离到Redis里独立存储，common\_response只携带基本广告元数据信息。

# Bid Backend API接口设计

为了兼容多家ADX，并结合当前的竞价算法，故设计公共参数部分和私有参数部分。公共参数，OpenRTB比较符合这个特点。而且很多ADX的API都是基于OpenRTB的接口协议演化而来，故本文选择最新版OpenRTB-API-Specification-Version-2-3作为基础，并做适当的调整和改动作为标准接口。Adapter Request公共接口仅需包含竞价算法所需要的参数即可，其余ADX携带的参数丢弃。Adapter Response的公共接口应该兼容所有ADX定义的Response的必要参数，故Redis中的创意对象应包含尽可能丰富的信息，以满足ADX的Response所需。

## Common Request

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **Comment** | **required** |
| id | string | Bid request的唯一id，由adx生成 | true |
| imp | Impression object array | Impression对象数组 | true |
| app | App object | App对象，描述App信息 | recommended |
| device | Device object | Device对象描述设备信息 | true |
| user | User object | User 对象，描述用户信息 | recommended |
| cur | string set | 竞价支持的若干个货币种类：”CNY”、”USD” | false |
| bcat | int set | 拒绝的广告行业类型。[参考“广告行业/APP类型”](#_广告行业/APP_类别_1) | false |
| badv | string set | 拒绝的广告主域名。（例如：”ford.com”） | false |
| wadv | string set | 支持的广告主 | false |
| at | int | 流量类型，默认为0（RTB） | false |
| is\_recommend | int | 是否使用了智能推荐 | false |
| Is\_secure | int | 是否要求安全连接 | false |
| support\_deep\_link | int | 是否支持deeplink；0：不支持，1：支持，无值：不支持 | false |

### Impression

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| id | string | 唯一id，由adx生成 | true |
| type | int | Impression类型。描述有效的类型是banner、video和native其中一个。[参考“Imp type”](#_Imptype) | true |
| banner | Banner object | Banner对象，当广告位是Banner时必填。 | false |
| video | Video object | Video对象，当广告位是Video时必填。 | false |
| native | Native\_request object | Native对象，当广告位是Native时必填。 | false |
| requestidlife | int | 竞价id有效期 | false |
| bidfloorcur | string | 底价的货币种类。只支持”CNY” | false |
| bidfloor | double | 底价。单位：元/CPM | true |
| adpos | int | 广告位置。[参考adpos](#_adpos_广告位置) | false |
| fprice | map<int, double> | 广告行业对应的底价 | false |
| dealprice | map<string, double> | PMP请求下每个交易id对应的底价，若是PMP流量，须将底价与交易id写入该字段 | false |
| ext | Impression Ext object | 扩展数据 | false |

#### Banner

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| w | int | 宽。单位：像素。 | true |
| h | int | 高。单位：像素。 | true |
| btype | int set | 拒绝的广告创意类型。[参考“Type广告创意类型”](#_Type广告类型) | false |
| battr | int set | 拒绝的广告属性。[参考“Attr广告属性”](#_Attr广告属性) | false |

#### Video

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| mimes | int set | 支持播放的视频格式。[参考“Video格式类型”](#_视频格式类型) | true |
| display | int | 广告位类型。[参考“Video广告位类型”](#_广告位类型) | false |
| minduration | int | 视频广告最小时长 | false |
| maxduration | int | 视频广告最大时长 | false |
| w | int | 广告位宽度 | true |
| h | int | 广告位高度 | true |
| battr | int set | 拒绝的广告属性。[参考“Attr广告属性”](#_Attr广告属性) | false |
| is\_limit\_size | bool | 是否限制尺寸，默认是限制尺寸 | false |

#### Native\_request

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| layout | int | 原生广告布局样式，固定为3。[参考“Layout”](#_Native_Layout) | true |
| plcmtcnt | int | 广告数量。默认值：1 | false |
| Img\_num | int | 所需图片的数量（由后端计算填充） | true |
| bctype | int set | 拒绝的广告点击类型。[参考“Ctype点击类型”](#_Ctype点击类型) | false |
| assets | Asset\_request object array | 原生广告元素列表，当前固定为5个元素，分别为标题（title），Icon（img），Large image（img），Description（data），Rating（data） | true |

##### Asset\_request

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| id | int | 广告元素ID | true |
| required | int | 广告元素是否必须，1：必须，0：可选 | false |
| type | int | Asset类型。描述有效的类型是Title、Image和Data其中一个。[参考“Asset type”](#_Assettype) | true |
| title | Title\_request object | 文字元素。 | false |
| img | Image\_request object | 图片元素。 | false |
| data | Data\_request object | 其他数据元素。 | false |

##### Title\_request

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| len | int | Title元素最大文字长度 | true |

##### Image\_request

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| type | int | Image元素的类型，仅支持1和3。[参考“Asset Image type”](#_Asset_Image_type) | true |
| w | int | Image元素的宽度，单位为像素。 | true |
| wmin | int | Image元素要求的最小宽度，单位为像素。 | false |
| h | int | Image元素的高度，单位为像素。 | true |
| hmin | int | Image元素要求的最小高度，单位为像素。 | false |
| mimes | int set | Image元素支持的图片类型。[参考”Ftype广告创意文件类型“](#_Ftype广告创意文件类型) | false |

##### **Data**\_request

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| type | int | 数据类型，仅支持2,3和12。 [参考“Asset Data type”](#_Asset_Data_type) | true |
| len | int | 元素最大文字长度 | true |

#### Impression Ext

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| instl | int | 是否全插屏广告。[参考“instl](#_instl)” | false |
| splash | int | 是否开屏广告。[参考“splash](#_splash)” | false |
| acceptadtype | int set | 可接受的广告投放行业。 | false |
| data | string | 扩展字段，保存请求中需要字段 | false |
| advid | int | 扩展字段 | false |
| productCat | int set | 扩展字段 | false |
| restrictedCat | int set | 扩展字段 | false |
| sensitiveCat | int set | 扩展字段 | false |

### App

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| id | string | 唯一id，由adx生成 | true |
| name | string | 应用名字 | false |
| cat | int set | App类型。[参考“广告行业/APP类型”](#_广告行业) | false |
| bundle | string | App软件包名 | false |
| storeurl | string | App市场下载地址 | false |

### Device

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| deviceidtype | int | 实际出价的设备id type，取值为request dpids/dids的中的 一个。（由后端填充） | true |
| deviceid | string | 实际出价的设备id，取值为request dpids/dids的中的 一个。（由后端填充） | true |
| dpids | map<int, string> | Key: 平台设备ID类型。[参考“Dpidtype](#_dpidtype)”  value: 平台设备ID (original/sha1/md5) ，如果为sha1/md5，则由适配器转换成小写。 | true |
| dids | map<int, string> | Key: 硬件设备ID类型。[参考“Didtype](#_didtype)”  value: 硬件设备ID (original/sha1/md5)，如果为sha1/md5，则由适配器转换成小写。 | true |
| ua | string | 用户浏览器User-Agent字符串 | true |
| ip | string | Ip地址(ipv4) | true |
| location | int | 通过IPB获取IP的编码 | true |
| locid | char \* | 通过lat和lon计算地域编码（由后端填充） | true |
| geo | Geo object | 地理位置信息 | true |
| carrier | int | 运营商。[参考“Carrier运营商”](#_Carrier运营商) | false |
| make | int | 制造商编码，[参考“Make设备制作商类型”](#_Make设备制作商类型) | false |
| makestr | string | 制造商，如”Apple” | false |
| model | string | 型号，如”iPhone” | false |
| os | int | 设备操作系统。[参考“OS 操作系统类型”](#_OS_操作系统类型) | true |
| osv | string | 设备操作系统版本号，如”3.1.2” | false |
| connectiontype | int | 网络连接类型。[参考“Connectiontype连接类型”](#_Connectiontype连接类型) | false |
| devicetype | int | 设备类型。[参考“Devicetype设备类型”](#_Devicetype设备类型) | false |

### User

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| id | string | 用户id | false |
| gender | int | 性别。[参考gender](#_gender_用户性别) | false |
| yob | int | 出生年份(默认-1) | false |
| keywords | string | 用户标签 | false |
| geo | Geo object | 用户位置信息 | false |
| searchkey | string | 用户历史搜索关键字 | false |

### Geo

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| lat | double | 纬度（-90~90），负数表示南纬。 | true |
| lon | double | 经度（-180~180），负数表示西经。 | true |

## Common Response

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| id | string | 响应的Bid request的唯一id | true |
| seatbid | Seat bid array | 该seat的0或多组bids。 | false |
| bidid | string | Bid response id，由DSP生成。（使用request id即可） | false |
| cur | string | 出价货币：”CNY” | false |

### Seat bid

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| bid | Bid array | DSP的出价。 | true |

#### Bid

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| mapid | string | 创意ID | true |
| impid | string | 对应的Impression id | true |
| price | double | 出价。单位：元/CPM | true |
| adid | string | 广告ID（后台用mapid表示） | true |
| policyid | string | 创意所属的投放策略ID | true |
| cat | int set | 广告行业类型（同一个项目只有一个公共的cat集合）。[参考“广告行业/APP类型”](#_广告行业/APP_类别_1) | true |
| type | int | 广告创意类型。[参考“Type广告创意类型”](#_Type广告类型) | true |
| ftype | int | 广告创意文件类型。[参考“Ftype广告创意文件类型”](#_Ftype广告创意文件类型) | false |
| ctype | int | 点击目标类型（下载App类型的项目，必须且只能定向到1种OS）。[参考“Ctype点击类型”](#_Ctype点击类型) | false |
| bundle | string | 点击后下载App的包名。仅当ctype为2(下载App)有效 | false |
| apkname | string | 点击后下载App的apk名。仅当ctype为2(下载App)有效 | false |
| adomain | string | 广告主检查主域名或顶级域名（同一个项目具有相同公共的adomain） | false |
| w | int | 广告物料宽度 | true |
| h | int | 广告物料高度 | true |
| curl | string | 广告点击目标url（redirect描述curl是否需要被重定向） | true |
| deeplink | string | 包含 deeplink 的点击跳转地址， 无法打开则使用 curl | false |
| redirect | int | 0: 不重定向创意的curl  1: 重定向创意的curl | true |
| effectmonitor | int | 0: 不需要效果监测  1: 需要效果监测 | true |
| monitorcode | string | 第三方广告监测代码片段（预先拼凑过的多个url） | true |
| cmonitorurl | string vector | 第三方广告点击监测地址 | false |
| imonitorurl | string vector | 第三方广告展示监测地址 | false |
| sourceurl | string | 广告素材URL。用于填充广告代码模板 | true |
| cid | string | Campaign ID | false |
| crid | string | 广告物料ID | false |
| attr | int set | 广告属性。[参考“Attr广告属性”](#_Attr广告属性) | false |
| dealid | string | PMP项目投放的交易id | false |
| native | Native\_response object | 原生（信息流）广告对象创意返回信息。 | false |
| groupinfo\_ext | GROUPINFO\_EXT | 项目（广告主）扩展信息 | false |
| creative\_ext | CREATIVE\_EXT | 创意扩展信息 | false |

#### Native\_response

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| assets | Asset\_response object array | 原生广告元素列表，当前固定为5个元素，分别为标题（title），Icon（img），Large image（img），Description（data），Rating（data） | true |

##### Asset\_response

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| id | int | 广告元素ID。 | true |
| required | int | 广告元素是否必须，1：必须，0：可选。 | false |
| type | int | Asset类型。描述有效的类型是Title、Image和Data其中一个。[参考“Assettype”](#_Assettype) | true |
| title | Title\_response object | 文字元素。 | false |
| img | Image\_response object | 图片元素。 | false |
| data | Data\_response object | 其他数据元素。 | false |

##### Title\_response

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| text | string | Title元素的内容文字。 | true |

##### Image\_response

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| url | string | Image元素的URL地址。 | true |
| w | int | Image元素的宽度，单位为像素。 | false |
| h | int | Image元素的高度，单位为像素。 | false |

##### **Data**\_response

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| label | string | 数据显示的名称。 | false |
| value | string | 数据的内容文字。 | true |

#### GROUPINFO\_EXT

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| advid | string | ADX平台备案的广告主ID | true |

#### CREATIVE\_EXT

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| id | string | ADX平台备案的创意ID | true |
| ext | string | 广告创意与发送请求的ADX相关的私有数据。（string内部为json格式） | true |

# Redis-投放设置（单机）设计

## 所有的投放活动ID(SET)

**说明：**

获取所有的在投活动ID

**常用命令：**

SADD key value (写入命令)

SREM key value （删除命令）

SMEMBERS key （返回所有活动ID）

**存储结构：**

key: dsp\_campaignids

value: （活动id）

value type：int

## 所有的投放策略ID(SET)

**说明：**

获取所有的在投策略ID

**常用命令：**

SADD key value (写入命令)

SREM key value （删除命令）

SMEMBERS key （返回所有策略ID）

**存储结构：**

key: dsp\_policyids

value: （投放策略id）

value type：int

## 单个投放策略下所有的创意ID(SET)

**说明：**

单个投放策略下对应的所有创意ID

**常用命令：**

SADD key value (写入命令)

SREM key value （删除命令）

SMEMBERS key （返回所有创意ID）

**存储结构：**

key: dsp\_policy\_mapids\_(policyid)

value: （mapid）

value type：int

## 单个投放策略的基本信息(STRING)

GET key value

key: dsp\_policy\_info\_(policyid)

value:

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| campaignid | int | 所属活动ID | **true** |
| autobid | int | 0-不自动出价；1-自动出价； | **true** |
| cat | int array | 广告主行业类型。[参考“广告行业/APP类型”](#_广告行业) | true |
| advcat | int | 广告行业分类。[参考“广告行业/APP类型”](#_广告行业) | true |
| adomain | string | 广告主域名 | true |
| adx | int array | 允许投放的Adx编号。[参考“Adx编号”](#_Adx编号) | true |
| auctiontype | list | 竞价类型列表 | true |
| adx | int | Adx编号。[参考“Adx编号”](#_Adx编号) | true |
| at | int | 竞价类型：默认为0  0-竞价方式（Open RTB或PMP）  1-定价方式（PD或PDB） | fasle |
| dealid | string | 定价方式项目的交易id | false |
| redirect | int | 0: 不重定向创意的curl  1: 重定向创意的curl | true |
| effectmonitor | int | 0: 不需要效果监测  1: 需要效果监测，默认为0 | false |
| exts(\*) | list | 广告主在adx注册信息 | false |
| adx | int | Adx编号。[参考“Adx编号”](#_Adx编号) | true |
| advid | string | 广告主审核ID（由adx提供）。 是否必选[参考“广告主审核ID（由adx提供”](#_广告主审核ID（由adx提供）) | false |

value的例子：

{

“campaignid”:12355

"cat": [

512,

768,

1024

],

"adomain": "honda.com",

“is\_secure”:0

"adx": [

1,

4

],

"redirect": 1,

"auctiontype": [

{

"adx": 2,

"at": 0,

“dealid”:”123456”

},

{

"adx": 14,

"at": 1

}

],

"exts": [

{

"adx": 1,

"advid": "aaa"

},

{

"adx": 4,

"advid": "ddd"

}

]

}

### 广告主审核ID（由adx提供）

|  |  |  |
| --- | --- | --- |
| **Adx** | **渠道编号** | **required** |
| tanx |  | true |
| amax |  | false |
| inmobi |  | false |
| iwifi |  | false |
| letv |  | false |
| toutiao |  | false |
| zplay |  | false |
| baidu |  | true |
| sohu |  | true |
| adwalker |  | false |
| iqiyi |  | true |
| goyoo |  | false |
| iflytek |  | false |
| adview |  | true |
| google |  | false |
| gdt |  | true |
| lefee |  | false |

## 单个投放策略的定向信息(STRING)

GET key value

key: dsp\_policyid\_target\_(policyid)

value:

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| campaignid | int | 所属的活动id | true |
| policyid | int | 策略ID | true |
| device | obj | 设备信息定向 | false |
| flag | int | 二进制数。每个定向分配两位二进制数（不支持黑白名单同时存在的定向条件，即不支持二进制11出现）：  二进制取值：  00（全部投放），  01（设置投放白名单），  10（设置投放黑名单），  11（全部不投放）。  1~10位每2位表示一个类别：  1-2: regioncode,  3-4: connectiontype,  5-6: os,  7-8: carrier,  9-10: devicetype  11-12：make | true |
| regioncode | int array | 地点定向编码列表，编码值为6位行政区划码 | false |
| connectiontype | int array | 网络定向编码列表。[参考“Connectiontype连接类型”](#_Connectiontype连接类型) | false |
| os | int array | 系统定向编码列表。[参考“OS 操作系统类型”](#_OS_操作系统类型) | false |
| carrier | int array | 运营商定向编码列表。[参考“Carrier运营商”](#_Carrier运营商) | false |
| devicetype | int array | 设备定向编码列表。[参考“Devicetype设备类型”](#_Devicetype设备类型) | false |
| make | Int array | 设备制作商编码列表。[参考“Make 设备制作商类型”](#_Make设备制作商类型) | false |
| app | obj | 应用程序信息定向 | false |
| id(\*) | list | App id 定向列表 | false |
| adx | int | Adx编号。[参考“Adx编号”](#_Adx编号) | true |
| flag |  | 二进制数。每个定向分配两位二进制数（当cat/id按白黑名单同时定向时，则id/cat必须至少设置白名单和黑名单定向中的一种，即**不允许flag二进制取值为11**）：  二进制取值：  00（全部投放），  01（设置投放白名单），  10（设置投放黑名单），  11（设置投放白名单和黑名单）。  1~2位每2位表示一个类别：  1-2: id, | true |
| wlist | string array | id白名单，不同adx对应不同的id | false |
| blist | string array | id黑名单，不同adx对应不同的id | false |
| cat | obj | 兴趣定向编码列表。[参考“广告行业/APP类型”](#_广告行业) | false |
| flag |  | 二进制数。每个定向分配两位二进制数（当cat/id按白黑名单同时定向时，则id/cat必须至少设置白名单和黑名单定向中的一种，即**不允许flag二进制取值为11**）：  二进制取值：  00（全部投放），  01（设置投放白名单），  10（设置投放黑名单），  11（设置投放白名单和黑名单）。  1~2位每2位表示一个类别：  1-2: cat | true |
| wlist | int array | cat白名单 | false |
| blist | int array | cat黑名单 | false |
| nodeviceid(\*) | list | 允许无设备Id列表 | false |
| adxcode | int | 渠道（adx）编号 | false |
| media(\*) | list | 该渠道下的媒体列表 |  |
| mediaid | string | 媒体id（appid） | false |
| imptype | int array | imptype是Impression类型。描述有效的类型是banner、video和native其中一个，取值1~3，参考附录中的“[Imptype](#_Imptype)” | false |
| scene | obj | 场景定向 | false |
| flag | int | 二进制数。每个定向分配两位二进制数（不支持黑白名单同时存在的定向条件，即不支持二进制11出现）：  二进制取值：  00（全部投放），  01（设置投放白名单），  10（设置投放黑名单），  11（全部不投放）。  1~2位每2位表示一个类别：  1-2:location | false |
| length | int | GeoHash length，不同编码长度对应不同精度(参考https://segmentfault.com/a/1190000002513514) | false |
| loc（\*） | list | 位置信息列表 | false |
| lat | double | 纬度从- 9 0到+9 0，负的是南 | false |
| lon | double | 经度从-1 8 0到+ 1 8 0，负的是西 | false |
| id | string | GeoHash编码（参考https://code.google.com/archive/p/python-geohash/source），目前和前端约定编码长度为12位 | false |

value的例子：

{

"campaignid": 123455,

"policyid": 668812,

"device": {

"flag": 341,

"regioncode": [

110000,

610000

],

"connectiontype": [

4,

5

],

"os": [

1,

2

],

"carrier": [

1,

2

],

"devicetype": [

1,

2

],

"make": [

1,

2,

3

]

},

"app": {

"id": [

{

"adx": 1,

"flag":1 ,

"wlist": [

"tanx-appid1",

"tanx-appid2"

]

},

{

"adx": 2,

"flag":1 ,

"wlist": [

"amax-appid3",

"amax-appid4"

]

}

],

"cat": {

"flag":2 ,

"wlist": [

1,

2

]

}

},

"nodeviceid":

[

{

"adxcode":12,

"media":

[

{

"mediaid":"123as3",

"imptype":[1,2]

},

{

"mediaid":"123a2343",

"imptype":[1,2,3]

}

],

},

{

"adxcode":3,

"media":

[

{

"mediaid":"123313",

"imptype":[1]

},

{

"mediaid":"127678fj7",

"imptype":[1,2,3]

}

],

}

],

" scene": {

"flag": 1,

“length”:6

"loc": [

{

"lat": 38.0468,

"lon": 115.9796,

"id": "wwf2rnjy8t9f"

},

{

"lat": 34.2607,

"lon": 109.0162,

"id": "wqjdbszpj96k"

}

]

}

}

## 单个投放策略引用的人群包(HASH)

**说明：**

单个人群包对应（被引用的）投放策略ID列表

**常用命令：**

HSET key field value

HMSET key field value [field value …]

HGET key field

HMGET key field [field…]

HGETALL key

**存储结构：**

key: dsp \_policy\_audienceid\_(policyid)

field: (audienceid)

value：[0/1]（0-黑名单,1-白名单）

## 获取单个创意的详细信息（STRING）

GET key value

key: dsp\_mapid\_(mapid)

value:

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| mapid | int | 单个策略下的创意ID | true |
| policyid | int | 创意所属的投放策略ID | true |
| creativeid | int | 奥丁平台中该创意的ID，该id要通过跟踪链接传递给跟踪系统，以便跟踪系统做单个创意的频次计数 | true |
| adid | int | 广告ID（后台用mapid表示） | false |
| type | int | 广告创意类型。[参考“Type广告创意类型”](#_Type广告类型) | true |
| ftype | int | 广告创意文件类型。[参考“Ftype广告创意文件类型”](#_Ftype广告创意文件类型) | true |
| ctype | int | 点击目标类型（下载App类型的项目，必须且只能定向到1种OS）。[参考“Ctype点击类型”](#_Ctype点击类型) | false |
| bundle | string | 点击后下载App的包名。仅当ctype为2(下载App) 时必选。 | false |
| apkname | string | 点击后下载App的apk名。仅当ctype为2(下载App) 时必选。 | false |
| w | int | 广告宽度。创意类型为图片或视频时，为该图片或视频的宽度。创意类型为原生时取第一个大图的宽，如果没有大图则取第一个icon的宽。 | true |
| h | int | 广告高度。创意类型为图片或视频时，为该图片或视频的高度。创意类型为原生时取第一个大图的高，如果没有大图则取第一个icon的高。 | true |
| curl | string | 广告点击时跳转url（非deeplink或者deeplink用户手机无app的情况）；  竞价程序会将此字段填写在response中的curl中；  如果物料包中有点击监测地址，则此处填写点击监测地址，如果物料包中没有点击监测地址则此处填写落地页地址；  ps：物料包中的监测地址均为第三发监控地址； | false |
| landingurl | string | 点击广告的着陆页（当curl不通过第三方跳转时，curl和landurl是一致的） | true |
| deeplink | string | deeplink可以唤起另一个app并传入参数直接进入内页，此功能需要渠道和媒体支持。 | false |
| monitorcode | string | 第三方广告监测代码片段（预先拼凑过的多个url） | true |
| cmonitorurl | string array | 第三方广告点击监测地址 | true |
| imonitorurl | string array | 第三方广告展示监测地址 | true |
| sourceurl | string | 素材地址 | true |
| cid | string | Campaign ID，投放活动ID | false |
| attr | int array | 广告属性。[参考“Attr广告属性”](#_Attr广告属性) | false |
| badx | Int array | 拒绝投放的Adx编号。[参考“Adx编号”](#_Adx编号) | false |
| imgs | object array | 原生多图信息，大图两张或以上，数组中的顺序为上传时的顺序。 | false |
| ftype | int | 广告创意文件类型。[参考“Ftype广告创意文件类型”](#_Ftype广告创意文件类型) | true |
| w | int | 广告宽度 | true |
| h | int | 广告高度 | true |
| sourceurl | string | 图片素材地址 | true |
| icon | object | 广告图标。仅当type为9（信息流）时必选。 | false |
| ftype | int | 广告图标文件类型。[参考“Ftype广告创意文件类型”](#_Ftype广告创意文件类型) | true |
| w | int | 广告图标宽度。 | true |
| h | int | 广告图标高度。 | true |
| sourceurl | string | 图标素材地址。 | true |
| title | string | 广告标题。仅当type为9（信息流）时必选。 | false |
| description | string | 广告描述。仅当type为9（信息流）且不包含App级别时必选。 | false |
| rating | int | 产品级别。仅当type为9（信息流）且包含App级别时必选。如App Store级别从0~5。默认：255 | false |
| ctatext | string | CTA描述，类似（Install, Read More）。仅当type为9（信息流）时必选。 | false |
| exts(\*) | list | 广告创意扩展数据。 | false |
| adx | int | Adx编号。[参考“Adx编号”](#_Adx编号) | true |
| id | string | 创意审核ID（由adx提供）。是否必选[参考“创意审核ID（由adx提供）”](#_创意审核ID（由adx提供）) | false |
| ext | Object | ADX相关的私有数据。是否必选[参考“ADX相关的私有数据”](#_ADX相关的私有数据)（string内部为json格式） | false |

Value：Banner的例子：

{

“mapid”:”mapid1”,

“policyid”:” policyid1”,

“creativeid”:” creativeid1”,

“adid”:”dfdfdfdfdfdfdfdf”,

“type”:2,+

“ftype”:17,

“ctype”:3,

“bundle”:” mobileqq\_android.apk”,

“apkname“: “ mobileqq “,

“w”:320,

“h”:50,

“curl”:” <http://www.pxene.com>”,

“monitorcode”:” var imgdc1 = new Image();imgdc1.src = 'http://impression.gridsumdissector.com/gs.gif?gscmd=impress&gsadid=gad\_140\_CDK04487';var imgdc2 = new Image();imgdc2.src = 'http://impression.gridsumdissector.com/gs.gif?gscmd=impress&gsadid=gad\_140\_CDK044873';”,

“cmonitorurl”:[” http://42.96.192.35:59011/c.gif”, ” http://42.96.192.36:59011/c.gif”]

“imonitorurl”:[” http://42.96.192.35:59011/m.gif”, ” http://42.96.192.36:59011/m.gif”]

“sourceurl”:”dfdgdgxv”,

“cid”:”wwwwwwwwwww”,

“attr”:[2,3],

"exts": [

{

"adx": 14,

"id": "15126",

"ext": {

}

}

] //具体结构按依ADX不同

}

Value：Native的例子：

{

“mapid”:”mapid1”,

“policyid”:” policyid1”,

“creativeid”:” creativeid1”,

“adid”:”dfdfdfdfdfdfdfdf”,

“type”:9,

“ftype”:17,

“ctype”:3,

“bundle”:” mobileqq\_android.apk”,

“apkname“: “ mobileqq “,

“w”:1200,

“h”:800,

“curl”:” <http://www.pxene.com>”,

“monitorcode”:” var imgdc1 = new Image();imgdc1.src = 'http://impression.gridsumdissector.com/gs.gif?gscmd=impress&gsadid=gad\_140\_CDK04487';var imgdc2 = new Image();imgdc2.src = 'http://impression.gridsumdissector.com/gs.gif?gscmd=impress&gsadid=gad\_140\_CDK044873';”,

“cmonitorurl”:[”http://42.96.192.35:59011/c.gif”, ”http://42.96.192.36:59011/c.gif”]

“imonitorurl”:[”http://42.96.192.35:59011/m.gif”, ”http://42.96.192.36:59011/m.gif”]

“sourceurl”:”dfdgdgxv.png”,

“cid”:”wwwwwwwwwww”,

“attr”:[2,3],

“imgs”:

[

{“w”:250,”h”:250,”ftype”:18,”sourceurl”:” <http://test.pxene.com:59001/dav/65d106ff-2cb9-4ae7-a2d8-897fe3f05f64/image/ef9056b87c474463909130d0c7bb6b02.jpg>”},

{“w”:250,”h”:250,”ftype”:17,”sourceurl”:” http://test.pxene.com:59001/dav/65d106ff-2cb9-4ae7-a2d8-897fe3f05f64/image/ef9056b87c474463909130d0c7bb6b03.png”}

]

“icon”:{

“ftype”:17,

“w”:100,

“h”:100,

“sourceurl”: ”frewgghtrh.png”

},

“title”:”这是一个广告标题”,

“description”:”这是一个广告描述”,

“rating”:5,

"exts": [

{

"adx": 14,

"id": "15126",

"ext": {

}

}

] //具体结构按依ADX不同

}

### 创意审核ID（由adx提供）

|  |  |
| --- | --- |
| **Adx** | **required** |
| tanx | false |
| amax | false |
| inmobi | false |
| iwifi | true |
| letv | true |
| toutiao | false |
| zplay | false |
| baidu | true |
| sohu | true |
| adwalker | false |
| iqiyi | true |
| goyoo | false |
| iflytek | false |
| adview | true |
| google | true |
| gdt | true |
| lefee | false |

### ADX相关的私有数据

#### Tanx

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
|  |  |  |  |

#### Amax

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| adt | int | 仅在开屏广告创意时，标识创意开屏广告的播放时长 | true |
| ade | string | 仅在开屏广告创意时，标识开屏广告的过期时间，格 式 :YYYYMMdd , 例 如 20140730 | true |

#### Inmobi

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
|  |  |  |  |

#### Iwifi

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
|  |  |  |  |

#### Letv

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
|  |  |  |  |

#### Toutiao

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
|  |  |  |  |

#### Zplay

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| sourceurl | string | 图片地址，视频的创意需要。 | true |
| duration |  | 创意时长，视频的创意需要。格 式 :hh:mm:ss , 例 如 00:00:15 | true |

{

"exts": [

{

"adx": 7,

"ext": {

“sourceurl”:<http://myservers.f3322.org:59001/dav/db2a06920-50-501.jpg>”,

“duration”:”00:00:15”

}

}

]

}

#### Baidu

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
|  |  |  |  |

#### Sohu

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| adtype | int array | 搜狐（adx为9）私有数据，用于标识可接受的广告投放类型。 | true |
| adxtype | int | 投放平台（1，搜狐新闻 2，搜狐视频） |  |

ext例子：

{

"exts": [

{

"adx": 9,

"id": " http://myservers.f3322.org:59001/dav/db2a069d-c15e-4af8-8932-4934db6b4ad1/320-50-501.jpg ",

"ext": {

"adtype":[

1,

3

],

“adx”:1

}

}

]

}

#### Adwalker

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
|  |  |  |  |

#### Iqiyi

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| duration | string | 创意时长，视频的创意需要。格 式 :hh:mm:ss , 例 如 00:00:15 | true |

#### Goyoo

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
|  |  |  |  |

#### iflytek

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
|  |  |  |  |

#### adview

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
|  |  |  |  |
|  |  |  |  |

ext例子：

"exts": [

{

"adx": 14,

"id": "15126",

"ext": {

}

}

]

#### google

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| advid | int array | Detected advertiser ID | false |
| procat | int array | Detected product categories | false |
| restrictcat | int array | All restricted categories for the ads that may be shown from this snippet. | false |
| sendtivecat | int array | Detected sensitive categories | false |

#### gdt

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| creative\_spec | int | 广告样式，见《广告样式说明》文档 | true |

{

"exts": [

{

"adx": 21,

"id": " 123",

"ext": {

" creative\_spec ":114

}

}

]

}

## 单个创意的出价信息（HASH）

**常用命令：**

HSET key field value

HMSET key field value [field value …]

HGET key field

HMGET key field [field…]

HGETALL key

**存储结构：**

key: dsp\_mapid\_bid\_(mapid)

fieldvalue:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **valuetype** | **comment** | **required** |
| (adxcode)\_(mediacode)\_( regioncode) | int | 精确到媒体和地区的出价。value<=0时为不出价，即不投。 | false |
| (adxcode)\_(mediacode)\_all | int | 指定媒体但不指定地区的出价。value<=0时为不出价，即不投。 | false |
| (adxcode)\_all\_( regioncode) | int | 指定adx和地区，但不指定媒体的出价。value<=0时为不出价，即不投。 | false |
| (adxcode)\_all\_all | int | 指定adx但不指定媒体和地区的出价。value<=0时为不出价，即不投。 | false |
| all\_all\_( regioncode) | int | 只指定地区的出价。value<=0时为不出价，即不投。 | false |
| all\_all\_all | int | 没有指定任何条件的，本mapid的出价。value<=0时为不出价，即不投。 | true |

**格式说明：**

adxcode：渠道（adx）编码

mediacode：媒体（app）编码，App时使用APPID

regioncode：地域（省、市）编码。编码后四位为城市（区/县），如果没有匹配的地区编码，可以将后四位替换为0后再次尝试匹配。

**使用说明：**

出价设定：当价格（value）<=0时为不出价，即不投。价格单位 人民币 分/cpm；

确定出价：以最精细（指定条件最详细）的方式优先寻找价格，如果没有则以次级精细的方式尝试寻找，直到找到符合当前流量的出价，具体顺序如下：

1. (adxcode)\_(mediacode)\_( regioncode)
2. (adxcode)\_(mediacode)\_all
3. (adxcode)\_all\_( regioncode)
4. (adxcode)\_all\_all
5. all\_all\_( regioncode)
6. all\_all\_all

**PS:**地区条件（regioncode）如果后四位不为“0000”且匹配不到时，需要将后四位替换为“0000”后再次进行匹配。

## 获取ADX的定向信息（保留？调研一下）

GET key value

key: dsp\_adx\_target\_(adx) (adx表示投放平台，长度固定为2位)

value:

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| adx | int | Adx编号。[参考“Adx编号”](#_Adx编号) | true |
| device | object | [参考“获取单个项目的定向信息”](#_获取单个项目的定向信息) | false |
| app | object | [参考“获取单个项目的定向信息”](#_获取单个项目的定向信息)，其中id(\*)应当只有本adx的数据 | false |
| imp | object | [参考“获取单个策略的定向信息”](#_获取单个项目的定向信息) | false |

比如：获取芒果平台定向信息

get dsp\_adx\_target\_02

返回值：

{

"adx": 12,

"device": {

"flag": 341,

"regioncode": [

110000,

610000

],

"connectiontype": [

4,

5

],

"os": [

1,

2

],

"carrier": [

1,

2

],

"devicetype": [

1,

2

],

"make": [

1,

2,

3

]

},

"app": {

"id": [

{

"flag": 1,

"adx": 2,

"wlist": [

"amax-appid3",

"amax-appid4"

],

"blist": [

"amax-appid103",

"amax-appid104"

]

}

],

"cat": {

"flag": 1,

"wlist": [

1,

2

],

"blist": [

3,

4

]

}

},

"scenario": {

"flag": 1,

"loc": [

{

"lat": 30.5235,

"lon": 114.5161

},

{

"lat": 26.063,

"lon": 119.29

}

]

}

}

## 单个活动的投放控制策略（HASH）

**常用命令：**

HSET key field value

HMSET key field value [field value …]

HGET key field

HMGET key field [field…]

HGETALL key

**存储结构：**

key: dsp\_campaign\_control\_(campaignid)

fieldvalue:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **valuetype** | **comment** | **required** |
| totalimp | int | 总展示数，-1或没有表示不限 | true |
| totalclk | int | 总点击数，-1或没有表示不限 | true |
| totalbudget | int | 总预算。-1表示不限，单位rmb 分/cpm | true |
| dayimp\_(月日) | int | 日展示量目标，(月日)为变量，java更新时只更新当日和次日的策略，每天凌晨2点清除前一日的field，value=-1表示不限，没有此日期的filed则表示那天不投。  例：  当日为2017年7月8日，则存在以下2个field  dayimp\_0708  dayimp\_0709 | true |
| dayclk\_(月日) | int | 日点击量目标，(月日)为变量，java更新时只更新当日和次日的策略，每天凌晨2点清除前一日的field，value=-1表示不限，没有此日期的filed则表示那天不投。  例：  当日为2017年7月8日，则存在以下2个field  dayclk\_0708  dayclk\_0709 | true |
| daybudget\_(月日) | int | 日预算，(月日)为变量，java更新时只更新当日和次日的策略，每天凌晨2点清除前一日的field，value=-1表示不限，没有此日期的filed则表示那天不投。单位rmb 分/cpm 。  例：  当日为2017年7月8日，则存在以下2个field  daybudget\_0708  daybudget\_0709 | true |
| frequencytype | int | 频次控制类型：  0：不限制  1：按活动下所有创意频次总和控制  2：按每个创意的频次控制 | true |
| frequencyaction | int | 频次控制行为；Type为0时，该字段为空：  1：展现  2：点击 | false |
| frequencyperiod | int | 频次控制周期；Type为0时，该字段为空：  取值（取值范围：0，1，2，3，4）:  0：全周期  1：1小时  2：1天  3：1周（按自然周）  4：1月（按自然月） | false |
| frequencyendtime | int | 当frequencyperiod为全周期时，频次控制结束日期。该日期为投放本活动的结束日期的23:59:59。格式为时间戳 | false |
| frequencycount | int | 频次控制数量; Type为0时，该字段为空： | flase |

## 单个策略的投放控制策略（HASH）

**常用命令：**

HSET key field value

HMSET key field value [field value …]

HGET key field

HMGET key field [field…]

HGETALL key

**存储结构：**

key: dsp\_policy\_control\_(policyid)

fieldvalue:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **valuetype** | **comment** | **required** |
| uniform | int | 是否匀速投放：0不匀速投放，1匀速投放 | true |
| totalimp | int | 总展示数，-1或没有表示不限 | true |
| totalclk | int | 总点击数，-1或没有表示不限 | true |
| totalbudget | int | 总预算。-1表示不限，单位rmb 分/cpm。 | true |
| allowtime\_(月日) | int | 允许的时段，(月日)为变量，java更新时只更新当日和次日的策略，每天凌晨2点清除前一日的field。  value是24位二进制数，从每天0点开始为右侧第一位，23点为左侧第24位，允许投放的时段对应的位取值为1，否则为0。例如：当天投放时段为凌晨2点~5点，早上8点~10点，则取值为：11100111100  判断当前时段是否允许投放：取当前时间整点数为n，以2的n次方和当前value进行“与”运算，结果为0则当前时段不允许投放； | true |
| dayimp\_(月日) | int | 日展示上限，(月日)为变量，java更新时只更新当日和次日的策略，每天凌晨2点清除前一日的field，value=-1表示不限，没有此日期的filed则表示那天不投。  例：  当日为2017年7月8日，则存在以下2个field  dayimp\_0708  dayimp\_0709 | true |
| dayclk\_(月日) | int | 日点击上限，(月日)为变量，java更新时只更新当日和次日的策略，每天凌晨2点清除前一日的field，value=-1表示不限，没有此日期的filed则表示那天不投。  例：  当日为2017年7月8日，则存在以下2个field  dayclk\_0708  dayclk\_0709 | true |
| daybudget\_(月日) | int | 日预算，单位rmb 分/cpm。 (月日)为变量，java更新时只更新当日和次日的策略，每天凌晨2点清除前一日的field，value=-1表示不限，没有此日期的filed则表示那天不投。  例：  当日为2017年7月8日，则存在以下2个field  daybudget\_0708  daybudget\_0709 | true |
| frequencytype | int | 频次控制类型：  0：不限制  1：按投放策略下所有创意频次总和控制 | true |
| frequencyaction | int | 频次控制行为；Type为0时，该字段为空：  1：展现  2：点击 | false |
| frequencyperiod | int | 频次控制周期；Type为0时，该字段为空：  取值（取值范围：0，1，2，3，4）:  0：全周期  1：1小时  2：1天  3：1周（按自然周）  4：1月（按自然月） | false |
| frequencyendtime | int | 当frequencyperiod为全周期时，频次控制结束日期。该日期为投放本投放策略结束日期的23:59:59。格式为时间戳 | false |
| frequencycount | int | 频次控制数量; Type为0时，该字段为空： | flase |

## 允许无设备ID的流量（STRING）

**说明：**

允许没有设备ID流量的渠道、媒体、广告形式。分为全平台和投放策略两层过滤，两层都允许的情况下无设备ID的广告请求才会被出价。由java后台逻辑控制全平台允许且有投放策略允许才会写入redis本数据；

**常用命令：**

SET key value [EX seconds] [PX milliseconds] [NX|XX] (写入命令)

GET key （返回所有活动ID）

**存储结构：**

key: dsp\_nodeviceid \_adx\_(adxcode)\_imptype\_(imptypeid)

说明：imptype是Impression类型。描述有效的类型是banner、video和native其中一个。参考“Imptype”；

value:

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| medialist | obj list |  | true |
| mediaid | string | 媒体的id，原appid | true |
| policyidlist | int list | 允许的投放策略id列表 | true |
|  |  |  |  |

value的例子：

{

“medialist”:

[

{

“mediaid”:”appid1”,

“policyidlist”:[1112,324234,4511,4321]

},

{

“mediaid”:”appid2”,

“policyidlist”:[1112,324234,4511,4321]

},

{

“mediaid”:”appid3”,

“policyidlist”:[1112,324234,4511,4321]

}

]

}

## 获取Adx的信息（STRING）

SET/GET key value

key: adx\_templates

value:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **name** | | **type** | **comment** | **required** |
| templates(\*) | | list | Adx模板列表 | true |
|  | adx | int | Adx编号。[参考“Adx编号”](#_Adx编号) | true |
|  | ratio | double | 价格换算比例，由adx价格定义和货币汇率影响。（创意price乘以此字段作为bid price返回给adx） | true |
|  | iurl | string array | 广告展示监测地址。 | true |
|  | cturl | string array | 广告点击监测地址。 | true |
|  | aurl | string | 激活效果地址。 | true |
|  | nurl | string | Win Notice URL。 | true |
|  | adms(\*) | list | 全部os和type决定的广告代码片段模板。 | true |
|  | os | int | 系统类型。[参考“OS 操作系统类型”](#_OS_操作系统类型) | true |
|  | type | int | 广告创意类型。[参考“Type广告创意类型”](#_Type广告类型) | true |
|  | adm | string | 广告代码片段模板。 | true |

value的例子：

{

“templates”:

[

{

"adx": 1,

"ratio": 1,

"cturl": [

"http://clk.pxene.com/m?adx=1&mtype=c&mapid=#MAPID#&bid=#BID#&deviceid=#DEVICEID#&deviceidtype=#DEVICEIDTYPE#"

],

"aurl": "http://active.pxene.com/m?adx=1&mtype=s&mapid=#MAPID#&bid=#BID#&deviceid=#DEVICEID#&deviceidtype=#DEVICEIDTYPE#",

"nurl": "http://settle.pxene.com/p?adx=1&price=%%SETTLE\_PRICE%%&mapid=#MAPID#&bid=#BID#&deviceid=#DEVICEID#&deviceidtype=#DEVICEIDTYPE#",

"iurl": [

"http://imp.pxene.com/m?adx=1&mtype=m&mapid=#MAPID#&bid=#BID#&deviceid=#DEVICEID#&deviceidtype=#DEVICEIDTYPE#"

],

"adms": [

{

"os": 1,

"type": 2,

"adm": "<html><head><meta http-equiv='Content-Type' content='text/html; charset=utf-8'/><meta name='viewport' content='user-scalable=no, width=device-width,initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0'/><script type='text/javascript' src='http://cdn.tanx.com/t/tanxmobile/mraid.js'><\\/script><style type='text/css'>html, body {margin: 0;padding: 0;}<\\/style><\\/head><body><type='text/javascript'>if ((mraid.getState() == 'loading')) {mraid.addEventListener('ready', loadAndDisplayAd);} else {loadAndDisplayAd();}function loadAndDisplayAd() {var img = new Image();img.onload = function() {document.body.appendChild(img);mraid.show();document.addEventListener('touchstart',function() {mraid.open('%%CLICK\_URL%%');})};img.src = '#SOURCEURL#';};mraid.addEventListener('actionChange', function(json) {if(json.command == 'open' && json.status == 'open') {var img = new Image();img.src = '#CURL#';window[new Date()] = img;}});mraid.addEventListener('viewableChange', function(viewable) {if (viewable) {var img = new Image();img.src = '#IURL#';window[new Date()] = img;#MONITORCODE#}});<\\/script><img style='width:0;height:0' src='#NURL#'/><\\/body><\\/html>"

},

{

"os": 2,

"type": 2,

"adm": "<html><head><meta http-equiv='Content-Type' content='text/html; charset=utf-8'/><meta name='viewport' content='user-scalable=no, width=device-width,initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0'/><script type='text/javascript' src='http://cdn.tanx.com/t/tanxmobile/mraid.js'><\\/script><style type='text/css'>html, body {margin: 0;padding: 0;}<\\/style><\\/head><body><type='text/javascript'>if ((mraid.getState() == 'loading')) {mraid.addEventListener('ready', loadAndDisplayAd);} else {loadAndDisplayAd();}function loadAndDisplayAd() {var img = new Image();img.onload = function() {document.body.appendChild(img);mraid.show();document.addEventListener('touchstart',function() {mraid.open('%%CLICK\_URL%%');})};img.src = '#SOURCEURL#';};mraid.addEventListener('actionChange', function(json) {if(json.command == 'open' && json.status == 'open') {var img = new Image();img.src = '#CURL#';window[new Date()] = img;}});mraid.addEventListener('viewableChange', function(viewable) {if (viewable) {var img = new Image();img.src = '#IURL#';window[new Date()] = img;#MONITORCODE#}});<\\/script><img style='width:0;height:0' src='#NURL#'/><\\/body><\\/html>"

}

]

},

{

"adx": 2,

"ratio": 1,

"cturl": [

"http://clk.pxene.com/m?adx=2&mtype=c&mapid=#MAPID#&bid=#BID#&deviceid=#DEVICEID#&deviceidtype=#DEVICEIDTYPE#"

],

"aurl": "http://active.pxene.com/m?adx=2&mtype=s&mapid=#MAPID#&bid=#BID#&deviceid=#DEVICEID#&deviceidtype=#DEVICEIDTYPE#",

"nurl": "http://settle.pxene.com/p?adx=2&price=#WIN\_PRICE#&mapid=#MAPID#&bid=#BID#&deviceid=#DEVICEID#&deviceidtype=#DEVICEIDTYPE#",

"iurl": [

"http://imp.pxene.com/m?adx=2&mtype=m&mapid=#MAPID#&bid=#BID#&deviceid=#DEVICEID#&deviceidtype=#DEVICEIDTYPE#"

],

"adms": [

{

"os": 1,

"type": 2,

"adm": "<meta http-equiv='Content-Type' content='text/html; charset=UTF-8' /><style type='text/css'>\*{padding:0px;margin:0px;} a:link{text-decoration:none;}<\\/style><a href='#CLICK\_URL#'><img width='100%%' height='100%%' src='#SOURCEURL#'><\\/img><img style='width:0;height:0' src='#IURL#'/><\\/a>"

},

{

"os": 2,

"type": 2,

"adm": "<meta http-equiv='Content-Type' content='text/html; charset=UTF-8' /><style type='text/css'>\*{padding:0px;margin:0px;} a:link{text-decoration:none;}<\\/style><a href='#CLICK\_URL#'><img width='100%%' height='100%%' src='#SOURCEURL#'><\\/img><img style='width:0;height:0' src='#IURL#'/><\\/a>"

}

]

},

{

"adx": 3,

"ratio": 0.162,

"cturl": [

"http://clk.pxene.com/m?adx=3&mtype=c&mapid=#MAPID#&bid=#BID#&deviceid=#DEVICEID#&deviceidtype=#DEVICEIDTYPE#"

],

"aurl": "http://active.pxene.com/m?adx=3&mtype=s&mapid=#MAPID#&bid=#BID#&deviceid=#DEVICEID#&deviceidtype=#DEVICEIDTYPE#",

"nurl": "http://settle.pxene.com/p?adx=3&price=${AUCTION\_PRICE}&mapid=#MAPID#&bid=#BID#&deviceid=#DEVICEID#&deviceidtype=#DEVICEIDTYPE#",

"iurl": [

"http://imp.pxene.com/m?adx=3&mtype=m&mapid=#MAPID#&bid=#BID#&deviceid=#DEVICEID#&deviceidtype=#DEVICEIDTYPE#"

],

"adms": [

{

"os": 2,

"type": 2,

"adm": "<meta http-equiv='Content-Type' content='text/html; charset=UTF-8' /><style type='text/css'>\*{padding:0px;margin:0px;} a:link{text-decoration:none;}<\\/style><a href='#CTURL#'><img width='100%%' height='100%%' src='#SOURCEURL#'><\\/img><img style='width:0;height:0' src='#IURL#'/><\\/a>"

},

{

"os": 1,

"type": 2,

"adm": "<meta http-equiv='Content-Type' content='text/html; charset=UTF-8' /><style type='text/css'>\*{padding:0px;margin:0px;} a:link{text-decoration:none;}<\\/style><a href='#CTURL#'><img width='100%%' height='100%%' src='#SOURCEURL#'><\\/img><img style='width:0;height:0' src='#IURL#'/><\\/a>"

}

]

}]

}

## 获取图片展现形式(STRING)

GET key value

key: dsp\_img\_imp\_type

value:

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| img\_imp\_type(\*) | list | 图片展现形式列表 | true |
| adx | int | Adx编号。[参考“Adx编号”](#_Adx编号) | true |
| imps(\*) | list | Adx图片展现形式列表 | true |
| type | int | 图片展现形式（0：横幅，1：全插屏，3：开屏） | true |
| sizes(\*) | list | size数组 | true |
|  | string | size字符串(如：640x100) | true |

value的例子：

{

"img\_imp\_type": [

{

"adx": 2,

"imps": [

{

"type": 0,

"sizes": [

"320x50",

"640x100"

]

},

{

"type": 1,

"sizes": [

"1136x640",

"640x1136"

]

},

{

"type": 3,

"sizes": [

"640x960",

"1080x1740"

]

}

]

},

{

"adx": 3,

"imps": [

{

"type": 0,

"sizes": [

"320x50",

"640x100"

]

},

{

"type": 1,

"sizes": [

"480x320",

"320x480"

]

}

]

}

]

}

## 日志输出内容动态配置(STRING)

GET key value

key: dsp\_log\_conf

value:

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **type** | **comment** | **required** |
| dsp\_log\_conf(\*) | list | 日志输出内容配置列表 | false |
| adx | int | Adx编号。[参考“Adx编号”](#_Adx编号) | false |
| ferr\_level | int | 后端错误信息详细信息级别（0：创意级别，1：项目级别，2：只返回最终错误码），默认为2 | false |
| is\_print\_time | int | 是否打印耗时日志（0：不打印，1：打印），默认为0 | false |

value的例子：

{

"dsp\_log\_conf": [

{

"adx": 2,

"is\_print\_time": 0,

"ferr\_level": 3

},

{

"adx": 3,

"is\_print\_time": 1,

"ferr\_level": 4

}

]

}

## ADX定价合同价格信息（HASH）

**常用命令：**

HSET key field value

HMSET key field value [field value …]

HGET key field

HMGET key field [field…]

HGETALL key

**存储结构：**

key: dsp\_fixprice\_(adxcode)\_ (dealid)

fieldvalue:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **valuetype** | **comment** | **required** |
| imp | int | 合同号和展示/点击选项。单位rmb 分/cpm。如果不按展示计价，则此处为“-1”。 | true |
| clk | int | 合同号和展示/点击选项。单位rmb 分/点击。如果不按点击计价，则此处为“-1”。 | true |

## 广告信息设置更新时间标识（string）

**说明：**

该时间戳用于竞价程序判断redis写入程序是否正常工作。

java每次更新redis服务器时更新该时间戳，如长时间没有数据更新，最迟5分钟更新一下时间戳，标示自己正常运行；

竞价程序每次（一般是5秒更新）更新数据时读取该数值，如果发现10时间戳10分钟以上没有更新，则整个系统立刻停止投放，知道时间戳再次更新才重新开始投放；

**常用命令：**

GET key value

**存储结构：**

key: dsp\_adsetting\_working

value: (lasttime)

valuetype：int

value格式说明：linux时间戳

# Redis-投放控制（单机）设计

## 推广活动暂停(STRING)

**常用命令：**

SET/GET key value （设置/获取暂停的推广活动）

EXPIRE key seconds （设置key失效时间）

**存储结构：**

key: pause\_campaign\_( campaignid)

value:（reasoncode）

|  |  |
| --- | --- |
| **ReasonCode** | **comment** |
| 11 | Kpi（展示和点击）达到上限 |
| 12 | 成本控制达到上限 |
|  |  |

## 投放策略暂停(STRING)

**常用命令：**

SET/GET key value （设置/获取暂停的投放策略）

EXPIRE key seconds （设置key失效时间）

**存储结构：**

key: pause\_policy\_( policyid)

value:（reasoncode）

|  |  |
| --- | --- |
| **ReasonCode** | **comment** |
| 21 | Kpi（展示和点击）达到上限 |
| 22 | 成本控制达到上限 |
| 23 | 匀速投放达到上限 |
| 24 | 不在投放时段 |

## 推广活动频次上限(SET)

**说明：**

设置/获取已经达到推广活动设置上限的设备ID

**常用命令：**

SADD key value (写入命令)

SREM key value （删除命令）

SMEMBERS key （返回所有活动ID）

SISMEMBER key member (判断设备ID是否存在)

EXPIRE key seconds （设置key失效时间）

**存储结构：**

key: fc\_campaign\_(campaignid)

value: did\_[idfa /androidid/imei/mac]\_[md5/SHA1/Clear]\_(deviceid)

value type：string

## 投放策略频次上限(SET)

**说明：**

设置/获取已经达到投放策略设置上限的设备ID

**常用命令：**

SADD key value (写入命令)

SREM key value （删除命令）

SMEMBERS key （返回所有活动ID）

SISMEMBER key member (判断设备ID是否存在)

EXPIRE key seconds （设置key失效时间）

**存储结构：**

key: fc\_policy\_( policyid)

value: did\_[idfa /androidid/imei/mac]\_[md5/SHA1/Clear]\_(deviceid)

value type：string

## 创意频次上限(SET)

**说明：**

设置/获取已经达到推广活动设置上限的单个创意的设备ID

**常用命令：**

SADD key value (写入命令)

SREM key value （删除命令）

SMEMBERS key （返回所有活动ID）

SISMEMBER key member (判断设备ID是否存在)

EXPIRE key seconds （设置key失效时间）

**存储结构：**

key: fc\_creative\_( creativeid)

value: did\_[idfa /androidid/imei/mac]\_[md5/SHA1/Clear]\_(deviceid)

value type：string

## 投放控制更新时间标识（string）

**说明：**

该时间戳用于竞价程序判断redis写入程序是否正常工作。

跟踪模块每次更新redis服务器时更新该时间戳，如长时间没有数据更新，最迟5分钟更新一下时间戳，标示自己正常运行；

竞价程序每次（一般是5秒更新）更新数据时读取该数值，如果发现10时间戳10分钟以上没有更新，则整个系统立刻停止投放，知道时间戳再次更新才重新开始投放；

**常用命令：**

HSET key field value

HMSET key field value [field value …]

HGET key field

HMGET key field [field…]

HGETALL key

**存储结构：**

key: dsp\_adcontrol\_working

value: (lasttime)

valuetype：int

value格式说明：linux时间戳

# Redis-投放计数（集群）设计

## 人群包黑白名单及频次控制计数（HASH）

**常用命令：**

HSET key field (设置人群包和全局黑名单)

HINCRBY key field 1 （频次计数）

HGET key field （获取频次次数）

HEXISTS key field （验证是否属于人群包或者全局黑名单）

HDEL key field [field ...] （删除已经过期的频次计数field、删除人群包、删除全局黑名单）

**存储结构：**

key: did\_[idfa /androidid/imei/mac/other/unknow]\_[md5/SHA1/Clear]\_(deviceid)

fieldvalue:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **valuetype** | **comment** | **required** |
| ap\_(audienceid) | int | 当前设备id所属的人群包，可用于投放的黑名单或白名单。一个设备id可能属于多个人群包。value暂时无用，值为-1。  （audienceid）为人群包ID  field举例：4598 | false |
| bl\_global | int | 当前设备id属于平台黑名单。value暂时无用，值为-1。 | false |
| fc\_[campaignid/creativeid]\_(action)\_(period)\_(enddatetime) | int | 活动/创意级频次控制计数。当前设备id在某个活动（创意）中的展示（或点击）的某种周期（小时、天、周、月、全周期）的结束时间未endtime的频次计数。value为次数。  **field计算方法**：  根据“[单个活动的投放控制策略](#_单个订单的投放控制策略（hash）)”中的frequencytype、frequencyaction、frequencyperiod、frequencyendtime（frequencyperiod=0时）  frequencytype=1：fc\_(campaignid)\_(action)\_(period)\_(enddatetime)  frequencytype=2：fc\_(creativeid)\_(action)\_(period)\_(enddatetime)  (action)= frequencyaction；  (period)= frequencyperiod；  **enddatetime的计算方法**：  1.根据当前时间和redis中“单个项目频次策略“中的周期类型（period）、enddate来确定enddatetime。PS:enddatetime的计算规则：  user.period=1（小时）：enddatetime=当前小时的最后59分59秒，如当前时间为2017-6-28 17:58:38，则enddatetime= timestamp( 2017-6-28 17:59:59)  user.period=2（天）：enddatetime=当天的23点59分59秒，如当前时间为2017-6-28 17:58:38，则enddatetime= timestamp( 2017-6-28 23:59:59)  user.period=3（周）：enddatetime=当周最后一天（周日）晚23点59分59秒，如当前时间为2017-6-28 17:58:38，则enddatetime= timestamp( 2017-7-2 23:59:59)  user.period=4（月）：enddatetime=当月最后一天的23点59分59秒，如当前时间为2017-6-28 17:58:38，则enddatetime= timestamp( 2017-6-30 23:59:59)  user.period=0（全周期）：enddatetime=user.enddate的23点59分59秒，如当前时间为2017-6-28 17:58:38，则enddatetime= timestamp( 2017-6-30 23:59:59)  **例子**：  活动id=48832  frequencyaction =1  frequencyperiod =3  当前时间为2017-6-28 17:58:38  则：  enddatetime=timestamp(2017-7-2 23:59:59)  field=fc\_48832\_1\_3\_1499011199 | false |
| fc\_(policyid)\_(action)\_(period)\_(enddatetime) | int | 策略级频次控制计数。当前设备id在某个策略中的展示（或点击）的某种周期（小时、天、周、月、全周期）的结束时间未endtime的频次计数。value为次数。  **field计算方法**：  根据 “[单个策略的投放控制策略](#_所有的投放策略ID(SET))”中的frequencytype、frequencyaction、frequencyperiod、frequencyendtime（frequencyperiod=0时）  frequencytype=1：fc\_(campaignid)\_(action)\_(period)\_(enddatetime)  frequencytype=2：fc\_(creativeid)\_(action)\_(period)\_(enddatetime)  (action)= frequencyaction；  (period)= frequencyperiod；  **enddatetime的计算方法**：  1.根据当前时间和redis中“单个项目频次策略“中的周期类型（period）、enddate来确定enddatetime。PS:enddatetime的计算规则：  user.period=1（小时）：enddatetime=当前小时的最后59分59秒，如当前时间为2017-6-28 17:58:38，则enddatetime= timestamp(2017-6-28 17:59:59)  user.period=2（天）：enddatetime=当天的23点59分59秒，如当前时间为2017-6-28 17:58:38，则enddatetime= timestamp(2017-6-28 23:59:59)  user.period=3（周）：enddatetime=当周最后一天（周日）晚23点59分59秒，如当前时间为2017-6-28 17:58:38，则enddatetime= timestamp(2017-7-2 23:59:59)  user.period=4（月）：enddatetime=当月最后一天的23点59分59秒，如当前时间为2017-6-28 17:58:38，则enddatetime= timestamp(2017-6-30 23:59:59)  user.period=0（全周期）：enddatetime=user.enddate的23点59分59秒，如当前时间为2017-6-28 17:58:38，则enddatetime= timestamp(2017-6-30 23:59:59)  **例子**：  策略id=4598  user.action=1  user.period=3  当前时间为2017-6-28 17:58:38  则：  enddatetime=timestamp(2017-7-2 23:59:59)  field=fc\_ 4598\_1\_3\_1499011199 | false |



### 频次计数清除索引(SET)

**说明：**

以小时为单位记录频次控制计数中的key和频次field，用于清除已经过期的频次控制field

**常用命令：**

SADD key value (写入命令)

SREM key value （删除命令）

SRANDMEMBER key （随机返回一个元素命令）

SCAN cursor MATCH del\_\* (扫描所有清除清除索引的key)

**存储结构：**

key: del\_[enddatetime]

value: did\_[idfa /androidid/imei/mac]\_[md5/SHA1/Clear]\_(deviceid)&$&fc\_[campaignid/creativeid/policyid]\_(action)\_(period)\_(enddatetime)

**value说明**：

value=黑白名单及频次控制计数key+&$&+频次计数的field。

## 活动及策略点击展示价格计数（HASH）

**常用命令：**

HINCRBY key field （计数）

HGET key field （获取计数）

HDEL key field [field ...] （删除已经过期的单日计数、小时计数和10分钟时段计数）

HMSET key field value [field value ...] （设置各种计数，数据校验和初始化时使用）

DEL（项目结束后，清理并释放计数内存）

**存储结构：**

key: dspcounter\_(campaignid)

fieldvalue:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **valuetype** | **comment** | **required** |
| imp\_c | int | 活动展示数总计 | true |
| clk\_c | int | 活动点击数总计 | true |
| cost\_c | int | 活动累计花费总计，价格单位为RMB 分/CPM\*100 | true |
| imp\_p\_(policyid) | int | 投放策略展示总计，（policyid）为投放策略id。 | true |
| clk\_p\_(policyid) | int | 投放策略点击总计，（policyid）为投放策略id。 | true |
| cost\_p\_(policyid) | int | 投放策略累计花费总计，（policyid）为投放策略id。价格单位为RMB 分/CPM\*100 | true |
| d\_imp\_c\_(月日) | int | 单日活动展示计数，（月日）为当前日期。例：d\_imp\_c\_0708 | true |
| d\_clk\_c\_(月日) | int | 单日活动点击计数，（月日）为当前日期。例：d\_clk\_c\_0708 | true |
| d\_cost\_c\_(月日) | int | 单日活动累计花费，（月日）为当前日期。例：d\_cost\_c\_0708。价格单位为RMB 分/CPM\*100 | true |
| d\_imp\_p\_(policyid)\_(月日) | int | 单日投放策略展示计数，（月日）为当前日期。  例：  投放策略id=*4598*  field=d\_*imp\_p\_4598\_0708* | true |
| d\_clk\_p\_(policyid)\_(月日) | int | 单日投放策略点击计数，（月日）为当前日期。  例：  投放策略id=*4598*  field=d\_*clk\_p\_4598\_0708* | true |
| d\_cost\_p\_(policyid)\_(月日) | int | 单日投放策略累计花费，（月日）为当前日期。价格单位为RMB 分/CPM\*100  例：  投放策略id=*4598*  field=d\_*cost\_p\_4598\_0708* | true |

## 投放策略10分钟计数（HASH）

**常用命令：**

HINCRBY key field （计数）

HGET key field （获取计数）

HDEL key field [field ...] （删除已经过期的单日计数、小时计数和10分钟时段计数）

HMSET key field value [field value ...] （设置各种计数，数据校验和初始化时使用）

DEL（设置生命周期为30分钟）

**存储结构：**

key: dsp10minutecounter\_(policyid) \_ (10分钟段)

fieldvalue:

|  |  |  |  |
| --- | --- | --- | --- |
| **field** | **valuetype** | **comment** | **required** |
| imp\_count | int | 匀速投放10分钟计数展现 | true |
| clk\_count | int | 匀速投放10分钟计数点击 | true |
| cost\_count | int | 10分钟消费计数 | true |

**说明：**

key：

(policyid)：投放策略id；

(10分钟段)：将一个小时分为6个时段，取值为0~5。如果当期时间是11:29:06则所处时段为2；

ps：key设置失效时间为30分钟失效。

field:

(mapid)：创意和策略的关联ID，其信息包括了投放策略、创意的类型、创意的宽高；

(adxcode)：渠道编号，代表某个adx；

(regioncode)：地理编码，按照IPB小组标准IP库地域信息格式；

**举例：**

policyid=4598

mapid=356

adxcode=3

regioncode= 1156110000

当前时间为：11:29:06

当前时段累计价格为：234.6546元

则：

key：dsp10minutecounter\_4598 \_ 2

field：cost\_356\_3\_1156110000

value：2346546

## 自动出价计数（HASH）

key: auto\_cost\_bid\_(policyid)\_(月)\_(日)

例: auto\_cost\_bid\_11\_0101

|  |  |  |  |
| --- | --- | --- | --- |
| cost­\_(at)\_(mapid)\_(adxcode)\_ (regioncode) | int | 单个策略当日投放成本。  价格单位为RMB 分/CPM\*100 | true |
| imp­\_(at)\_(mapid)\_(adxcode)\_ (regioncode) | int | 当日展示计数 | true |
| clk­\_(at)\_(mapid)\_(adxcode)\_ (regioncode) | int | 当日点击计数 | true |
| bid­\_(at)\_(mapid)\_(adxcode)\_ (regioncode) | int | 当日出价次数 | true |
| bidprice­\_(at)\_(mapid)\_(adxcode)\_ (regioncode) | int | 当日出价累计金额  价格单位为RMB 分/CPM | true |

## 投放策略 10分钟未完成目标(SET)

**说明：**

10分开始的时候写入所有在投放的策略，当某个策略满足以下条件时，在向投放控制redis写入相应数据后，将该策略从本队列中删除：

1. 某策略满足10分钟kpi时；
2. 某策略满足当日投放kpi或当日成本上限时；
3. 某投放活动满足当日kpi或当日成本上限时，该活动下的所有投放策略；

10分钟初始化一次添加全部策略ID

已完成目标的策略/活动 在追踪程序del

**常用命令：**

SADD key value (写入命令)

SREM key value （删除命令）

SMEMBERS key （返回所有活动ID）

**存储结构：**

key: auto\_bid\_policy\_queue

value: polictid

## 设置/获取竞价/结算/展现/点击标志位（STRING）

SET/GET key value

key: dsp\_id\_flag\_(requestid)\_(impid)\_(advid)

value: 二进制标志位，从低到高第一位为1表示有竞价请求，第二位为1表示有结算价格信息，第三位为1表示有展现信息，第四位为1表示有点击信息。这个字段是目的为了去除重复的结算/展现/点击信息。

说明：

有些渠道（目前只有广点通）的出价为三个层级，每个都是一对多的关系。

有些渠道出价为2个层级一对多，为requestid→advid，这种情况下竞价引擎会将advid升级为impid处理

# 重要功能用法

## 基本广告投放

1 适配器接收流量，将请求数据构造为通用请求数据

2 检查请求数据的合法性，包括adx平台定向，各项数据是否有效（如IP是否属于国内）

3 流量与投放数据的基本信息比较，筛除部分投放数据

4 从剩余投放数据里，检查投放次数限制，检查用户频次，黑白名单检查

5 根据出价表，选最高价格的创意（可能选中多个）

6 从上一步选中的创意里，选行业关联度最高的（可能选中多个）

7 若最后选出了多个创意，则随机选中一个。

8 构造通用返回数据

9 各适配器生成响应数据并发回adx

10 发送日志

## 频次控制

**相关数据：**

|  |  |  |
| --- | --- | --- |
| **数据表** | **数据项** | **说明** |
| [单个活动的投放控制策略](#_单个订单的投放控制策略（hash）) | frequencytype | 频次控制类型 |
| frequencyaction | 频次控制行为 |
| frequencyperiod | 频次控制周期 |
| frequencyendtime | 全周期时的结束日期 |
| frequencycount | 频次控制数量 |
| [单个策略的投放控制策略](#_单个策略的投放控制策略（hash）) | frequencytype | 频次控制类型 |
| frequencyaction | 频次控制行为 |
| frequencyperiod | 频次控制周期 |
| frequencyendtime | 全周期时的结束日期 |
| frequencycount | 频次控制数量 |
| [人群包黑白名单及频次控制计数](#_黑白名单及频次控制计数（hash）) | fc\_[campaignid/creativeid/policyid]\_(action)\_(period)\_(enddatetime) | 频次控制计数 |
| [频次计数清除索引](#_频次计数清除索引(set)) | did\_[idfa /androidid/imei/mac]\_[md5/SHA1/Clear]\_(deviceid)&$&fc\_[campaignid/creativeid/policyid]\_(action)\_(period)\_(enddatetime) | 黑白名单及频次控制计数key+&$&+频次计数的field |
|  |  |  |

**Java端：**

HMSET key field value [field value …] 命令向“[单个活动的投放控制策略](#_单个订单的投放控制策略（hash）)”和“[单个策略的投放控制策略](#_单个策略的投放控制策略（hash）)”的数据项中写入相关策略数据。

**跟踪模块（C）：**

HINCRBY key field 1 命令，根据规则向“[人群包黑白名单及频次控制计数](#_黑白名单及频次控制计数（hash）)”的“fc\_[campaignid/creativeid/policyid]\_(action)\_(period)\_(enddatetime)”field进行增1的操作。

SADD key value 命令，根据规则向“[频次计数清除索引](#_频次计数清除索引(set))”写入未来要清除的键值。

**上限到达标识:**

如果达到频次上限，则向“Redis-投放控制”中的推广活动（投放策略、创意）频次上限中写入上限标记并根据enddatetime设置key的失效时间：

1. HMGET key field [field…] 命令（或HGETALL key命令）从“[单个活动的投放控制策略](#_单个订单的投放控制策略（hash）)”和“[单个策略的投放控制策略](#_单个策略的投放控制策略（hash）)”的数据项中**预读**频次控制相关数据；
2. 根据向“[人群包黑白名单及频次控制计数](#_黑白名单及频次控制计数（hash）)”的进行自增操作时返回的值和投放设置比较当前设备的展示（或点击）“推广活动”（或“单个创意”），“投放策略”是否已经达到频次上线；
3. 如果达到上限则在“[Redis-投放控制（单机）](#_Redis-投放控制（单机）设计)”中的“[推广活动频次上限](#_推广活动频次上限(SET))”（“[投放策略频次上限](#_投放策略频次上限(SET))”或“[创意频次上限](#_创意频次上限(SET))”）中设置当前设备ID；
4. 设置KEY的失效时间；

**竞价模块（C）**

SISMEMBER key member命令，根据“[单个活动的投放控制策略](#_单个订单的投放控制策略（hash）)”和“[单个策略的投放控制策略](#_单个策略的投放控制策略（hash）)”中的投放设置，分别在“[推广活动频次上限](#_推广活动频次上限(SET))”、“[投放策略频次上限](#_投放策略频次上限(SET))”、“[创意频次上限](#_创意频次上限(SET))”中获取当前设备ID是否已经达到频次上限，如果达到则不进行投放，同时写入“不出价原因”。；

**清除频次控制计数释放内存（java）**

SCAN cursor MATCH del\_\* 命令，扫描所有“[频次计数清除索引](#_频次计数清除索引(set))”的key

LOOP (key中的enddatetime<当前时间) {

LOOP(缓存清除总索引中key所有的value){

SRANDMEMBER key 从“[频次计数清除索引](#_频次计数清除索引(set))”中随机获取一个要清除的缓存

HDEL key field 从“[人群包黑白名单及频次控制计数](#_黑白名单及频次控制计数（hash）)”中删除相应的field

SREM key value 从“[频次计数清除索引](#_频次计数清除索引(set))”中删除刚才取出来的值

}

}

## 人群包

**相关数据：**

|  |  |  |
| --- | --- | --- |
| **数据表** | **数据项** | **说明** |
| [单个投放策略引用的人群包](#_单个人群包被投放策略引用信息(SET)) | (audienceid) | 指定投放策略所应用的人群包，以及引用的黑白名单 |
| [人群包黑白名单及频次控制计数](#_黑白名单及频次控制计数（hash）) | ap\_(audienceid) | 属于某个人群包 |
| bl\_global | 平台黑名单 |
|  |  |  |

**Java端：**

HMSET key field value [field value …] 命令向“[单个投放策略引用的人群包](#_单个人群包被投放策略引用信息(SET))”中写入该策略引用的人群包名。

HDEL key field 命令 当项人群包取消引用时从“[单个投放策略引用的人群包](#_单个人群包被投放策略引用信息(SET))”中删除该策略引用的人群包名。

**竞价模块（C）**

HGET key field 命令，根据规则从“[人群包黑白名单及频次控制计数](#_黑白名单及频次控制计数（hash）)”的bl\_global域中获取当前广告请求的设备ID是否属于平台黑名单。

HMGET key field [field…] 命令（或HGETALL key命令）从“[单个投放策略引用的人群包](#_单个人群包被投放策略引用信息(SET))”中获取某个投放策略所引用的人群包并确定是黑名单还是白名单。

HGET key field 命令，根据规则从“[人群包黑白名单及频次控制计数](#_黑白名单及频次控制计数（hash）)”的ap\_(audienceid) 域中获取当前广告请求的设备ID所属人群包是否属于某个投放策略。

## 投放KPI及成本控制

**相关数据：**

|  |  |  |
| --- | --- | --- |
| **数据表** | **数据项** | **说明** |
| [单个活动的投放控制策略](#_单个订单的投放控制策略（hash）) | totalimp | 活动总展示数目标 |
| totalclk | 活动总点击数目标 |
| totalbudget | 活动总预算上线 |
| dayimp\_(月日) | 活动日展示数目标 |
| dayclk\_(月日) | 活动日点击数目标 |
| daybudget\_(月日) | 活动日预算上线 |
| [单个策略的投放控制策略](#_单个策略的投放控制策略（hash）) | totalimp | 策略总展示数目标 |
| totalclk | 策略总点击数目标 |
| totalbudget | 策略总预算上线 |
| dayimp\_(月日) | 策略日展示数目标 |
| dayclk\_(月日) | 策略日点击数目标 |
| daybudget\_(月日) | 策略日预算上线 |
| [活动及策略点击展示价格计数](#_订单及策略点击展示价格计数（HASH）) | imp\_c | 活动总展示计数 |
| clk\_c | 活动总点击计数 |
| cost\_c | 活动总成本计数 |
| imp\_p\_(policyid) | 策略总展示计数 |
| clk\_p\_(policyid) | 策略总点击计数 |
| cost\_p\_(policyid) | 策略总成本计数 |
| d\_imp\_c\_(月日) | 活动日展示计数 |
| d\_clk\_c\_(月日) | 活动日点击计数 |
| d\_cost\_c\_(月日) | 活动日成本计数 |
| d\_imp\_p\_(policyid)\_(月日) | 策略日展示计数 |
| d\_clk\_p\_(policyid)\_(月日) | 策略日点击计数 |
| d\_cost\_p\_(policyid)\_(月日) | 策略日成本计数 |
| [ADX定价合同价格信息](#_ADX定价合同价格信息（HASH）) | (dealid)\_[imp/clk] | 合同号和展示/点击选项 |

**Java端：**

HMSET key field value [field value …] 命令向“[单个活动的投放控制策略](#_单个订单的投放控制策略（hash）)”和“[单个策略的投放控制策略](#_单个策略的投放控制策略（hash）)”中写入活动和策略的各项控制指标。

**清除缓存定时器（JAVA）：**

**DEL key [key ...]** 命令 凌晨2点时从“[单个活动的投放控制策略](#_单个订单的投放控制策略（hash）)” 中删除前一天已经结束的相关活动key，同时删除“[单个策略的投放控制策略](#_单个策略的投放控制策略（hash）)”中前一天已经结束的相关活动的策略key。

**DEL key [key ...]** 命令 凌晨2点时从“[活动及策略点击展示价格计数](#_订单及策略点击展示价格计数（HASH）)”中删除前一天已经结束投放的活动key。

HDEL key field 命令 凌晨2点时从“[单个活动的投放控制策略](#_单个订单的投放控制策略（hash）)”和“[单个策略的投放控制策略](#_单个策略的投放控制策略（hash）)”中删除前一天的投放指标控制数据。

HDEL key field 命令 凌晨2点时从“[活动及策略点击展示价格计数](#_订单及策略点击展示价格计数（HASH）)”中删除前一天的各项单日计数数据。

**跟踪模块（C）**

HINCRBY key field 命令 向“[活动及策略点击展示价格计数](#_订单及策略点击展示价格计数（HASH）)”写入计数数据。

**成本计数**:

成本控制需要计算当前请求价格，定价和竞价有所区别：

竞价：根据赢价通知中的价格字段（price）或者展示通知中的价格字段（price）；

定价：根据展示（或点击）通知中传入的adx，dealid，从“[ADX定价合同价格信息](#_ADX定价合同价格信息（HASH）)”中获取价格；

**上限到达标识：**

根据投放设置当KPI或成本到达上限时，则向“Redis-投放控制”中的推广活动（投放策略、创意）频次上限中写入上限标记并将次日0:00:00设置为key的失效时间：

1. HMGET key field [field…] 命令（或HGETALL key命令）从“[单个活动的投放控制策略](#_单个订单的投放控制策略（hash）)”和“[单个策略的投放控制策略](#_单个策略的投放控制策略（hash）)”中获取所有活动和投放策略的投放控制指标并缓存于本地；
2. 根据HINCRBY key field 命令 向“[活动及策略点击展示价格计数](#_订单及策略点击展示价格计数（HASH）)”写入计数数据时返回的计数值进行比较；
3. 如果KPI（如果展示和点击都有目标，则全部完成目标才算到达上限）或者成本达到设定上限，则向“[Redis-投放控制（单机）](#_Redis-投放控制（单机）设计)”中的“[推广活动暂停](#_推广活动暂停(STRING))”或“[投放策略暂停](#_投放策略暂停(STRING))”中写入标识，并设置key的失效时间为次日凌晨0:00:00；

**竞价模块（C）**

HMGET key field [field…] 命令从 “[Redis-投放控制（单机）](#_Redis-投放控制（单机）设计)”中的“[推广活动暂停](#_推广活动暂停(STRING))”或“[投放策略暂停](#_投放策略暂停(STRING))”中预读（每隔5秒更新数据）投放暂停信息，如果有推广活动或投放策略的暂停信息，则该推广活动或投放策略不进行投放，同时写入“不出价原因”。；

**广告请求时比对各项指标要求和计数值，规则如下**：

1. 活动（或策略）总展示指标和活动（或策略）总点击指标均有值，且都**大于**0时：活动（或策略）总展示计数**小于**活动（或策略）总展示指标**或**活动（或策略）总点击计数**小于**活动（或策略）总点击指标则该活动（或策略）继续投放；
2. 活动（或策略）总展示指标和活动总击指标均有值，只有一个大于0，另一个等于-1时：则大于0的那一项的计数值**小于**指标时活动（或策略）继续投放；
3. 活动（或策略）总预算指标大于0时：活动（或策略）总预算计数**小于总**指标值活动（或策略）继续投放；
4. 活动（或策略）总预算指标等于-1时：活动（或策略）继续投放；
5. 活动（或策略）当日展示指标和活动（或策略）当日点击指标均有值，且都**大于**0时：活动（或策略）当日展示计数**小于**活动（或策略）当日展示指标**或**活动（或策略）当日点击计数**小于**活动（或策略）当日点击指标则该活动（或策略）继续投放；
6. 活动（或策略）当日展示指标和活动当日击指标均有值，只有一个大于0，另一个等于-1时：则大于0的那一项的计数值**小于**指标时活动（或策略）继续投放；
7. 活动（或策略）当日预算指标大于0时：活动（或策略）当日预算计数**小于**当日指标值活动（或策略）继续投放；
8. 活动（或策略）当日预算指标等于-1时：活动（或策略）继续投放；

**不投放情况：**

|  |  |  |  |
| --- | --- | --- | --- |
| **情况** | **说明** | **类型** | **后续处理** |
| 活动（或策略）没有总展示、总点击或总预算指标 | 这几项是必填项 | 错误 | 错误日志 |
| 活动（或策略）总展示、总点击或总预算指标等于0或小于-1 | 0和小于-1的值都是非法数值 | 错误 | 错误日志 |
| 活动（或策略）同时没有当日展示、当日点击和当日预算的指标 | 这代表活动（或策略）当日不投 | 正常逻辑 | 无 |
| 活动（或策略）的当日展示、当日点击和当日预算指标中有一项（或2项）没有 | 如果当日不投则展示、点击和预算3项指标当日都没有 | 错误 | 错误日志 |
| 活动（或策略）的当日展示、当日点击和当日预算指标中有一项（或2项）等于0或小于-1 | 0和小于-1的值都是非法数值 | 错误 | 错误日志 |
|  |  |  |  |

## 匀速投放

**相关数据：**

|  |  |  |
| --- | --- | --- |
| **数据表** | **数据项** | **说明** |
| [单个策略的投放控制策略](#_单个策略的投放控制策略（hash）) | uniform | 是否匀速投放 |
| dayimp\_(月日) | 策略日展示数目标 |
| dayclk\_(月日) | 策略日点击数目标 |
| allowtime\_(月日) | 允许时段 |
| [活动及策略点击展示价格计数](#_订单及策略点击展示价格计数（HASH）) | d\_imp\_p\_(policyid)\_(月日) | 策略日展示计数 |
| d\_clk\_p\_(policyid)\_(月日) | 策略日点击计数 |
| [投放策略10分钟计数](#_设置/获取竞价/结算/展现/点击标志位（STRING）) | imp | 10分钟时段内展示计数 |
| clk | 10分钟时段内点击计数 |

**Java端：**

HMSET key field value [field value …] 命令向 “[单个策略的投放控制策略](#_单个策略的投放控制策略（hash）)”中写入策略的allowtime\_(月日)和unifrom数据。（其它相关指标已经在“投放KPI及成本控制功能”中已经描述，注意不要重复写入）

**清除缓存定时器（JAVA）：**

HDEL key field 命令 凌晨2点时从 “[单个策略的投放控制策略](#_单个策略的投放控制策略（hash）)”中删除前一天的allowtime\_(月日)数据。（其它单日kpi目标数据已经在“投放KPI及成本控制功能”删除）

**跟踪模块（C）**

HINCRBY key field 命令 向“[活动及策略点击展示价格计数](#_订单及策略点击展示价格计数（HASH）)” 和“[投放策略10分钟计数](#_设置/获取竞价/结算/展现/点击标志位（STRING）)”写入计数数据。（已经在“投放KPI及成本控制功能”中已经写入）

**上限到达标识：**

如果某个策略的匀速投放达到上限，则在“[Redis-投放控制（单机）](#_Redis-投放控制（单机）设计)”中的 “[投放策略暂停](#_投放策略暂停(STRING))”中写入标识，并设置key的失效时间为当前10分钟时段的最后一秒：

1. HMGET key field [field…] 命令（或HGETALL key命令） “[单个策略的投放控制策略](#_单个策略的投放控制策略（hash）)”中获取所有投放策略的投放控制（这里主要是是否匀速投放）指标并缓存于本地；
2. 如果匀速投放从“[活动及策略点击展示价格计数](#_活动及策略点击展示价格计数（HASH）)”和“[投放策略10分钟计数](#_设置/获取竞价/结算/展现/点击标志位（STRING）)”中获取活动和策略的展示、点击的当日数据和当前10分钟数据；
3. 计算当前10分钟目标展示、点击数量，方法如下：
4. 日剩余展示量=日展示数目标-日展示计数；日剩余点击量=日点击数目标-日点击计数；
5. 日剩余时段数=允许时段（allowtime\_(月日)）所记录时段中在当前时段之后的所有时段数量之和。如当前时间是8:16:10，允许时段值为11100111100，则日剩余时段数=2。允许时段的记录方法详见“[单个策略的投放控制策略](#_单个策略的投放控制策略（hash）)”中allowtime；
6. 当前时段10分钟数=本小时内当前所属10分钟时段及本小时在当前10分钟时段之后剩下10分钟时段的总合。如：当前时间是8:16:10，则当前时段10分钟数=5；
7. 日剩余10分钟时段数=当前时段10分钟数+日剩余时段数\*6；
8. 日可投10分钟时段数=if(日剩余10分钟时段数>=2）{日剩余10分钟时段数-2}else{ 1}；(留20分钟的缓冲，如果不能提前20分钟完成目标，则后两个10分钟时段全速投放)
9. 当前10分钟目标展示量=日剩余展示量/日可投10分钟时段数; 当前10分钟目标点击量=日剩余点击量/日可投10分钟时段数；
10. 当前10分钟目标展示、点击量与当前10分钟已发生展示、点击数据比对，如果已发生的展示、点击数量均超过目标展示、点击数量，则当前10分钟停止投放；
11. 如果判断为当前10分钟停止投放，则向“[Redis-投放控制（单机）](#_Redis-投放控制（单机）设计)”中的 “[投放策略暂停](#_投放策略暂停(STRING))”中写入标识，并设置key的失效时间为当前10分钟时段的最后一秒

**竞价模块（C）**

HMGET key field [field…] 命令从 “[Redis-投放控制（单机）](#_Redis-投放控制（单机）设计)”中的 “[投放策略暂停](#_投放策略暂停(STRING))”中预读（每隔5秒更新数据）投放暂停信息，如果有推广活动或投放策略的暂停信息，则该推广活动或投放策略不进行投放，同时写入“不出价原因”。

## 出价控制

**相关数据：**

|  |  |  |
| --- | --- | --- |
| **数据表** | **数据项** | **说明** |
| [单个创意的出价信息](#_单个创意的出价信息（HASH）) | (adxcode)\_(mediacode)\_( regioncode) | 精确到媒体和地区的出价。 |
| (adxcode)\_(mediacode)\_all | 指定媒体但不指定地区的出价。 |
| (adxcode)\_all\_( regioncode) | 指定adx和地区，但不指定媒体的出价。 |
| (adxcode)\_all\_all | 指定adx但不指定媒体和地区的出价。 |
| all\_all\_( regioncode) | 只指定地区的出价。 |
| all\_all\_all | 没有指定任何条件的，本mapid的出价。 |

**Java端：**

HMSET key field value [field value …] 命令向 “[单个创意的出价信息](#_单个创意的出价信息（HASH）)”中写入出价信息。

**清除缓存定时器（JAVA）：**

**DEL key [key ...]** 命令 凌晨2点时从“[单个创意的出价信息](#_单个创意的出价信息（HASH）)”中删除前一天已经结束投放的活动中涉及的mapid。

**竞价模块（C）**

HMGET key field [field…] 命令（或HGETALL key命令）从 “[单个创意的出价信息](#_单个创意的出价信息（HASH）)”中获取所有mapid的所有价格并缓存在本地。

竞价时以[“单个创意的出价信息”的使用说明](#确定出价规则)中的方式确定出价。

## 竞价刹车机制

**相关数据：**

|  |  |  |
| --- | --- | --- |
| **数据表** | **数据项** | **说明** |
| [**广告信息设置更新时间标识**](#_广告信息设置更新时间标识（string）) | dsp\_adsetting\_working | java后台更新广告信息心跳时间戳 |
| [**投放控制更新时间标识**](#_投放控制更新时间标识（string）) | dsp\_adcontrol\_working | 跟踪系统投放控制心跳时间戳 |

**Java端：**

每次更新广告信息时用SET key value命令向 “[**广告信息设置更新时间标识**](#_广告信息设置更新时间标识（string）)”中写入心跳时间戳。如无广告信息更新，最迟5分钟更新一下时间戳，标示自己正常运行。

**跟踪模块（C）：**

每次更新投放控制信息时用SET key value命令向 “[**投放控制更新时间标识**](#_投放控制更新时间标识（string）)”中写入心跳时间戳，最迟5分钟更新一下时间戳，标示自己正常运行。

**竞价模块（C）**

用GET KEY分别从“[**广告信息设置更新时间标识**](#_广告信息设置更新时间标识（string）)”和“[**投放控制更新时间标识**](#_投放控制更新时间标识（string）)”获取最后一次更新时间。

如果，任意一个超过10分钟没有更新，则立刻停止所有投放并发出错误日志，声明redis工作不正常停止投放；

如果由于此原因停止投放，竞价模块仍然持续获取时间戳，直到时间戳有更新并确定所有时间标识都正常则继续开始投放，同时发出错误日志，声明已经恢复投放。

## deeplink支持

1. 在[Common Request](#_Common_Response)中判断support\_deep\_link是否存在且等于1，来判断请求流量是否支持deeplink。如果请求流量支持deeplink，则对deeplink的创意优先匹配；
2. 在“[获取单个创意的详细信息](#_获取单个创意的详细信息（STRING）)”中找到deeplink有值的创意，优先进行匹配。如果匹配成功则在response中按照deeplink的方式进行物料回填：
   1. 将“[获取单个创意的详细信息](#_获取单个创意的详细信息（STRING）)”中的deeplink填写到Common Response→Seat bid→bid的deeplink中；
   2. 将“[获取单个创意的详细信息](#_获取单个创意的详细信息（STRING）)”中的curl（或securecurl）填入Common Response→Seat bid→bid的curl中；

## 自动出价调整

**相关数据：**

|  |  |  |
| --- | --- | --- |
| **数据表** | **数据项** | **说明** |
| [单个策略的投放控制策略](#_单个策略的投放控制策略（hash）) |  |  |
| dayimp\_(月日) | 策略日展示数目标 |
| dayclk\_(月日) | 策略日点击数目标 |
| allowtime\_(月日) | 允许时段 |
| [活动及策略点击展示价格计数](#_订单及策略点击展示价格计数（HASH）) | d\_imp\_p\_(policyid)\_(月日) | 策略日展示计数 |
| d\_clk\_p\_(policyid)\_(月日) | 策略日点击计数 |
| [投放策略10分钟计数](#_设置/获取竞价/结算/展现/点击标志位（STRING）) | cost­\_(at)\_(mapid)\_(adxcode)\_ (regioncode) | 10分钟时段内展示计数 |
| imp­\_(at)\_(mapid)\_(adxcode)\_ (regioncode) | 10分钟时段内点击计数 |
| clk­\_(at)\_(mapid)\_(adxcode)\_ (regioncode) | 单个策略10分钟时段的投放成本 |
| bid­\_(at)\_(mapid)\_(adxcode)\_ (regioncode) | 当前10分钟时段出价次数 |
| bidprice­\_(at)\_(mapid)\_(adxcode)\_ (regioncode) | 当前10分钟时段出价累计金额，价格单位为RMB 分/CPM\*100 |

**投放模块/跟踪模块/自动出价模块(C)**

计数：HINCRBY key field 命令 向“[投放策略10分钟计数](#_设置/获取竞价/结算/展现/点击标志位（STRING）)”写入计数数据。（已经在“投放KPI及成本控制功能”中也有写入描述，注意不要重复写入）；

1. **模块**分工
   * 1. **跟踪模块**：发送10分钟未到完成KPI的消息, 并删除auto\_bid\_policy\_queue 中的策略ID
     2. **自动出价格模块C**：每10分钟运行一次，发送10分钟已到未完成KPI的消息，并初始化auto\_bid\_policy\_queue, 只添加dsp\_policy\_info->autobid = 1 的策略ID进入队列
     3. **投放模块：计数**bid­\_(at)\_(mapid)\_(adxcode)\_ (regioncode)/bidprice­\_(at)\_(mapid)\_(adxcode)\_ (regioncode)
2. 出价计算通知：传入参数如下：
   * 1. info[time] 本次消距离本次10分钟时段剩余时间，以秒为单位当前10分钟时段已经过去的时间，比如：当前十分钟已经过去6分钟，则此处为60\*6=360
     2. info[policyid] 要计算的投放策略ID
     3. info[surplus\_10min] 剩余10分钟时段数
     4. info[daily\_imp\_kpi] 当日要完成的展示KPI
     5. info[daily\_clk\_kpi] 当日要完成的点击KPI
     6. info[daily\_imp\_count] 当日已完成的展示量
     7. info[daily\_clk\_count] 当日已完成的点击量
     8. info[this\_10min\_imp\_kpi] 当前十分钟展现KPI
     9. info[this\_10min\_clk\_kpi] 当前十分钟点击KPI
     10. info[this\_10min\_imp\_count] 当前十分钟展现量
     11. info[this\_10min\_clk\_count] 当前十分钟点击量
     12. info[next\_10min\_imp\_kpi] 下个十分钟展现KPI
     13. info[next\_10min\_clk\_kpi] 下个十分钟点击KPI
3. 自动出价格PYTHON模块(根据传入通知读取count信息)
   * 1. count[cost\_(at)\_(mapid)\_(adxcode)\_(regioncode)] 10分钟时段的投放成本
     2. count[imp\_(at)\_(mapid)\_(adxcode)\_(regioncode)] 10分钟时段内展示计数
     3. count[clk\_(at)\_(mapid)\_(adxcode)\_(regioncode)] 10分钟时段内点击计数
     4. count[bid\_(at)\_(mapid)\_(adxcode)\_(regioncode)] 当前10分钟时段出价次数
     5. count[bidprice\_(at)\_(mapid)\_(adxcode)\_(regioncode)] 当前10分钟时段出价累计金额
     6. …
4. 自动出价格PYTHON模块 写入新的出价信息

bid = BID(obj) # 线程中创建

# mapid, adxcode, mediacode, regioncode, value

bid.write("888", "1", "2", "3", "-1")

bid.flush() # 写入 HSET dsp\_mapid\_bid\_(mapid)

# 展现/点击/结算

由于展现/点击和赢价部分的请求参数是自己定义的，因此可以写成一个公用的程序，来支持各家的Adx平台。包含的所有变量如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 参数名 | 数据类型 | 取值范围 | 含义 | 必须 |
| 1 | adx | int | 1~26 | Adx编号 | 是 |
| 2 | mtype | string | w: 赢价通知  i: 展示通知  c: 点击通知 | 点击展现类型 | 是 |
| 3 | mapid | string | 0～9、a～f、A～F  32位：xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxxxx | 创意ID | 是 |
| 4 | bid | string |  | Request ID | 是 |
| 5 | deviceid | string |  | 设备ID | 是 |
| 6 | deviceidtype | string | 0x00: unknown  0x10: IMEI/无算法  0x11: IMEI/SHA1  0x12: IMEI/MD5  0x20: MAC/无算法  0x21: MAC/SHA1  0x22: MAC/MD5  0x60: ANDROIDID/无算法  0x61: ANDROIDID/SHA1  0x62: ANDROIDID/MD5  0x70: IDFA/无算法  0x71: IDFA/SHA1  0x72: IDFA/MD5  0xFF: OTHER/其他算法 | 设备ID类型（[参考“Didtype/Dpidtype”](#_Didtype/Dpidtype)） | 是 |
| 7 | impid | string |  | 广告展示ID | 是 |
| 8 | at | int | null：竞价 0：竞价 1：定价 |  | 否 |
| 9 | nrtb | string | null:实时竞价  n：实时竞价  y:非实时竞价(媒体外采) |  | 否 |
| 10 | price | string | 价格统一单位处理, 统一成千次展现分（RMB 分/CPM）:  adx = 2、10、14、22: price / 100  adx = 3、13、24: price \* 100  adx = 16: pirce / 10  adx = 其余: price 不变 | 价格:  货币单位（RMB）  计费方式（CPM）  价格单位(RMB 分/CPM)， 小数点后2位 | 否 |
| 11 | impt | int | 0: unknown  1: 图片广告（banner）  2：视频广告（video）  3：信息流（Native） |  | 否 |
| 12 | impm | string | impt=1时 impm=instl,splash  instl：  0：未知  1：插屏  2：全屏  splash:  0：未知  1：开屏  // 注: 该数据来自于流量， kafka 的判断是:  //0,0：横幅  //0,1: 开屏  //1,0：插屏/全屏  //1,1: 未知  //2,0：插屏/全屏  //2,1: 未知 | 广告展示机会的图片展示形式  （和图片创意展示形式相对应） | 否 |
|  | Impt=2时，无impm参数 | 广告展示机会的视频展示形式  （和视频创意展示形式相对应） |
| string | impt=3时 impm=小图数量,大图数量,标题数量,描述数量  数量含义由 common/bid\_filter.cpp >>> filter\_creative\_by\_btype\_battr\_size\_price\_cb () 定义:  小图数量: cimp.native.asset.img.type == ASSET\_IMAGETYPE\_ICON 的数量  大图数量：cimp.native.asset.img.type == ASSET\_IMAGETYPE\_MAIN 的数量  标题数量：cimp.native.asset.type == NATIVE\_ASSETTYPE\_TITLE 的数量  描述数量：cimp.native.asset.data.type==ASSET\_DATATYPE\_DESC 的数量 | 广告展示机会的信息流展示形式  （和信息流创意展示形式相对应）  storm 需要将该字段值转为几图几文的说法 |
| 13 | w | int |  | 广告展示机会宽度: [640]\*360 （和创意宽度相对应）  如果创意是图片，则是图片的宽  如果创意是视频，则是视频的宽  如果创意是信息流，则有大图选第一个大图的宽，无大图选第一个icon的宽 | 否 |
| 14 | h | int |  | 广告展示机会高度： 640\*[360]（和创意高度相对应）  如果创意是图片，则是图片的宽  如果创意是视频，则是视频的宽  如果创意是信息流，则有大图选第一个大图的高，无大图选第一个icon的高 | 否 |
| 15 | curl | string | 1.如果点击通知中有 curl，即是 302 跳转url | 302 跳转 url | 否 |
| 16 | appid | string |  | 媒体ID | 否 |
| 17 | nw | int | 0: unknown  1: Wifi网络  2: 蜂窝数据网络-未知  3: 蜂窝数据网络-2G  4: 蜂窝数据网络-3G  5: 蜂窝数据网络-4G  6: 蜂窝数据网络-5G | 联网类型 | 否 |
| 18 | os | int | 0: unknown  1: IOS  2: Android  3: Windows | 移动设备操作系统 | 否 |
| 19 | gp | int | 1156000000～1158000000 | 地理位置编码，精确到城市 | 否 |
| 20 | tp | int | 0：unknown  1：Phone  2：Tablet  3：PC（目前没用） | 移动设备类型 | 否 |
| 21 | mb | int | 1～255  1: iphone (苹果)  2: samsung（三星）  3：xiaomi（小米）  …  255：other（其他） | 移动设备品牌 | 否 |
| 22 | md | string |  | 移动设备型号 | 否 |
| 23 | op | int | 0：unknown  1：China Mobile  2：China Unicom  3：China Telecom | 移动设备运营商，就是 API 文档中的 carrier | 否 |
| 24 | ds | int | null：dsp  0：dsp  1：pap | 数据平台来源 | 否 |

请求格式e.g.

赢价：

http://track.pxene.com/tracker?mtype=w&bid=XXX&at=0&mapid=XXX&impid=XXX&impt=1&impm=1,0&w=640&h=360&deviceid=XXX&deviceidtype=XXX&price=XXX&adx=7&appid=XXX&nw=XXX&os=XXX&gp=XXX&tp=XXX&mb=XXX&md=XXX&op=XXX&ds=0

展现：

http://track.pxene.com/tracker?mtype=i&bid=XXX&at=0&mapid=XXX&impid=XXX&impt=1&impm=2,1&w=640&h=360&deviceid=XXX&deviceidtype=XXX&price=XXX&adx=8&appid=XXX&nw=XXX&os=XXX&gp=XXX&tp=XXX&mb=XXX&md=XXX&op=XXX&ds=0

点击：

[http://track.pxene.com/tracker?mtype=c&bid=XXX&at=0&mapid=XXX&impid=XXX&impt=1&impm=1,0&w=640&h=360&deviceid=XXX&deviceidtype=XXX&curl=XXX&adx=8&appid=XXX&nw=XXX&os=XXX&gp=XXX&tp=XXX&mb=XXX&md=XXX&op=XXX&ds=1](http://track.pxene.com/tracker?mtype=c&bid=XXX&at=0&impt=1&impm=1,0&w=640&h=360&deviceid=XXX&deviceidtype=XXX&price=XXX&adx=8&appid=XXX&nw=XXX&os=XXX&gp=XXX&tp=XXX&mb=XXX&md=XXX&op=XXX&ds=dsp)

## 赢价/展现/点击流程图

流程图，如下图5.1，详细流程图可参考“追踪系统流程图”文档



图5.1程序流程图

### 价格解密

由于各家平台的价格加密方法不一致，导致在结算程序中需要使用不同的程序来对价格进行解密，使用的方法是动态加载动态库，使用配置文件来添加动态库的路劲。

定义结构体:

*struct adx\_content*

*{*

*int index; //表示是哪个平台的*

*void \*dl\_handle; //该平台对应的动态库句柄*

*};*

*Typedef struct adx\_content ADX\_CONTENT;*

配置文件price.conf定义：（目前使用数字定义来表示各个平台）

*#sample configuration file, the path of adx*

*1 = ./amax.so ## .so文件是用于解密的动态链接库，对外暴露了一个*

*2 = ./tanx.so ## decode函数用于解密。由于平台不一样，因此解密的*

*3 = ./inmobi.so ## 方式也不一样*

*4 = ./letv.so*

*5 = ./youku.so*

依次读取price.conf的值，分别将这些数据存入vector<ADX\_CONTENT>中。解密的时候（在请求里面会增加adx=?，表示哪个平台），根据adx的值来拿到dl\_handle，使用dlsym函数导出DecodeWinningPrice函数，就可以解密成功了。



图5.2 多平台价格解密的过程

# 日志模块（待更新）

## 日志文件内容的一般格式

一般来说，每条信息均会记录以下几个重要数据：

事件发生的日期和时间；

日志级别

日志的实际数据内容

## 日志的配置文件

在日志程序启动及运行过程中会读取配置文件相应字段来执行不同的操作。

日志的配置文件保存日志文件大小限制，日志输出级别以及一些其他辅助信息。配置文件格式如下：

例如：

[log]

logsize=1024(单位为字节)

loglevel=1

日志等级说明

|  |  |  |
| --- | --- | --- |
| 等级 | 等级名称 | 说明 |
| 1 | INFO | 仅是基本的信息说明 |
| 2 | WARNING | 警示的信息，可能有问题，但还不至于影响程序运行 |
| 3 | ERROR | 错误信息 |
| 4 | DEBUG | 调试信息 |

## 日志文件的迁移

日志文件需要占据很大的存储，需要考虑日志文件的保留时效以及后续日志迁移。

## 日志模块对外提供接口

### openlog

打开日志

语法：

uint64\_t openLog(char \* conffile, char \*section, bool is\_textdata, bool is\_print\_queue\_size = false);

参数：

conffile：配置文件路径

section：指定日志使用哪个section

is\_textdata：是否为文本文件

is\_print\_queue\_size：是否打印队列大小

返回值：

返回打开的文件描述符

### writelog

写日志

语法：

int  writeLog(const uint64\_t logid,  uint8\_t loglevel,  char \*fmt, ...);

int  writeLog(const uint64\_t logid,  uint8\_t loglevel, string data);

参数：

logid：打开的文件描述符

loglevel：日志级别

fmt：可变参数

data：打印的数据

返回值：

0表示成功，1代表失败

### closelog

关闭日志文件

语法：

int closeLog(const uint64\_t logid);

参数：

logid：打开的文件描述符

返回值：

0表示成功，1代表失败

### 竞价与追踪日志

#### 追踪日志分类说明及格式

|  |  |  |  |
| --- | --- | --- | --- |
| **追踪日志共五大类别** | | | |
| 模块类别 | 模块名称 | 类型 | 说明 |
| 1. 本地日志模块 | Local\_log | USERLOG obj | a.初始化信息 b. 读、写、解析、查询 操作 |
| 1. 本地flume日志模块 | Local\_flume | USERLOG obj | 发往flume的日志数据失败时，记录到本地日志flume， 以便后续对数, 记录连接flume 失败次数【重连】 |
| 1. 本地kafka 日志模块 | Local\_kafka | USERLOG obj | 发往kafka的日志数据失败时，记录到本地日志kafka， 以便后续对数, 记录连接kafka 失败次数【重连】 |
| 1. 远程flume 日志模块 | G\_datatransfer\_f | DATATRANSFER obj array | 追踪错误日志 发的远程flume 的信息 |
| 1. 远程kafka 日志模块 | G\_datatransfer\_g | DATATRANSFER obj array | 追踪正确/错误日志 发的远程kafka 的信息 |

格式示例及说明

|  |  |
| --- | --- |
| 分类 | 示例 |
| 参数解析错误 | Incoming parameter is invalid, err:[0x%x], in:[%s], on:[%d] |
| 发送flume信息失败 | Send flume log failed, err:[0x%x], in:[%s], on:[%d] |
| 连接/读写reids失败 | Get id flag reconnect failed, err:[0x%x] |
| 重复的winnotice错误 | Repeated winnotice, err:[0x%x], bid:[%s], impid:[%s], mapid:[%s], adx:[%d], at:[%d], mtype[%c] |
| 重复的imp 错误 | Repeated impnotice, err:[0x%x], bid:[%s], impid:[%s], mapid:[%s], adx:[%d], at:[%d], mtype[%c] |
| 重复的 clk 错误 | Repeated clknotice, err:[0x%x], bid:[%s], impid:[%s], mapid:[%s], adx:[%d], at:[%d], mtype[%c] |
| 价格解析错误 | Decode price is zero, err:[0x%x], bid:[%s], impid:[%s], mapid:[%s], adx:[%d], at:[%d], mtype[%c], price:[0], on:[%s], in:[%d] |
| 频次计算错误 | Bid:[%s], mtype:[%c], frequencycapping add failed |
| Kafka 信息 | time=XXX|mtype=x|ip=XXX|bid=XXX|at=0|mapid=XXX|impid=XXX|impt=1|impm=1,0|  w=640|h=360|deviceid=XXX|deviceidtype=XXX|adx=9|curl=1|appid=XXX|nw=XXX|  os=XXX|gp=XXX|tp=XXX|mb=XXX|md=XXX|op=XXX|ds=1|  iv=1|ic=十六进制错误码，多个错误以逗号隔开  (点击日志格式)  time=XXX|mtype=x|ip=XXX|bid=XXX|at=0|mapid=XXX|impid=XXX|impt=1|impm=1,0|  w=640|h=360|deviceid=XXX|deviceidtype=XXX|adx=9|price=xxx|appid=XXX|nw=XXX|  os=XXX|gp=XXX|tp=XXX|mb=XXX|md=XXX|op=XXX|ds=1|  iv=1|ic=十六进制错误码，多个错误以逗号隔开  (赢价展现日志格式) |

#### 竞价程序日志

从业务层面有三类工作线程，故竞价程序日志分三种类别:

注： 程序数据初始化及redis数据初始化错误信息不做，直接退出

|  |  |
| --- | --- |
| 分类 | 说明 |
| 1.广告数据更新及读取线程 | 对应双缓存模块 |
| 2.竞价线程 | 对应doit 函数 适配器，过滤策略/创意 数据回填 |
| 3.日志发送线程 | 对应 connect\_log\_server 发送 |

发往flume 日志数据

1.针对请求 adapter 适配器数据错误日志

2.实际竞价过滤包括频次、创意、策略 错误信息全部放入一个结构体中， 最后发送给flume

一条请求bid 对应多个策略及策略下创意的错误信息,同时添加日志模块分类信息

表现形式

例： bid : 123 A/B/C {策略1：{创意1：错误信息（错误码）}，策略2：{创意2：错误信息（错误码）}}

3.Connect\_log\_server 发送日志模块错误信息

日志格式： 新竞价程序（待定）

# 附录

## Adx编号

|  |  |
| --- | --- |
| 取值（int） | 描述 |
| 0 | Unknown（调试专用） |
| 1 | Tanx |
| 2 | Amax |
| 3 | Inmobi |
| 4 | iWifi |
| 5 | Letv |
| 6 | Toutiao |
| 7 | Zplay |
| 8 | Baidu |
| 9 | Sohu |
| 10 | Adwalker |
| 11 | Iqiyi |
| 12 | Goyoo |
| 13 | Iflytek |
| 14 | Adview |
| 15 | H2 |
| 16 | Google |
| 17 | BaiduVideo |
| 18 | Meizu |
| 19 | 16wifi |
| 20 | moji |
| 21 | gdt |
| 22 | lefee |

## 广告行业/APP类型

参考《行业分类对应表》和《APP分类对应表》。

## Imp type

|  |  |
| --- | --- |
| 取值（int） | 描述 |
| 0 | 未知 |
| 1 | Banner |
| 2 | Video |
| 3 | Native |

## Native

### Layout

|  |  |
| --- | --- |
| 取值（int） | 描述 |
| 0 | 未知 |
| 1 | Content Wall |
| 2 | App Wall |
| 3 | News Feed |
| 4 | Chat List |
| 5 | Carousel |
| 6 | Content Stream |
| 7 | Grid Adjoining The Content |

### Asset type

|  |  |
| --- | --- |
| 取值（int） | 名称 |
| 0 | 未知 |
| 1 | Title |
| 2 | Image |
| 3 | Data |

### Asset Image type

|  |  |  |
| --- | --- | --- |
| 取值（int） | 名称 | 描述 |
| 0 | 未知 |  |
| 1 | Icon | Icon image |
| 2 | Logo | Logo image for the brand/app. |
| 3 | Main | Large image preview for the ad |

### Asset Data type

|  |  |  |  |
| --- | --- | --- | --- |
| 取值（int） | 名称 | 描述 | 格式 |
| 0 | 未知 |  |  |
| 1 | Sponsored | Sponsored By message where response should  contain the brand name of the sponsor. | text |
| 2 | Desc | Descriptive text associated with the product or  service being advertised. | text |
| 3 | Rating | Rating of the product being offered to the user. For example an app’s rating in an app store from 0 -5. | number formatted as string |
| 4 | Likes | Number of social ratings or “likes” of the product  being offered to the user. | number formatted as string |
| 5 | Downloads | Number downloads/installs of this product | number formatted as string |
| 6 | Price | Price for product / app / in -app purchase. Value should include currency symbol in localised format. | number formatted as string |
| 7 | Saleprice | Sale price that can be used together with price to indicate a discounted price compared to a regular price. Value should include currency symbol in localised format. | number formatted as string |
| 8 | Phone | Phone number | formatted  string |
| 9 | Address | Address | text |
| 10 | Desc2 | Additional descriptive text associated with the product or service being advertised | text |
| 11 | Displayurl | Display URL for the text ad | text |
| 12 | Ctatext | CTA description - descriptive text describing a ‘call  to action’ button for the destination URL. | text |

## Didtype/Dpidtype

### Didtype

|  |  |  |
| --- | --- | --- |
| 取值（int） | ID类型 | 哈希算法 |
| 0x0 | 未知 | 未知 |
| 0x10 | IMEI | 无 |
| 0x11 | IMEI | SHA1 |
| 0x12 | IMEI | MD5 |
| 0x20 | MAC | 无 |
| 0x21 | MAC | SHA1 |
| 0x22 | MAC | MD5 |
| 0xFF | OTHER | 其他 |

### Dpidtype

|  |  |  |
| --- | --- | --- |
| 取值（int） | ID类型 | 哈希算法 |
| 0x0 | 未知 | 未知 |
| 0x60 | ANDROIDID | 无 |
| 0x61 | ANDROIDID | SHA1 |
| 0x62 | ANDROIDID | MD5 |
| 0x70 | IDFA | 无 |
| 0x71 | IDFA | SHA1 |
| 0x72 | IDFA | MD5 |
| 0xFF | OTHER | 其他 |

## Carrier运营商

|  |  |
| --- | --- |
| 取值（int） | 描述 |
| 0 | 未知 |
| 1 | China Mobile |
| 2 | China Unicom |
| 3 | China Telecom |

## Connectiontype连接类型

|  |  |
| --- | --- |
| 取值（int） | 描述 |
| 0 | 未知 |
| 1 | Wifi网络 |
| 2 | 蜂窝数据网络-未知 |
| 3 | 蜂窝数据网络-2G |
| 4 | 蜂窝数据网络-3G |
| 5 | 蜂窝数据网络-4G |
| 6 | 蜂窝数据网络-5G |

## Devicetype设备类型

|  |  |
| --- | --- |
| 取值（int） | 描述 |
| 0 | 未知 |
| 1 | Phone |
| 2 | Tablet |

## Ctype点击类型

|  |  |
| --- | --- |
| 取值（int） | 描述 |
| 0 | 未知 |
| 1 | 打开网页 |
| 2 | 下载App |
| 3 | 到AppStore下载App（iOS Only） |
| 4 | 发送短信 |
| 5 | 拨打电话 |
| 6 | 用webview打开网页 |
| 7 | 打开为详情页（Android Only） |
| 8 | 应用内下载且支持触发url（Android Only） |

## Type广告创意类型

|  |  |
| --- | --- |
| 取值（int） | 描述 |
| 0 | 未知 |
| 1 | 文字链接广告 |
| 2 | 图片广告 |
| 3 | GIF广告 |
| 4 | HTML5广告 |
| 5 | MRAID v2.0广告 |
| 6 | 视频广告 |
| 7 | Flash广告 |
| 8 | 下载App |
| 9 | 信息流 |

## Ftype广告创意文件类型

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 取值（int） | 物料类型 | 对应文件类型 | MIME | 备注 |
| 0 | 未知 | 未知 | 未知 | 未知 |
| 0x11 | png | \*.png | image/png | PNG图片 |
| 0x12 | jpg | \*.jpg/\*.jpeg | image/jpeg | JPG或JPEG图片 |
| 0x13 | gif | \*.gif | image/gif | GIF图片 |
| 0x21 | flv | \*.flv | video/x-flv | FLV视频 |
| 0x22 | mp4 | \*.mp4 | video/mp4 | MP4视频 |

## Attr广告属性

|  |  |
| --- | --- |
| 取值（int） | 描述 |
| 0 | 未知 |
| 1 | Audio Ad(Auto-Play) |
| 2 | Audio Ad(User Initiated) |
| 3 | Expandable(Automatic) |
| 4 | Expandable(User Initiated-Click) |
| 5 | Expandable(User Initiated-Rollover) |
| 6 | In-Banner Video Ad(Auto-Play) |
| 7 | In-Banner Video Ad(User Initiated) |
| 8 | Pop(e.g., Over, Under, or Upon Exit) |
| 9 | Provocative or Suggestive Imagery |
| 10 | Shaky, Flashing, Flickering, Extreme Animation, Smileys |
| 11 | Surveys |
| 12 | Text Only |
| 13 | User Interactive(e.g., Embedded Games) |
| 14 | Windows Dialog or Alert Style |
| 15 | Has Audio On. Off Button |
| 16 | Ad Can be Skipped(e.g., Skip Button on Pre-Roll Video |

## OS 操作系统类型

|  |  |
| --- | --- |
| 取值（int） | 描述 |
| 0 | 未知 |
| 1 | iOS |
| 2 | Android |
| 3 | Windows |

## Video广告位类型

|  |  |
| --- | --- |
| 取值（int） | 描述 |
| 0 | 未知 |
| 1 | 页面广告 |
| 2 | 拉幕广告 |
| 3 | 前贴片广告 |
| 4 | 中贴片广告 |
| 5 | 后贴片广告 |
| 6 | 暂停广告 |
| 7 | 浮层广告 |

## Video格式类型

|  |  |
| --- | --- |
| 取值（int） | 描述 |
| 0 | 未知 |
| 33 | video/x-flv |
| 34 | video/mp4 |
| 35 | video/x-ms-wmv |
| 36 | application/x-shockwave-flash |

## 图片格式类型

|  |  |
| --- | --- |
| 取值（int） | 描述 |
| 0 | 未知 |
| 1 | Jpg |
| 2 | Jpeg |
| 3 | Png |
| 4 | Gif |

## Impression 扩展

### instl

|  |  |
| --- | --- |
| 取值（int） | 是否全插屏广告 |
| 0 | 不是 |
| 1 | 插屏 |
| 2 | 全屏 |

### splash

|  |  |
| --- | --- |
| 取值（int） | 是否开屏广告 |
| 0 | 不是 |
| 1 | 是 |

### acceptadtype

|  |  |
| --- | --- |
| 取值（int） | 描述 |
| 0 | 未知 |
| 1 | 品牌 |
| 2 | 效果\_电商 |
| 3 | 效果\_游戏 |
| 4 | 效果\_其他 |

## Make设备制作商类型

参考《Make设备制作商对应表》。

## adpos 广告位置

|  |  |
| --- | --- |
| 取值（int） | 描述 |
| 0 | 未知 |
| 1 | 第一屏 |
| 2 | 第二屏 |
| 3 | 其他屏 |
| 4 | 页面顶部 |
| 5 | 页面底部 |
| 6 | 侧边栏 |
| 7 | 全屏 |

## gender 用户性别

|  |  |
| --- | --- |
| 取值（int） | 性别 |
| 0 | 未知 |
| 1 | 女性 |
| 2 | 男性 |

## 第三方监测编号

|  |  |  |
| --- | --- | --- |
| 取值（int） | 描述 | 域名标识参考 |
| 0 | 未知 |  |
| 1 | 秒针 | [miaozhen.com](http://miaozhen.com/) |
| 2 | Admaster | [admaster.com.cn](http://admaster.com.cn/) |
| 3 | 尼尔森 | [cr-nielsen.com](http://cr-nielsen.com/) |

# 错误码

## 公共

|  |  |  |
| --- | --- | --- |
| **错误号** | **宏** | **含义** |
| 0x00 | SUCCESS | 成功 |
| 0x01 | UNKNOWN | 未知错误 |
| 0x02 | BAD\_REQUEST | 请求格式或内容错误 |
| 0x03 | INVALID\_PARAM | 参数错误 |
| 0x04 | ACCESS\_DENIED | 拒绝访问 |
| 0x05 | FCGI\_SERVICE\_ EXCEPTION | fcgi服务异常 |
| 0x06 | LOG\_OPEN\_FAIL | 打开日志文件失败 |
| 0x07 | LOG\_SERVER\_CONNECT\_FAIL | 日志服务器连接失败 |
| 0x08 | REDIS\_SERVER\_CONNECT\_FAIL | redis服务器连接失败 |
| 0x09 | REDIS\_CACHE\_EXCEPTION | redis缓存异常 |
| 0x0A | REDIS\_CONTENT\_INVALID | redis上下文失效 |
| 0x0B | DL\_OPEN\_FAIL | 动态库打开失败 |
| 0x0C | DL\_GET\_FUNCTION\_FAIL | 无法获取动态库导出函数 |
| 0x0D | NOT\_FOUND\_ REQUESTID | 找不到requestid |
| 0x0E | NOT\_FOUND\_DID\_DPID | did或dpid为空 |

## 竞价

|  |  |  |
| --- | --- | --- |
| **错误号** | **宏** | **含义** |
| 0x101 | MMDB\_OPEN\_FAIL | 打开mmdb失败 |
| 0x102 | NO\_SUITABLE\_CREATIVE | 没有合适的创意 |

## 结算/展现/点击

|  |  |  |
| --- | --- | --- |
| **错误号** | **宏** | **含义** |
| 0x201 | NOT\_FOUND\_MAPID | mapid为空 |
| 0x202 | REPEATED\_REQUEST | 重复的请求 |

# 开发备忘录

1.（2017年10月23日）“[单个活动的投放控制策略](#_单个活动的投放控制策略（HASH）)”和“[单个策略的投放控制策略](#_单个策略的投放控制策略（hash）)”中每日的投放控制信息：

由于研发进度的问题，目前（2017年10月23日）暂时存放该项目的所有日期的控制数据，而不是文档中所述当日和第二天的数据。