

Tanx ADX API

V4.2

安全密级(B2)

目录

| | | |
|-------|----------------------------------|----|
| 1 | 总体说明 | 1 |
| 1.1 | 概述 | 1 |
| 1.2 | 接入步骤 | 2 |
| 2 | 实时竞价 | 3 |
| 2.1 | 实时竞价接口 | 4 |
| 2.1.1 | 请求接口 | 4 |
| 2.1.2 | 应答接口 | 10 |
| 2.1.3 | 创意应答接口 | 13 |
| 2.2 | 宏配置说明 | 15 |
| 2.2.1 | 结算价格宏 | 15 |
| 2.2.2 | 点击宏 | 15 |
| 2.2.3 | 创意序列宏 | 19 |
| 2.3 | 信息处理 | 20 |
| 2.3.1 | 结算价格解析 | 20 |
| 2.3.2 | 点击加密参数计算 | 22 |
| 2.3.3 | 设备 ID 号解析 | 23 |
| 3 | Cookie Mapping | 25 |
| 3.1 | 概述 | 25 |
| 3.2 | 工作原理 | 26 |
| 3.2.1 | DSP 存储 Cookie 映射表 | 26 |
| 3.2.2 | Tanx 存储 Cookie 映射表（Beta 版） | 27 |
| 3.3 | 接口说明 | 29 |
| 4 | 信息同步 | 30 |
| 4.1 | 实时接口 | 30 |
| 4.1.1 | 创意信息同步 | 31 |
| 4.1.2 | 私有交易市场信息同步 | 41 |
| 4.1.3 | 客户资质信息同步 | 46 |
| 4.2 | 离线数据 | 59 |

| | |
|------------------------|----|
| 5 附加说明 | 61 |
| 5.1 字典文件 | 61 |
| 5.2 相关下载 | 62 |
| 5.3 优化建议 | 63 |
| 5.3.1 保持长连接 | 63 |
| 5.4 特别说明 | 64 |
| 5.4.1 移动设备上的横幅推广位..... | 64 |
| 5.4.2 Native..... | 64 |

1 总体说明

1.1 概述

Tanx ADX 是一套供 DSP(Demand Side Platform, 需求方平台) 使用的实时竞价营销平台。通过在该系统中的竞价, DSP 可以根据自身需求实时和自由的选择流量, 并统计投放情况。Tanx 目前只支持按 CPM 计费。

Tanx 和 DSP 之间的实时交互, 使用标准 HTTP 协议、protocol buffers 格式进行交互。您可以通过 <http://code.google.com/p/protobuf> 获取最新的 protocol buffers 技术支持。具体的 protocol buffers 的格式定义, 请查看 [tanx-bidding.proto](#)。

1.2 接入步骤

- 1) 商务洽谈，联系 Tanx 团队为您创建 Ad Exchange 的 DSP 帐户，获取 DSPID。
- 2) 使用 DSP 帐户，您可以获取 Tanx 团队的技术支持文档，帮助您完成接口的开发。
- 3) 请联系 Tanx 团队获取协议中涉及到的所有密钥数据，并妥善保管。
- 4) 向 Tanx 团队提供以下信息：
 - ✓ 可以访问的实时竞价 url 接口（例如：<http://www.dsp.com/bid>）
 - ✓ 竞价 url 接口可支持的最大 QPS（Query Per Second）
 - ✓ 可以接收 Cookie Mapping 重定向请求的接口（例如：<http://www.dsp.com/tid=>）
- 5) 当您的应用开发完成并通过内部测试后，请联系 Tanx 团队进行联调测试。在此期间 Tanx 会向您的竞价接口发送测试请求，请保持服务可用。
- 6) 对您的 DSP 帐户完成充值，并登录您的 DSP 帐户选择您所需要的流量类型。
- 7) 联系 Tanx 团队，开通 DSP 的相关竞价权限。
- 8) 至此，您的 DSP 帐户已完成和 Tanx 系统的对接。

2 实时竞价

Tanx 和 DSP 之间的实时交互方式如下：



1. Tanx 将请求发送给 DSP 提供的竞价接口。此请求会采用 `::Tanx::BidRequest` 协议格式封装请求数据，并使用标准的 HTTP POST 方式发送，其中 HTTP 的头部字段 Content-Type 会被设置为 `application/octet-stream`。
2. DSP 将竞价广告应答给 Tanx。此请求请采用 `::Tanx::BidResponse` 协议格式封装应答数据，并且在您的竞价广告中包含特定的宏，以便对点击/展现等的跟踪。有关宏的详细信息，请查看[宏说明部分](#)。

请注意：Tanx 会等待一定的时间（160 毫秒）接收 DSP 的应答。未在这段时间内成功应答（或应答格式错误）的 DSP，将不能参与此次竞价。当应答失败（包括格式错误）比率超过 15% 时，Tanx 将逐步降低向该 DSP 转发的流量，直至其比率小于 15%。为了减少时延，需要 DSP 在服务端开启连接保持（长连接），并且建议连接保持时间在 15 分钟以上。更多的优化建议，请查看[优化建议部分](#)。

3. 当页面广告展现时，会从媒体端直接发起 HTTP 请求，将此次竞价成功的结算价格反馈给 DSP。详细说明，请查看[结算价格宏部分](#)。

2.1 实时竞价接口

2.1.1 请求接口

即 [tanx-bidding.proto](#) 中定义的 BidRequest 消息。具体说明如下：

1) 协议基本信息

- ✓ **version**: 当前协议版本号，目前为 3。
- ✓ **bid**: Tanx 生成的唯一竞价 ID，32 个字符组成的字符串。
- ✓ **is_test**: 标记本次请求是否是测试请求。当为测试请求时，DSP 需要返回一个带有广告的应答，Tanx 不会将该应答广告展现给用户，也不会对该次广告展现计费。该标记适用于 Tanx 和 DSP 之间的联调测试。

is_ping: 标记本次请求是否是心跳检测请求。当为 ping 请求时，DSP 需要返回一个没有广告（即不设置 BidResponse.ads 字段）的应答。

2) 用户信息

- ✓ **tid**: 用户的身份识别码。如果 DSP 选择 Cookie 托管，且 hosted_match_data 字段存在，则该值不会被设置。详细说明，请查看 [Cookie Mapping 部分](#)。
- ✓ **ip**: 用户的 ip 地址。目前只支持 ipv4，点分十进制的字符串格式。例如：192.168.1.1
- ✓ **user_agent**: 用户的浏览器类型，来源于 HTTP 请求头部的 User-Agent。例如：
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)Chrome/33.0.1750.146 Safari/537.36
- ✓ **timezone_offset**: 用户所处时区的分钟偏移量。例如：如果是东八区，则 $timezone_offset = 60 * 8 = 480$ 。
- ✓ **user_vertical**: 用户垂直类别，预留字段，暂无实际值。
- ✓ **tid_version**: tid 字段的版本号。目前为 1。
- ✓ **hosted_match_data**: DSP 进行 Cookie 托管的用户标识 id。对应于 Cookie Mapping（托管模式）的 tanx_hm 参数。详细说明，请查看 [Cookie Mapping 部分](#)。
- ✓ **user_attribute**: DSP 进行 Cookie 托管的用户属性。对应于 Cookie Mapping（托管模式）的 tanx_ua 参数。详细说明，请查看 [Cookie Mapping 部分](#)。
- ✓ **UserAttribute.id**: DSP 进行 Cookie 托管的用户属性 id。详细说明，请查看 [Cookie Mapping](#)

部分。

- ✓ **UserAttribute.timestamp:** DSP 进行 Cookie 托管的用户属性 id 所对应的时间戳。详细说明，请查看 [Cookie Mapping 部分](#)。

3) 推广位通用信息

- ✓ **excluded_click_through_url:** 媒体禁止的目标跳转 url（广告点击跳转的最终页面），多值。基于网络传输成本的考虑，Tanx 只填充部分条目（保证填充的条目总长度不超过 200 个字符），建议 DSP 使用离线数据获取媒体设置的全量数据。详细离线数据的获取方式，请查看 [离线数据部分](#)。
- ✓ **url:** 推广位所在的页面 url。
- ✓ **category:** 推广位所在网站的分类。详细请查看 [字典文件部分](#)。
- ✓ **adx_type:** 流量所属来源，预留字段，目前为 0。
- ✓ **anonymous_id:** 匿名 id，预留字段，暂无实际值。
- ✓ **detected_language:** 预留字段，暂无实际值。
- ✓ **category_version:** 网站分类的版本号。目前为 1。
- ✓ **adzinfo:** 推广位基本信息。目前单次竞价仅支持一个推广位。
- ✓ **AdzInfo.id:** 推广位编号，目前固定为 0。和应答（BidResponse）里的 Ads.adzinfo_id 对应。该字段主要是为后续单次竞价中包含多推广位所预留。
- ✓ **AdzInfo.pid:** 推广位的唯一标识。此 pid 可以用于查询媒体推广位设置的过滤信息列表。详细请查看 [离线数据部分](#)。
- ✓ **AdzInfo.size:** 推广位的尺寸。格式：宽 x 高，例如：300x250。
- ✓ **AdzInfo.ad_bid_count:** 推广位希望从单个 DSP 获取的竞价广告数量。DSP 可以提供小于等于此值的广告个数（只会有一个广告胜出），多于 AdzInfo.ad_bid_count 数值的广告，将被截断而不参与竞价。
- ✓ **AdzInfo.view_type:** 推广位的展现形式，详细请查看 [字典文件部分](#)。

| 推广位类型 | | | 创意填充字段 | 封装格式 |
|------------|--------|------------------------------------|---------------|----------------|
| In-Browser | Banner | 固定/弹窗/悬停 | html_snippet | xhtml |
| | | 固定(移动网页) | | |
| | Video | 视频贴片 | video_snippet | VAST-linear |
| | | 视频暂停 | | VAST-nonlinear |
| In-App | Banner | 无线弹窗/无线横幅 (AdzInfo.api 为 mraid) | html_snippet | xhtml+mraid |

| | | | | |
|--|--------|--------------------------------|---|--------------------------|
| | | 无线弹窗/无线横幅 (AdzInfo. api 为空) | html_snippet | xhtml |
| | Native | 无线墙 | resource_address 或 mobilecreative | 参见 mobilecreative |
| | | 无线 Feeds 流 无线焦点图 | | |
| | Video | 无线视频贴片 | video_snippet | VAST-linear |
| | | 无线视频暂停 | | VAST-nonlinear+m raid |

- ✓ **AdzInfo.excluded_filter:** 媒体禁止的创意类型。多值，基于网络传输成本的考虑，Tanx 只填充部分条目，建议 DSP 使用离线数据获取媒体设置的全量数据。详细离线数据及获取方式，请查看[离线数据部分](#)。
- ✓ **AdzInfo.min_cpm_price:** 最低竞标价格，货币单位为人民币，数值含义为分/千次展现。如果 DSP 出价小于该值，则会被判定为格式错误而不参与竞价。
- ✓ **AdzInfo.adz_location:** 已经废弃，请忽略。
- ✓ **AdzInfo.view_screen:** 推广位在页面所在的屏数，有七个值：SCREEN_NA 代表无详细数据，SCREEN_OTHER 代表推广位在第六屏及以上，其余分别代表第一至第五屏。
屏数的定义为：显示器垂直分辨率为 768 时(例如分辨率 1024x768)，网页页面（最大化或全屏时）总共需要翻页的总页数。
- ✓ **page_session_id:** 标识一个页面 PV。对于同一个页面的单个 PV，Tanx 会生成一个相同且唯一的 page_session_id。DSP 可以通过本字段识别同一页面 PV 的广告请求。
- ✓ **AdzInfo.page_session_ad_idx:** 推广位在整个页面的渲染顺序。从 0 开始递增，可以和 page_session_id 配合使用。
注：一个页面可能会部署多个推广位，该页面的一次 PV 即会触发多个（每个推广位一个）广告请求。
- ✓ **AdzInfo. api:** 创意要支持的 apiFramework，详细请[查看字典部分](#)。
- ✓ **excluded_sensitive_category:** 禁止的敏感类目。多值，基于网络传输成本的考虑，Tanx 只填充部分条目，建议 DSP 使用离线数据获取媒体设置的全量数据。详细离线数据及获取方式，请查看[离线数据部分](#)。
- ✓ **excluded_ad_category:** 禁止的广告行业类目。多值，基于网络传输成本的考虑，Tanx 只填充部分条目，建议 DSP 使用离线数据获取媒体设置的全量数据。详细离线数据及获取方式，请查看[离线数据部分](#)。
- ✓ **content_categories:** 基于推广位所在页面内容的分类信息。

- ✓ **ContentCategory.id**: 基于推广位所在页面内容的网页类目 ID, 详细请查看[字典文件部分](#)。
- ✓ **ContentCategory.confidence_level**: 对应网页类目 ID 的置信分数。取值范围[0,1000]。

4) 移动设备扩展信息: 来自移动设备的广告请求

- ✓ **mobile**: 若本字段被设置, 则本次广告请求, 来自移动设备。
- ✓ **Mobile.is_app**: 标识该次广告请求是否来自 APP 应用。
- ✓ **Mobile.ad_num**: 需要展示的创意数量。仅对于 **AdzInfo.view_type** 为无线墙时, 本字段有效。详细请查看[移动设备上的广告 无线墙部分](#)。
- ✓ **Mobile.ad_keyword**: 关键词, 预留字段, 暂无实际值。
- ✓ **Mobile.is_fullscreen**: 推广位是否全屏展示。
- ✓ **Mobile.package_name**: 推广位所在的应用包名, 例如: **com.moji.MojiWeather**。
- ✓ **Mobile.device**: 移动设备信息。
- ✓ **Mobile.Device.platform**: 设备平台, 全为小写字符。例如: **android, iphone, ipad**。
- ✓ **Mobile.Device.brand**: 设备制造厂商, 全为小写字符。例如: **nokia, samsung**。
- ✓ **Mobile.Device.model**: 设备型号, 全为小写字符。例如: **n70, galaxy**。
- ✓ **Mobile.Device.os**: 设备操作系统, 全为小写字符。例如: **android, ios**。
- ✓ **Mobile.Device.os_version**: 设备操作系统版本。例如: **7.0.2**。
- ✓ **Mobile.Device.network**: 设备所处网络环境。该值为整型数值, 其值含义如下 (未来会对本字段的取值进行扩充):

- 0 ----- 未知
- 1 ----- wifi 环境
- 2 ----- 2G 网络
- 3 ----- 3G 网络
- 4 ----- 4G 网络

- ✓ **Mobile.Device.operator**: 设备的网络运营商。该值为整型数值, 其值含义如下 (未来会对本字段的取值进行扩充):

- 0 ----- 未知
- 1 ----- 中国移动
- 2 ----- 中国联通
- 3 ----- 中国电信

- ✓ **Mobile.Device.longitude:** 设备所在地理位置的经度（小数点格式）。例如：116.41667。
 - ✓ **Mobile.Device.latitude:** 设备所在地理位置的纬度（小数点格式）。例如：39.91667。
 - ✓ **Mobile.Device.device_size:** 设备的屏幕分辨率。格式：高 x 宽，例如：1024x768。
 - ✓ **Mobile.Device.device_pixel_ratio:** 用于高密度显示设备（如 iOS 的视网膜显示器）。默认值为 1000，非默认值表示当前的屏幕尺寸（以像素为单位）没有描述屏幕像素的实际数量。例如，宽度和高度是 640x320 的屏幕，实际上有 1080x640 像素，这时
`device_pixel_ratio_millis = 2000, device_size = 640x320。`
 - ✓ **Mobile.Device.device_id:** 设备 id 号。对于 iOS 设备，该值为系统的 mac(OS 版本 6 以下)/idfa 号(OS 版本 6 及以上)编码后的值；对于 android 设备，该值为系统的 imei 号编码后的值。编码规则，请查看[设备 ID 号解析部分](#)
 - ✓ **Mobile.native_template_id:** Native 创意可选的样式模板编号，详细请查看[Native 模板说明文档](#)
 - ✓ **Mobile.landing_type:** Native 可选的广告打开方式，详细请查看[字典文件部分](#)
- 5) 视频扩展信息：来自视频播放器的广告请求。视频分类参见 `content_categories` 字段。
- ✓ **Video:** 若本字段被设置，则本次广告请求，来自视频播放器
 - ✓ **Video.VideoFormat:** 预留字段，请暂时忽略。
 - ✓ **Video.Content:** 视频的描述信息
 - ✓ **Video.Content.title:** 视频标题，utf-8 编码
 - ✓ **Video.Content.duration:** 视频时长，单位秒
 - ✓ **Video.Content.keywords:** 视频关键字
 - ✓ **Video.videoad_start_delay:** 在视频中贴片资源的开始时间，0 表示前贴片，-1 表示后贴片，大于 0 的值表示中插，该数值代表具体起始时间，单位秒。
 - ✓ **Video.videoad_section_start_delay:** 贴片资源是一个或多个贴片的集合。该字段表示贴片在集合中的相对起始时间，单位毫秒
 - ✓ **Video.min_ad_duration:** 贴片最小播放时长，视频创意时长不可小于该值，单位毫秒
 - ✓ **Video.max_ad_duration:** 贴片最大播放时长，视频创意时长不可大于该值，单位毫秒
 - ✓ **Video.protocol:** 本期使用 VAST 3.0 版本，向下兼容
- 6) 私有交易市场的交易信息：包括优先交易和私有竞价两种交易模式。优先交易是定价不

保量的售卖模式，一般只指定唯一买方。私有竞价是只有媒体指定的 DSP 或广告主才能参与的竞价模式，价高者得，按照第二名报价加一分结算。

deals: DSP 与媒体签订的交易信息

deals.prefer_deal: 优先交易

deals.prefer_deal . dealid: 交易 ID

deals.prefer_deal . advertiser_ids: 允许的客户白名单，不设置则默认所有均允许。

deals.prefer_deal . fix_cpm_price: 优先交易的固定价格

deals.priv_auc: 私有竞价

deals.priv_auc . dealid: 交易 ID

deals.priv_auc . buyer_rules: 底价规则

deals.priv_auc . buyer_rules . advertiser_ids: 允许的客户白名单，不设置则默认所有均允许

deals.priv_auc . buyer_rules . min_cpm_price: 当前白名单中客户参与私有竞价的底价

2.1.2 应答接口

即 [tanx-bidding.proto](#) 中定义的 BidResponse 消息。具体说明如下：

1) 协议基本信息

- ✓ **version**: 当前协议版本号，目前为 3。
- ✓ **bid**: 唯一竞价 ID。请填写对应的 BidRequest 中的 bid 字段值。

2) 广告信息

- ✓ **ads**: DSP 返回的竞价广告信息。若 DSP 不对本次请求报价，则不要设置本字段。
- ✓ **Ads.adzinfo_id**: 标记该广告对应的推广位的编号，目前请填写 0。请填写对应的 BidRequest 中的 AdzInfo.id 字段值。
- ✓ **Ads.max_cpm_price**: 最高竞标价格，货币单位为人民币，数值含义为分/千次展现。该字段值不允许为 0，且不能低于推广位最低竞标价格（BidRequest.min_cpm_price），否则该广告将会被 Tanx 作为格式错误处理。
- ✓ **Ads.ad_bid_count_idx**: 当前广告在对应推广位上的下标值。该值严格从 0 递增，并且必须小于 BidRequest.AdzInfo.ad_bid_count。
- ✓ **Ads.html_snippet**: 广告代码片段。在移动设备上，创意需要对不同尺寸屏幕自适应。无线 banner 中 adzinfo.api 设置为 mraid 表示创意代码需要支持 mraid，详见 bidrequest.adzinfo.view_type 的字段说明。请参考[宏说明部分](#)在本字段中添加相应的宏。
- ✓ **Ads.click_through_url**: 广告的点击地址。本地址可以为 DSP（或第三方）的点击地址，比如 DSP 的点击跳转服务器地址，但必须保证能通过该地址到达正确的广告目标页面，即跳转后的最终地址必须和目标地址（Ads.destination_url 字段）一致。本字段必须以 http://或者 https://开头，并配合%%CLICK_URL%%点击宏使用。当 DSP 使用其他类型的点击宏时，不需要设置本字段。
- ✓ **Ads.category**: DSP 声明的本广告所属的广告行业类目和敏感类目。请参考[字典文件](#)的广告行业类目和敏感类目部分，正确填充本广告所属的类目 ID。当广告所声明的类目在媒体设置的过滤名单中时，该广告将被标记为过滤而不参与竞价。关于媒体的设置，请查看[离线数据部分](#)。
- ✓ **Ads.creative_type**: DSP 声明的广告创意类型。请参考[字典文件](#)的创意类型部分，正确

填充本广告的创意类型。当广告所声明的创意类型在媒体设置的过滤名单中时，该广告将被标记为过滤而不参与竞价。关于媒体的设置，请查看[离线数据部分](#)。

- ✓ **Ads.network_guid:** DSP 自定义数据。本字段即将废弃。要求：字段值长度不超过 32 位，且只能包含数字、字母和下划线("_")。
- ✓ **Ads.extend_data:** 本字段已经废弃。
- ✓ **Ads.destination_url:** 广告跳转的最终目标页面地址。必须填写，必须以 http://或者 https://开头，否则会被标记为格式错误而不能参与竞价。此字段和 Ads.click_through_url 不同：destination_url 必须是未经加密或跳转的最终目标地址，click_through_url 可以为中间跳转地址（比如 DSP 的点击服务器地址），但跳转后的最终地址必须和 destination_url 一致。
 - 1) 当广告只有一个最终目标页面地址时，请正确完整填写该地址。
 - 2) 当广告有多个最终目标页面地址（必须属于同一域名）时，请按完整域名进行合并处理。例如：

假设广告的点击地址有如下几个：

`http://item.taobao.com/item.htm?id=10693514300&k=237`

`http://item.taobao.com/item.htm?id=5688017331&cm_cat=50069524`

`http://item.taobao.com/item.htm?id=8297687951`

则本字段需要填写为：

`http://item.taobao.com`

- ✓ **Ads.creative_id:** 由 DSP 设置的广告的创意 ID。DSP 需要针对不同的创意分配对应的创意 ID，并保持创意 ID 始终关联到相同的创意。创意 ID 是 Tanx 审查创意的维度，故不要对相同的创意指定不同的创意 ID，或对不同的创意指定相同的创意 ID，都会影响创意的投放。要求：本字段值长度不能超过 64 位，且不能包含空格/回车/换行和其他不可见字符。Tanx 会对 DSP 的每个创意进行审核，DSP 可以通过实时 API 获取每个创意的审核状态，详细请查看[实时查询接口部分](#)。
- ✓ **Ads.resource_address:** 素材资源地址。本字段仅适用于移动设备上
 BidRequest.AdzInfo.view_type 是 Native 的竞价请求。如果 DSP 竞价胜出，该素材资源地址将被请求，DSP 需要以 MobileCreative 消息格式进行响应，返回真正展现的广告素材。
 详细请查看[移动设备上的广告 Native 部分](#)。
- ✓ **Ads.feedback_address:** 广告展示时的反馈地址。本字段仅适用于移动设备上的竞价请

求。只有本广告成功展现，该地址才会被请求。请参考[宏说明部分](#)，在本字段中添加相应的宏。

- ✓ **Ads.video_snippet:** 视频创意代码片段，遵循 VAST 协议，参见[视频创意格式的说明](#)。
- ✓ **Ads.mobile_creative:** DSP 可以使用本字段返回创意内容，不返回 Ads.resource_address。使用本字段不支持翻页。字段含义详见[创意应答接口的说明](#)。本字段仅适用于移动设备上 BidRequest.AdzInfo.view_type 是 Native 的竞价请求。
- ✓ **Ads.dealid:** 私有交易市场的交易 ID，需要和 bidrequest.deal.dealid 一致。不设置表示参与公开竞价。
- ✓ **Ads.advertiser_ids:** 客户 ID，不设置表示没有办法标识。获得客户 ID 的办法查看 4.1.3.1 新增客户接口的说明。
- ✓ **Ads.download_complete:** app 下载完成的反馈地址，适用于 Android 平台 APP 内 banner 且 AdzInfo.api 不是 mraid 的流量。
- ✓ **is_recognized_user:** 标记用户是否已完成 CookieMapping。适用于申请 Tanx 发起 CookieMapping 服务且非托管存储的 DSP。设置为 true 表示当 DSP 竞价失败时 Tanx 不发起 CookieMapping，设置为 false 表示当 DSP 竞价失败时 Tanx 按一定概率发起 CookieMapping。该字段默认值 false。

2.1.3 创意应答接口

该接口仅适用于移动设备上 `BidRequest.AdzInfo.view_type` 是 `Native` 的竞价请求，对应 [tanx-bidding.proto](#) 中定义的 `MobileCreative` 消息。具体说明如下：

1) 协议基本信息

- ✓ `version`: 当前协议版本号，目前为 3。
- ✓ `bid`: 竞价 ID，32 个字符组成的字符串。对应 `BidRequest.bid` 字段。请填写正确的 `bid` 值（在竞价过程中，DSP 可将 `bid` 以参数的形式包含在 `resouce_address` 参数中；当请求 `resouce_address` 地址时，即可从参数列表中获取到 `bid`）。

2) 创意信息

- ✓ `view_type`: 预留字段，请暂时忽略。
- ✓ `creatives`: 创意基本信息
- ✓ `Creative.img_url`: 图片地址，暂不支持 gif。
- ✓ `Creative.img_size`: 图片尺寸，格式为：宽 x 高，例如：200x200
- ✓ `Creative.title`: 创意说明，utf-8 编码，编码后长度不超过 50 个字符。
- ✓ `Creative.click_url`: 创意点击地址。
- ✓ `Creative.destination_url`: 创意的最终目标地址，必须以 `http://`或者 `https://`开头
- ✓ `Creative.attr`: 属性（描述）信息
- ✓ `Creative.Attr.name`: 属性名，详细请查看 [Native 模板说明文档](#) Mobile Creative ATTR 字段表
- ✓ `Creative.Attr.value`: 属性值
- ✓ `Creative.creative_id`: 由 DSP 设置的创意 ID。本字段值长度不能超过 64 位，且不能包含空格/回车/换行和其他不可见字符
- ✓ `Creative.category`: 预留字段，请暂时忽略。
 - ✓ `Creative.tracking_events`: 事件跟踪，当用户触发相应事件，Tanx 请求对应的地址
 - ✓ `Creative.tracking_events.impression_event`: 展现反馈地址
 - ✓ `Creative.tracking_events.click_event`: 点击反馈地址

- ✓ Creative.tracking_events.download_complete_event: 应用下载完成反馈地址
- ✓ native_template_id: DSP 从 bidrequest.Mobile.native_template_id 中选择的 Native 创意展示模板编号，详细请查看 [Native 模板说明文档](#)

2.2 宏配置说明

2.2.1 结算价格宏

原型：%%SETTLE_PRICE%%

含义：对于竞价成功的 DSP，可通过该宏获取竞价成功的广告的结算价格。必选。

展开：Tanx 会将所有出现的%%SETTLE_PRICE%%替换为加密后的结算价格。如何解密该结算价格，请查看[结算价格确认部分](#)

使用：

- 1) 该宏可出现在 BidResponse.Ads.html_snippet 字段中。例如：

```
<a href="%%CLICK_URL%%"></a>
```

```

```

注意：无线流量中 adzinfo.api 为非 mraid 的竞价返回，html_snippet 中的价格宏无效。

- 2) 该宏可出现在 BidResponse.Ads.resource_address 字段中。例如：

```
http://dsp.com/show?p=%%SETTLE_PRICE%%&bid=xxx
```

- 3) 该宏可出现在 BidResponse.Ads.feedback_address 字段中。例如：

```
http://dsp.com/feedback?p=%%SETTLE_PRICE%%&bid=xxx
```

以上 url，除必须带有%%SETTLE_PRICE%%宏外，其余部分（比如 bid 参数）可自行添加。

对于 BidResponse 里的其他任何字段出现该宏，Tanx 将不会做替换。

2.2.2 点击宏

原型：点击宏分为以下四种：

```
%%CLICK_URL%%
```

```
tanxclick=%%CLICK_URL%%
```

```
%%CLICK_URL_PRE_UNENC%%
```

```
%%CLICK_URL_PRE_ENC%%
```

含义：DSP 必须在竞价的广告中按照一定方式使用点击宏，以便 Tanx 跟踪广告的点击情况

展开：该宏会被 Tanx 替换为特定的点击字符串。

使用：

1) %%CLICK_URL%%

需要配合 BidResponse.click_through_url 字段使用。

只能出现在 html_snippet 字段中。

适用于点击地址只有一个，且在竞价时即可确定的场景。

DSP 需要将广告的点击地址单独写入 click_through_url 字段，并在 html_snippet 广告片段中，将点击地址替换为该宏。Tanx 会将 DSP 声明的点击地址 click_through_url 进行加密形成 Tanx 的点击地址，替换掉 html_snippet 中的该宏。例如：

DSP 声明的 html_snippet 为：

```
<a href="%%CLICK_URL%%"></a>
```

DSP 声明的 click_through_url 为：

```
http://dsp.com/clk?a=1&b=2
```

广告最终的展现如下（示例）：

```
<a href="http://click.tanx.com/xx?u=http%3A%2F%2Fdsp.com%2Fclk%3Fa%3D1%26b%3D2"></a>
```

2) tanxclick=%%CLICK_URL%%

[此宏即将废弃，请使用其他宏代替。](#)

需要配合 http://cdn.tanx.com/t/tanxclick.js 使用。

只能出现在 html_snippet 字段中。适用于 iframe 的广告

DSP 需要在 src 参数（iframe 的 src 元素）中包含“tanxclick=%%CLICK_URL%%”，并在 src 页面的最底部部署 tanxclick.js：<script src="http://cdn.tanx.com/t/tanxclick.js"></script>

例如：

DSP 声明的 html_snippet 为：

```
<iframe width=300 height=250 src="http://dsp.com/ad.php?a=1&tanxclick=%%CLICK_URL%%">
```

同时，http://dsp.com/ad.php 页面最底部必须部署如下 js

```
<script src="http://cdn.tanx.com/t/tanxclick.js"></script>
```

广告最终的展现如下（示例）：

```
<iframe width=300 height=250 src="http://dsp.com/ad.php?a=1&tanxdspv=http%3A%2F%2Fclick.tanx.com%2Fclk%3Fxx%3Dxx">
```

3) %%CLICK_URL_PRE_UNENC%%

只能出现在 `html_snippet` 字段和 `resource_address` 字段中。

适用于点击地址有一个或多个，且在竞价时即可确定的场景。

- ✓ 对于有固定跳转域名（比如 DSP 的跳转服务器）的 DSP，推荐 DSP 注册点击域名白名单，否则需要动态生成验证参数。请提供一级域名白名单列表，联系 Tanx 团队进行注册。宏使用方法如下：

DSP 需要将广告里所有的点击串进行 `urlencode`，并在其前面缀上该宏。例如：

原始广告为（其中 `dsp.com` 可能是 DSP 的跳转服务器地址）：

```
<a href="http://dsp.com/clk?a=1&b=2"></a>
```

则 DSP 需要注册点击域名白名单为：`dsp.com`（一级域名）

并将原始广告修改为：

```
<a href="%%CLICK_URL_PRE_UNENC%%http%3A%2F%2Fdsp.com%2Fclk%3Fa%3D1%26b%3D2"></a>
```

广告最终展现如下（示例）：

```
<a href="http://click.tanx.com/xx?tanx_u=http%3A%2F%2Fdsp.com%2Fclk%3Fa%3D1%26b%3D2"></a>
```

- ✓ 对于没有固定跳转域名的 DSP，需要配合点击密钥使用。点击密钥的获取，请联系 Tanx 团队。宏使用方法如下：

DSP 需要将广告里所有的点击串进行 `urlencode`，在其前面缀上该宏，并计算 `tanx_de` 和 `tanx_dk` 参数。关于如何计算 `tanx_de` 和 `tanx_dk`，请查看[点击加密参数计算部分](#)。

例如：

原始广告为（其中 `dsp.com` 可能是 DSP 的跳转服务器地址）：

```
<a href="http://dsp.com/clk?a=1&b=2"></a>
```

则 DSP 需要计算出 `tanx_de` 参数和 `tanx_dk` 参数：

```
tanx_de=fb90f14379352fef, tanx_dk=27
```

DSP 需要将原始广告修改为：

```
<a href="%%CLICK_URL_PRE_UNENC%%http%3A%2F%2Fdsp.com%2Fclk%3Fa%3D1%26b%3D2&tanx_de=fb90f14379352fef&tanx_dk=27"></a>
```

广告最终展现如下（示例）：

```
<a href="http://click.tanx.com/xx?tanx_u=http%3A%2F%2Fdsp.com%2Fclk%3Fa%3D1%26b%3D2&tanx_de=fb90f14379352fef&tanx_dk=27">
</a>
```

注意：对所有的广告，DSP 只能选择以上两种方式中的一种，不可混合使用。

4) %%CLICK_URL_PRE_ENC%%

只能出现在 `html_snippet` 字段中。

适用于点击地址有一个或多个，但在竞价时不能确定的场景。

类似%%CLICK_URL_PRE_UNENC%%宏，唯一的不同，就是%%CLICK_URL_PRE_ENC%%会进行 `urlencode`。例如：

DSP 声明的 `html_snippet` 为：

```
<iframe width=300 height=250 src="http://dsp.com/ad.php?a=1&tclick_pre=%%CLICK_URL_PRE_ENC%%">
```

广告最终的展现如下（示例）：

```
<iframe width=300 height=250 src="http://dsp.com/ad.php?a=1&tclick_pre=http%3A%2F%2Fclick.tanx.com%2Fxx%3Ftanx_u%3D%3A%2F%2Fclk.tanx.com%2Fclk%3Fa%3D1%26b%3D2&tanx_de=fb90f14379352fef&tanx_dk=27"></a></body> </html>
```

当请求 `ad.php` 页面时，其服务端可获取到 `tclick_pre` 参数（参数名可自定义），DSP 需要将原本的所有点击串进行 `urlencode`，并在其前面缀上（`urlencode` 后的）`tclick_pre` 参数。

如下是一个简单的 `ad.php` 页面示例（处理后的最终的页面展示）：

```
<html><head></head><body><a href="http://click.tanx.com/xx?tanx_u=http%3A%2F%2Fdsp.com%2Fclk%3Fa%3D1%26b%3D2&tanx_de=fb90f14379352fef&tanx_dk=27"></a></body> </html>
```

（点击宏须要注册点击域名白名单或点击秘钥，如果注册了点击域名白名单，示例中的 `tanx_de` 参数和 `tanx_dk` 参数则不需要填写。）

对于点击宏%%CLICK_URL_PRE_UNENC%%和%%CLICK_URL_PRE_ENC%%，DSP 在对原有的点击 `url` 进行处理时，可在其后附加其他自定义参数。Tanx 将在 302 重定向跳转时将这些参数附加在原有点击 `url` 的后面。例如：

处理后页面的点击串：`http://click.tanx.com/xx?tanx_u=http%3A%2F%2Fdsp.com%2Fclk%3`

Fa%3D1%26b%3D2&tanx_de=fb90f14379352fef&tanx_dk=27&id=123&k=456

则 Tanx 302 重定向跳转的 URL 为: <http://dsp.com/clk?a=1&b=2&id=123&k=456>

注意: 对于点击宏%%CLICK_URL_PRE_UNENC%%和%%CLICK_URL_PRE_ENC%%, DSP 的自定义的参数名, 不能以"tanx_"为前缀。所有以"tanx_"为前缀的参数名, 均为 Tanx 保留参数, Tanx 在 302 重定向跳转时, 均不会附加这些参数。

对于移动设备上的 APP 竞价请求 (Mobile.is_app 为 1), DSP 所返回的 BidResponse 可以不包含点击宏。其他竞价请求, DSP 所返回的 BidResponse, 必须包含上述宏中的一种, 且只能包含其中的一种, 不能在一个 BidResponse 中同时出现多种类型的宏。

宏可在规定的字段重复出现, Tanx 将按照约定的格式进行多次替换。

2.2.3 创意序列宏

该宏仅适用于 BidRequest.AdzInfo.view_type 是无线墙的竞价请求。

原型: %%RESOURCE_OFFSET%%

含义: 对于 BidRequest.AdzInfo.view_type 是无线墙的推广位, 其广告展现的素材地址由 BidResponse.Ads.resource_address 指定, 并且会根据用户的操作 (比如下拉刷新) 多次请求该地址。每一次请求, 都会将序列号和所需创意数告知 DSP, 以方便 DSP 组织数据进行应答。

DSP 可以通过该宏获取到上述信息。

展开: 在请求 resource_address 时, 该宏会被替换为"(起始序列号,所需创意数)"

使用:

只能且必须出现在 BidResponse.Ads.resource_address 字段中。例如:

DSP 声明的 resource_address 字段为:

http://dsp.com/ra?id=%%RESOURCE_OFFSET%%&xx=xx

则在请求该地址时, 对应的宏会被替换为:

[http://dsp.com/ra?id=\(0,10\)&xx=xx](http://dsp.com/ra?id=(0,10)&xx=xx)

表示本次请求, 起始序列号为 0, 需要 DSP 返回 10 个创意

如若再发生第二次请求, 则对应的宏会被替换为:

[http://dsp.com/ra?id=\(10,10\)&xx=xx](http://dsp.com/ra?id=(10,10)&xx=xx)

表示本次请求, 起始序列号为 10, 需要返回 10 个创意。

2.3 信息处理

2.3.1 结算价格解析

DSP 获取到的结算价格，是经过加密后的结算价格。需要配合密钥才能解密成功。关于如何获取加密的结算价格，请查看[结算价格宏部分](#)。

每个 DSP 有一个唯一的结算价格解密密钥，请联系 Tanx 团队获取，并妥善保管。

为方便说明，约定如下变量与操作：

P_{settle} : 真实结算价格

P_{encrypt} : 加密后的价格

K_{dsp} : DSP 的结算价格解密密钥

Version: 协议版本号

BidID: 竞价 id，对应于 BidRequest 里的 bid 字段

CRC: 校验码

+: 字符串拼接

^: 异或

H4(): 取前（高位）4 个字节

MD5(): md5 值计算

Base64Enc(): 标准 base64 编码 ([RFC2045](#))

Base64Dec(): 标准 base64 解码 ([RFC2045](#))

UrlEnc(): url 编码

UrlDec(): url 解码，需要将 '+' 转换为 ' '（空格）

E_{enc} : 加密后的密文， $E_{\text{enc}} = \text{UrlEnc}(\text{Base64Enc}(E_{\text{src}}))$

E_{src} : 原始密文， $E_{\text{src}} = \text{Base64Dec}(\text{UrlDec}(E_{\text{enc}}))$

原始密文 E_{src} 格式为：

| | | | |
|------------|-----------|--------------------------|--------|
| Version(1) | BidID(16) | P_{encrypt} (4) | CRC(4) |
|------------|-----------|--------------------------|--------|

Version: 1 字节的版本号 Version，目前值为 0x01。未来版本升级会告知 DSP。

BidID: 16 字节的 BidID，十六进制存储，对应于 BidRequest 的 bid 字段。

P_{encrypt}: 4 字节的密文价格 P_{encrypt}。

CRC: 4 字节的校验码 CRC。

解密方式（DSP 获取到的数据是 E_{enc}，即加密后的密文字符串）：

- 1) 转换编码，从 E_{enc} 获取 E_{src}: $E_{src} = \text{Base64Dec}(\text{UrlDec}(E_{enc}))$
- 2) 获取结算价格: $P_{settle} = P_{encrypt} \wedge H4(\text{MD5}(\text{BidID} + K_{dsp}))$
- 3) 计算校验码: $\text{CRC} = H4(\text{MD5}(\text{Version} + \text{BidID} + P_{settle} + K_{dsp}))$
- 4) 验证校验码: 上一步计算得到的 CRC 和 E_{src} 中的 CRC 是否一致。只有当该校验码验证通过，上述计算得到的结算价格 P_{settle} 才真实有效。

详细示例程序，请参考[结算价格解密示例程序（C++版）](#)

2.3.2 点击加密参数计算

当 DSP 使用%%CLICK_URL_PRE_UNENC%%或%%CLICK_URL_PRE_ENC%%点击宏时，如果 DSP 未在 Tanx 注册点击域名白名单，则需要对点击进行加密，即计算点击加密参数 tanx_de 和 tanx_dk。关于如何使用这些类型的点击宏，请查看[点击宏部分](#)。

每个 DSP 有一个唯一的点击加密密钥，请联系 Tanx 团队获取，并妥善保管。

为方便说明，约定如下变量与操作：

Version: 协议版本号，为 0x01。未来版本升级会告知 DSP

C_{orig}: 广告的原有点击地址，即 Tanx 302 重定向跳转的地址。

K_{dsp}: DSP 的点击加密密钥。

MD5(): md5 值计算

H16(): 取前（高位）16 个字节

+: 字符串拼接

^: 异或

Len(): 字符串长度

tanx_de 参数的计算: $H16(MD5(C_{orig} + K_{dsp}))$

tanx_dk 参数的计算: $Len(C_{orig}) \wedge Version$

举例如下：

假设

$C_{orig} = \text{http://dsp.com/clk?a=1\&b=2}$

$K_{dsp} = 123456789012345678901234567890ab$

$Version = 1$

则

$tanx_de = fb90f14379352fef$

$tanx_dk = 27$

2.3.3 设备 ID 号解析

DSP 获取到的设备 ID 号，是经过编码后的密文 ID。需要配合密钥才能解码成功。

每个 DSP 有一个唯一的解码密钥，请联系 Tanx 团队获取，并妥善保管。

为方便说明，约定如下变量与操作：

ID_{real} ：真实的设备 ID 号，对于 iOS 设备，可以是 mac/idfa，对于 android 设备，是 imei

$ID_{encrypt}$ ：编码后的设备 ID 号

K_{dsp} ：DSP 的解码密钥

Version：协议版本号

CRC：校验码

+: 字符串拼接

$H(A)(B)$ ：取 B 的前（高位）A 个字节

$MD5()$ ：md5 值计算

$Len()$ ：字符串长度

\wedge ：循环异或，比如 $A \wedge B$ 表示：当 $Len(B) \geq Len(A)$ ，则等价于 $A \wedge B$ ；当 $Len(B) < Len(A)$ ，则等价于 $A \wedge (H(Len(A))(B+B+B+...))$

$Base64Dec()$ ：标准 base64 解码（[RFC2045](#)）

$Base64Dec(device_id)$ 后的格式为：

| | | | |
|------------|-------------------|-----------------|--------|
| Version(1) | Len_ID_encrypt(1) | ID_encrypt(...) | CRC(4) |
|------------|-------------------|-----------------|--------|

Version：1 字节的版本号 Version，目前值为 0x01。未来版本升级会告知 DSP。

Len_ID_encrypt：1 字节的 $ID_{encrypt}$ 长度。

$ID_{encrypt}$ ：不定长的编码后的设备 ID 号，长度由 $Len_ID_{encrypt}$ 确定。

CRC：4 字节的校验码 CRC。

解码方式（DSP 获取到的数据是 $device_id$ ，即编码后的密文字符串）：

- 1) 转换编码： $Base64Dec(device_id)$
- 2) 获取设备 ID 的长度： $len = Len_ID_{encrypt}$
- 3) 获取真实设备 ID： $ID_{real} = H(len)(ID_{encrypt} \wedge MD5(K_{dsp}))$

- 4) 计算校验码: $CRC = H(4)(MD5(\text{Version} + \text{len} + ID_{\text{real}} + K_{\text{dsp}}))$
- 5) 验证校验码: 上一步计算得到的 CRC 和 device_id 中的 CRC 是否一致。只有当该校验码验证通过, 上述计算得到的 ID_{real} 才真实有效。

详细示例程序, 请参考[设备 ID 号解码示例程序 \(C++版\)](#)

3 Cookie Mapping

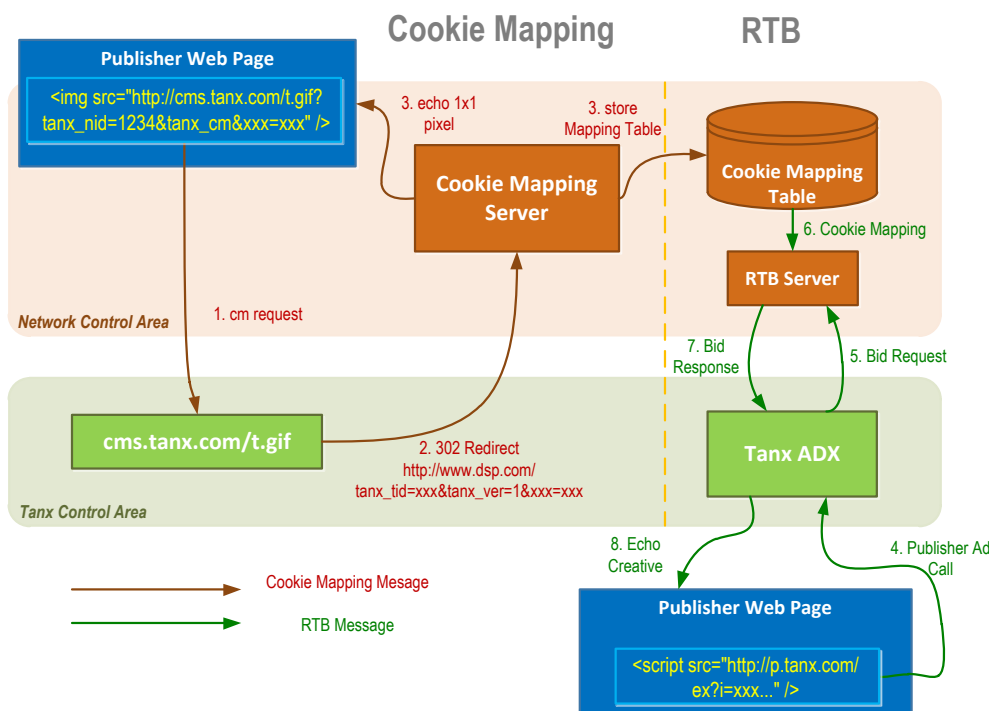
3.1 概述

- ✓ 由于浏览器限制跨域之间的 Cookie 访问，所以需要进行 Cookie Mapping 以关联 Tanx 域与各个 DSP 域下的 Cookie。
- ✓ Tanx Cookie 到 DSP Cookie 的映射关系叫做 Cookie 映射表。DSP 可以自行维护映射表，也可以委托 Tanx 托管存储（Beta 版）。需要进行托管存储的 DSP，请联系 Tanx 团队获取支持。
- ✓ 当用户浏览器没有 Tanx 域下的 Cookie 时，DSP 有权决定是否允许（默认是不允许）Tanx 在用户浏览器下种植 Tanx Cookie。如果需要开启该功能，请联系 Tanx 团队获取支持。
- ✓ Tanx 接收 Cookie Mapping 服务的地址为：<http://cms.tanx.com/t.gif?>
- ✓ 为了增加 Tanx 和 DSP 的 Cookie 匹配度，在实时竞价过程中，Tanx 会根据 DSP 的历史消耗和有效报价率，选中一个（非竞价胜出的）DSP(A)，在胜出 DSP(B) 的广告代码展现时，给该选中的 DSP(A) 一次发起 Cookie Mapping 的机会（通过页面执行 DSP 注册的 Cookie Mapping 代码片段）。该功能需要 DSP 提出申请，联系 Tanx 团队配合实施。
- ✓ 更多参数的说明，请查看[接口说明部分](#)

3.2 工作原理

3.2.1 DSP 存储 Cookie 映射表

当 DSP 存储 Cookie 映射表时（即非托管模式），Tanx 会以 HTTP 重定向参数的形式把 tanx_tid 告知 DSP，其 Cookie Mapping 过程和 Real-time Bidding 过程如下：



- 1) 浏览器加载 Cookie Mapping 代码片段：

```

```

其中，tanx_nid 为 DSP 的帐户 ID。tanx_cm 为必选参数。xxx 为 DSP 自定义参数，Tanx 会在 302 跳转时原样附加到跳转的 url 上。以 tanx_ 为前缀的参数名为保留参数名，DSP 自定义参数不能以 tanx_ 为前缀。

- 2) Tanx 接收到此请求后，会重定向到 DSP 的 Cookie Mapping 服务器，重定向 url 为：

```
http://www.dsp.com/tanx_tid=Yahe86sdfgeraH&tanx_ver=1&xxx=xxx
```

（该 url 地址是 DSP 提前在 Tanx 系统注册的 Cookie Mapping 地址，如需注册/更改，请联系 Tanx 团队获取支持）

tanx_tid 是 Tanx Cookie，tanx_ver 为当前 tanx_tid 的版本号。xxx 是 DSP 自定义参数

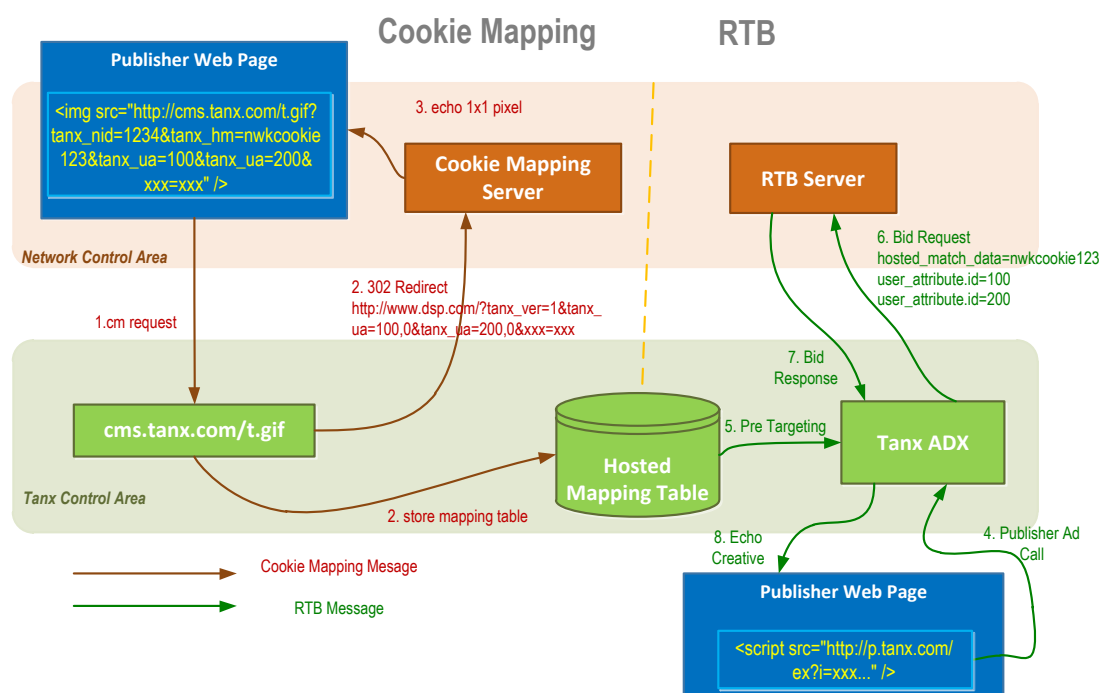
- 3) DSP 收到此重定向请求后，必须返回一个 1x1 大小的图片作为应答，同时需要在内部存储 tanx_tid 与 DSP Cookie 的 Cookie 映射表。

至此，Cookie Mapping 过程结束。

- 4) 页面发起一次广告请求到 Tanx
- 5) Tanx 向 DSP 发送带有 tanx_tid 的竞价请求，对应 BidRequest 消息里的 tid 字段
- 6) DSP 收到竞价请求后查询内部 Cookie Mapping Table，找到 tanx_tid 对应的 DSP Cookie
- 7) DSP 选择合适的广告向 Tanx 进行报价
- 8) Tanx 将胜出的 DSP 的广告展现到页面

3.2.2 Tanx 存储 Cookie 映射表（Beta 版）

当使用 Tanx 存储 Cookie 映射表时（即托管模式），DSP 需要将 DSP Cookie 以及其他数据，以 HTTP 参数的形式告知 Tanx，其 Cookie Mapping 过程和 Real-time Bidding 过程如下：



- 1) 浏览器执行 DSP 的 Cookie Mapping 代码请求到 Tanx 的 Cookie Mapping 服务器。其 URL 为：

`http://cms.tanx.com/t.gif?tanx_nid=1234&tanx_hm=nwkcookie123&tanx_ua=100&tanx_ua=200&xxx=xxx`

tanx_nid 为 DSP 的帐户 ID, tanx_hm 用于标识 DSP 给该用户的标识 ID, 即 DSP 的 Cookie。
tanx_ua 是 DSP 自定义的属性。目前 Tanx 在每个 DSP 的每个 tanx_hm 上最多支持 5 个 tanx_ua 的托管存储。xxx 为 DSP 自定义参数, Tanx 会在 302 跳转时原样附加到跳转的

url 上。

DSP 必须指定 tanx_hm 参数或者 tanx_ua 参数，或者两个同时指定。

- 2) Tanx 接收到 Cookie Mapping 请求，将重定向到 DSP 的 Cookie Mapping 服务器，反馈本次托管存储的结果。重定向 url 为：

http://www.dsp.com/?tanx_ver=1&tanx_hm=100,0&tanx_ua=200,0&xxx=xxx

（该 url 地址是 DSP 提前在 Tanx 系统注册的 Cookie Mapping 地址，如需注册/更改，请联系 Tanx 团队获取支持。）

xxx 是 DSP 自定义参数。

- 3) DSP 收到此重定向请求后，必须返回一个 1x1 大小的图片作为应答
- 至此，Cookie Mapping 过程结束。
- 4) 页面发起一次广告请求到 Tanx
- 5) Tanx 查询内部 Cookie Mapping Table，获取 DSP 托管存储的 tanx_hm 和 tanx_ua
- 6) Tanx 向 DSP 发送带有 tanx_hm 和 tanx_ua 的竞价请求，tanx_hm 对应 BidRequest 里的 hosted_match_data，tanx_ua 对应 BidRequest 里的 user_attribute。DSP 可以设置对流量的预筛选来选择合适的流量，例如：可以选择只让 Tanx 转发有匹配到 tanx_hm 的流量等。需要此功能的 DSP，请联系 Tanx 团队获取支持。
- 7) DSP 选择合适的广告向 Tanx 进行报价。
- 8) Tanx 将胜出的 DSP 的广告展现到页面。

3.3 接口说明

| | |
|----------|---|
| tanx_nid | <p>Tanx 的输入参数：发起 Cookie Mapping 的 DSP 账户 ID</p> <ul style="list-style-type: none"> 长度为 8 的字符串（数字） |
| tanx_hm | <p>Tanx 的输入参数：DSP 希望存储的用户标志 ID，即 DSP Cookie</p> <ul style="list-style-type: none"> 只能由“a-z”，“A-Z”，“0-9”，“+”，“/”，“=”组成 需要进行 urlencode 不超过 32 个字符 <p>Tanx 的输出参数：当托管存储 tanx_hm 数据出错时，设置本参数值为错误码。如果未出错，则不会设置本参数。错误码如下：</p> <ul style="list-style-type: none"> 1 格式错误 2 内部错误 |
| tanx_cm | <p>Tanx 的输入参数：回传 tanx_tid，当 DSP 需要 tanx_tid，请设置该参数</p> <ul style="list-style-type: none"> 为本参数设置的值会被忽略，并且可以省略，即(xx=xx&tanx_cm&xx=xx) 当该参数被设置，tanx_tid 会作为重定向参数传给 DSP |
| tanx_ua | <p>Tanx 的输入参数：DSP 委托 Tanx 存储的用户属性</p> <ul style="list-style-type: none"> 用以托管存储 DSP 设置的用户属性信息 格式 tanx_ua=用户属性 ID[, 时间戳]，例如： tanx_ua=10040021,1366350245，或者 tanx_ua=10040021 用户属性 ID 是不超过 8 位的数字 时间戳是 10 位的数字 可以在请求串中重复出现，用以标记一个用户的多个属性，但不多于 5 个 <p>Tanx 的输出参数：托管存储 tanx_ua 数据的返回码，每一个 tanx_ua 都有一个返回码</p> <ul style="list-style-type: none"> 格式：用户属性 ID，返回码。例如 tanx_ua = 10040021, 0 返回码： <ul style="list-style-type: none"> 0 成功 1 格式错误 2 内部错误 可重复出现 |
| tanx_tid | <p>Tanx 输出参数：Tanx 的用户标识 ID，即 Tanx Cookie</p> <ul style="list-style-type: none"> Tanx 在重定向时会对其进行 URL Encode，DSP 需要对其进行 URL Decode |
| tanx_ver | <p>Tanx 输出参数：tanx_tid 的版本</p> <ul style="list-style-type: none"> tanx_ver=1 |
| tanx_err | <p>Tanx 输出参数：本次 Cookie Mapping 的错误码</p> <ul style="list-style-type: none"> 1 没有 Tanx Cookie 且 DSP 不允许种植 Tanx Cookie 2 DSP 没有托管存储的权限 3 无效的指令（比如请求参数中仅有 tanx_nid） 4 内部错误 |

4 信息同步

4.1 实时接口

前期准备：

Tanx 所提供的实时查询接口，均需要使用淘宝开放平台的 `appkey`。DSP 需要在淘宝开放平台注册自己的 APP 应用，并联系 Tanx 团队，对该 `appkey` 进行访问的授权。步骤如下：

- 1) 进入淘宝开放平台，使用实名认证的淘宝帐户登录，填写基本信息提交后成为开放平台的开发者。
- 2) 创建一个应用，获取应用的 `appkey` 和 `appsecret`，并申请上线。
- 3) 联系 Tanx 团队，提交需要授权的 `appkey`（即上一步获取到的 `appkey`）。

具体操作步骤，请查看：[淘宝开放平台 开发入门](#)

当 `appkey` 得到授权后，DSP 需要编写调用代码，获取数据。API 调用方法，请查看：[API 调用方法详解](#)。示例代码，请参考[实时查询 API 示例程序——Java 版](#)。

请根据实际情况调整 API 调用周期，最高日调用次数不得超过 10 万次，每秒请求不得多于 50 次。

4.1.1 创意信息同步

4.1.1.1 创意预审接口

某些对创意要求较高的媒体，其推广位上所展示的广告，需要事先对创意进行审核，审核通过后才允许该广告进行展示。

该接口用于：对于某些创意，DSP 可在使用前，预先提供该创意供 Tanx 审核，审核通过后再使用。否则，DSP 将不能在创意要求较高的媒体上参与正常的广告竞价。该接口支持视频暂停和网页 banner 推广位的创意预审。

接口定义如下：

1) 接口名称：taobao.tanx.creative.add

2) 应用级输入参数：

| 名称 | 类型 | 可选 | 描述 | 示例值 |
|-------------------------|--------|----|--|---|
| member_id | Number | 必须 | DSP 的 memberId | 123456 |
| token | String | 必须 | 验证令牌，其值为：MD5 (tokenBaseKey + sign_time) | bf619ca70c8c084af019f5456495b752 |
| sign_time | Number | 必须 | 时间戳，1970-01-01 后的秒数 | 1382529851 |
| creative_id | String | 必须 | 创意 id | 34281 |
| adboard_type | String | 必须 | 广告类目 多个，以逗号分隔 | 62601,62602,62603,62604,62605,62606 |
| sensitive_type | String | 可选 | 敏感词类目，多个以逗号分隔 | 50010,50011 |
| adboard_data | String | 必须 | 创意代码 | pc 网页 banner:<iframe width=300 height=250 src='http://dsp.com/ad'></iframe> PC 视频暂停: <VAST version="3.0">.....</VAST> |
| destination_url | String | 必须 | 目标地址 | http://www.taobao.com |
| adboard_size | String | 必须 | 创意尺寸,宽 x 高 | 200x300 |
| creative_package_format | Number | 可选 | 创意封装类型: 1Htmlsnippet(pc 网页),2 vast-nonlinear (视频暂停), | 1 |

| 名称 | 类型 | 可选 | 描述 | 示例值 |
|----|----|----|------|-----|
| | | | 默认 1 | |

3) 返回结果:

| 名称 | 类型 | 是否必须 | 示例值 | 描述 |
|---------|---------|------|------|--------------|
| message | String | 是 | 调用成功 | 调用的成功信息或失败信息 |
| code | Number | 是 | 200 | 调用返回码 |
| is_ok | Boolean | 是 | true | 是否成功 |

4) 返回示例

XML 数据格式:

```
<?xml version="1.0" encoding="utf-8" ?>
<tanx_creative_add_response>
  <message>
    调用成功
  </message>
  <is_ok>
    true
  </is_ok>
  <code>
    200
  </code>
</tanx_creative_add_response>
```

JSON 数据格式:

```
{
  "tanx_creative_add_response": {
    "message": "调用成功",
    "is_ok": "true",
    "code": 200
  }
}
```

4.1.1.2 获取单个创意审核状态

该接口用于：使用创意 ID（BidResponse.Ads.creative_id）查询该创意的审核状态，可用于查询预审、先投后审和托管三类创意的审核状态。

接口定义如下：

1) 接口名称：taobao.tanx.creative.get

2) 应用级输入参数：

| 名称 | 类型 | 必须 | 描述 | 示例值 |
|-------------|--------|----|---|----------------------------------|
| member_id | Number | 必须 | DSP 的 memberId | 123456 |
| token | String | 必须 | 验证令牌，其值为：MD5 (tokenBaseKey + sign_time) | bf619ca70c8c084af019f5456495b752 |
| sign_time | Number | 必须 | 当前时间戳，1970-01-01 后的秒数 | 1382529851 |
| creative_id | String | 必须 | 创意 ID | 12345678 |

3) 返回结果

| 名称 | 类型 | 是否必须 | 示例值 | 描述 |
|---------|-------------|------|------|--------------|
| result | CreativeDto | 是 | | 创意查询的返回结果 |
| message | String | 是 | 调用成功 | 调用的成功信息或失败信息 |
| code | Number | 是 | 200 | 调用返回码 |
| is_ok | Boolean | 是 | true | 是否成功 |

其中，CreativeDto 类型定义如下：

| 名称 | 类型 | 示例值 | 描述 |
|-------------|--------|----------|--|
| creative_id | String | 12345678 | 创意 ID |
| status | String | PASS | 创意审核的状态（通过 PASS,拒绝 REFUSE,未审核 WAITING） |
| level | Number | 1 | 创意通过的等级，1 表示一级创意，99 表示普通创意 |

| 名称 | 类型 | 示例值 | 描述 |
|--------------------|-----------------------------|--------|--------------|
| refuse_cause | String | 创意无法显示 | 创意拒绝的原因 |
| publishersAuditDto | CreativePublisherAuditDto[] | | 创意在拒登媒体的审核信息 |

其中，CreativePublisherAuditDto 类型定义如下：

| 名称 | 类型 | 示例值 | 描述 |
|--------------|--------|----------|--|
| publisher_id | Number | 12345678 | 媒体的 memberId,pid:mm_xxx_xxx_xxx, mm 后面的第一串数字 |
| status | String | WAITING | 审核状态（拒绝 REFUSE,未审核 WAITING） |
| refuse_cause | String | 创意无法显示 | 审核状态为 REFUSE 时的拒绝原因 |

4) 返回示例

XML 数据格式：

```
<?xml version="1.0" encoding="utf-8" ?>
<tanx_creative_get_response>
  <message>
    调用成功
  </message>
  <result>
    <creative_id>
      12345678
    </creative_id>
    <level>
      1
    </level>
    <status>
      PASS
    </status>
    <refuse_cause>
    </refuse_cause>
    < publishers_audit_dto >
      <publisher_id>
        12345678
      </ publisher_id>
```

```

        <status>
            REFUSE
        </status>
        <refuse_cause>
            创意无法显示
        </refuse_cause>
    </publishers_audit_dto >

</result>
<is_ok>
    true
</is_ok>
<code>
    200
</code>
</tanx_creative_get_response>

```

JSON 数据格式:

```

{
  "tanx_creative_get_response": {
    "message": "调用成功",
    "result": {
      "creative_id": "12345678",
      "level": 1,
      "status": "PASS",
      "refuse_cause": "",
      "publishers_audit_dto ": [{
        "publisher_id": "12345678",
        "status": "REFUSE",
        "refuse_cause": "创意无法显示"
      }]
    },
    "is_ok": "true",
    "code": 200
  }
}

```

4.1.1.3 批量获取创意审核状态

该接口用于：批量查询创意的审核状态。创意审核状态会分页显示，每页 1000 条，按创建时间的升序排列。DSP 可以通过指定不同分页来获取全部投放过的创意。

接口定义如下：

1) 接口名称：taobao.tanx.creatives.get

2) 应用级输入参数：

| 名称 | 类型 | 必须 | 描述 | 示例值 |
|-----------|--------|----|---|----------------------------------|
| member_id | Number | 必须 | DSP 的 memberId | 123456 |
| token | String | 必须 | 验证令牌，其值为：MD5 (tokenBaseKey + sign_time) | bf619ca70c8c084af019f5456495b752 |
| sign_time | Number | 必须 | 当前时间戳，1970-01-01 后的秒数 | 1382529851 |
| status | String | 必须 | 创意的状态（全部 ALL,通过 PASS,拒绝 REFUSE,未审核 WAITING） | REFUSE |
| page | Number | 必须 | 分页的页码(第一页为 1) | 2 |

3) 返回结果

| 名称 | 类型 | 是否必须 | 示例值 | 描述 |
|---------|----------------|------|------|--------------|
| results | CreativeDto [] | 是 | | 创意列表查询的返回结果 |
| message | String | 是 | 调用成功 | 调用的成功信息或失败信息 |
| code | Number | 是 | 0 | 调用返回码 |
| is_ok | Boolean | 是 | true | 调用是否成功 |

其中，CreativeDto 类型定义如下：

| 名称 | 类型 | 示例值 | 描述 |
|-------------|--------|----------|--|
| creative_id | String | 12345678 | 创意 ID |
| status | String | PASS | 创意审核的状态（通过 PASS,拒绝 REFUSE,未审核 WAITING） |

| 名称 | 类型 | 示例值 | 描述 |
|--------------------|-----------------------------|--------|----------------------------|
| level | Number | 1 | 创意通过的等级，1 表示一级创意，99 表示普通创意 |
| refuse_cause | String | 创意无法显示 | 创意拒绝的原因 |
| publishersAuditDto | CreativePublisherAuditDto[] | | 创意在拒登媒体的审核信息 |

其中，CreativePublisherAuditDto 类型定义如下：

| 名称 | 类型 | 示例值 | 描述 |
|--------------|--------|----------|--|
| publisher_id | Number | 12345678 | 媒体的 memberId,pid:mm_xxx_xxx_xxx, mm 后面的第一串数字 |
| status | String | WAITING | 审核状态（拒绝 REFUSE,未审核 WAITING） |
| refuse_cause | String | 创意无法显示 | 审核状态为 REFUSE 时的拒绝原因 |

返回示例

XML 数据格式：

```
<?xml version="1.0" encoding="utf-8" ?>
<tanx_creatives_get_response>
  <message>
    调用成功
  </message>
  <results>
    <creative_dto>
      <creative_id>
        12345678
      </creative_id>
      <level>
        1
      </level>
      <status>
        PASS
      </status>
      <refuse_cause>
        创意无法显示
      </refuse_cause>
      < publishers_audit_dto >
```



```

        <publisher_id>
            12345678
        </ publisher_id>
        <status>
            PASS
        </status>
        <refuse_cause>
            创意无法显示
        </refuse_cause>
    </publishers_audit_dto >
</creative_dto>
</results>
<is_ok>
    true
</is_ok>
<code>
    0
</code>
</tanx_creatives_get_response>

```

JSON 数据格式:

```

{
  "tanx_creatives_get_response": {
    "message": "调用成功",
    "results": {
      "creative_dto": [{
        "creative_id": "12345678",
        "level": 1,
        "status": "PASS",
        "refuse_cause": "创意无法显示",
        " publishers_audit_dto ": [{
          "publisher_id": "12345678",
          "status": "PASS",
          "refuse_cause": "创意无法显示"
        }]
      }]
    },
    "is_ok": "true",
    "code": 0
  }
}

```

4.1.1.4 创意托管上传接口

视频贴片推广位上投放的创意需要在投放前托管在 Tanx 服务器，Tanx 对托管的创意进行审核，DSP 可使用创意状态查询接口（`taobao.tanx.creative.get` 或 `taobao.tanx.creatives.get`）获取审核状态，投放时使用托管的创意。

接口定义如下：

1) 接口名称：`taobao.tanx.depositcreative.add`

2) 应用级输入参数：

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|----------------------|--------|------|--|--------------------------------------|
| member_id | Number | 必须 | DSP 的 memberId | 123456 |
| token | String | 必须 | 验证令牌，其值为：MD5 (tokenBaseKey + sign_time) | bf619ca70c8c084af0 19f5456495b752 |
| sign_time | Number | 必须 | 当前时间戳，1970-01-01 后的秒数 | 1382529851 |
| creative_id | String | 必须 | dsp 系统中的创意 id | 1234567 |
| creative_size | String | 可选 | 创意尺寸，长高中间用小写英文 x | 200x300 |
| creative_category_id | String | 必须 | 创意的类目，多个值用逗号，分隔 | 72205,71606 |
| creative_format | String | 必须 | 创意类型 1:文字 2:图片 3:Flash 4:视频贴片 | 4 |
| creative_name | String | 可选 | 托管创意的名称 | aaa |
| api_framework | String | 可选 | 创意支持的 apiFramework 协议 ,1:VPAID1.0;2:VPAID2.0;3:MA RID-1;4:ORMMA;5:1MRAID-2, 只 能单选，不能多选 | 1 |
| file_type | String | 必须 | 文件格式，视频类型：flv、avi、 mp4，图片类型：jpg、png、gif | mp4 |
| creative_url | String | 必须 | 创意存储地址 | http://www.tanx.co m/aaa.mp4 |
| click_url | String | 必须 | 创意点击地址 | http://www.tanx.co m |
| duration | Number | 特殊可选 | 创意时长，单位是毫秒，允许±0.5 秒误差。 | 100 |

3) 返回结果:

| 名称 | 类型 | 是否必须 | 示例值 | 描述 |
|---------|---------|------|------|--------------|
| message | String | 是 | 调用成功 | 调用的成功信息或失败信息 |
| code | Number | 是 | 200 | 调用返回码 |
| is_ok | Boolean | 是 | true | 是否成功 |

4) 返回示例

XML 数据格式:

```
<?xml version="1.0" encoding="utf-8" ?>
<tanx_depositcreative_add_response>
  <message>
    调用成功
  </message>
  <is_ok>
    true
  </is_ok>
  <code>
    200
  </code>
</tanx_creative_add_response>
```

JSON 数据格式:

```
{
  "tanx_depositcreative_add_response": {
    "message": "调用成功",
    "is_ok": "true",
    "code": 200
  }
}
```

4.1.2 私有交易市场信息同步

4.1.2.1 按交易 id 查询交易信息

该接口用于：使用交易 ID 查询该交易的相关信息。

接口定义如下：

1) 接口名称：taobao.tanx.deal.get

2) 应用级输入参数：

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|-----------|--------|------|------------------|------|
| dsp_id | Number | 必须 | dsp 用户 id | 1 |
| deal_id | Number | 必须 | 交易 id | 1 |
| sign_time | Number | 必须 | 1970 年到现在的的时间，毫秒 | 111 |
| token | String | 必须 | 验证 token | afad |

3) 返回结果：

| 名称 | 类型 | 是否必须 | 示例值 | 描述 |
|---------|-------------|------|------|------|
| code | Number | 否 | 1 | 结果代码 |
| messag | String | 否 | OK | 结果信息 |
| Success | Boolean | 否 | true | 查询结果 |
| result | DealInfoDTO | 否 | | 查询结果 |

其中，DealInfoDTO 类型定义如下：

| 名称 | 类型 | 示例值 | 描述 |
|------------|-----------|------------|---------------------------|
| addresses | DicDTO [] | | 交易对应的地域定向信息 |
| deal_id | Number | 28 | 交易对象 id |
| price | Number | 52 | 优先交易的固定价格或者私有竞价的最低价格，单位是分 |
| begin_time | Date | 2000-01-01 | 交易开始时间 |

| | | | |
|-------------------|-----------|------------------------|--------------------|
| | | 00:00:00 | |
| end_time | Date | 2000-01-01 00:00:00 | 交易结束时间 |
| adzone_ids | Number [] | 21046113 | 交易对应的广告位 |
| deal_type | Number | 1 | 交易类型：1 优先采购 2 私有交换 |
| seller_site_names | String [] | www.taobao.com | 交易买方对应的网站列表 |
| advertiser_ids | Number [] | 44 | 交易相关的广告客户 ID |
| interval_ids | String [] | 1 | 交易对应的时间段定向信息 |

其中，DicDTO 类型定义如下：

| 名称 | 类型 | 示例值 | 描述 |
|-------|--------|-----|--------|
| code | Number | 153 | 数据项 ID |
| value | String | abc | 数据项值 |

4) 返回示例

JSON 数据格式：

```
{
  "tanx_deal_get_response": {
    "code": 1,
    "messag": "OK",
    "sucess": true,
    "result": {
      "addresses": {
        "dic_d_t_o": [{
          "code": 153,
          "value": "abc"
        }]
      },
      "deal_id": 28,
      "price": 52,
      "begin_time": "2000-01-01 00:00:00",
      "end_time": "2000-01-01 00:00:00",
      "adzone_ids": {
        "number": [21046113]
      },
      "deal_type": 1,
      "seller_site_names": {
        "string": ["www.taobao.com"]
      }
    }
  }
}
```

```

    },
    "dsp_ids": {
      "number": [17825897]
    },
    "advertiser_ids": {
      "number": [44]
    },
    "interval_ids": {
      "string": ["1"]
    }
  }
}
}

```

4.1.2.2 批量获取交易信息

该接口用于：按交易类型批量查询交易的信息。交易信息会分页显示,DSP 可以通过指定不同分页来获取全部符合条件的交易信息。

接口定义如下：

接口名称：taobao.tanx.deals.get

1) 应用级输入参数：

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|-----------|--------|------|--------------------|-----------------|
| dsp_id | Number | 必须 | dsp 用户 id | 10891958 |
| token | String | 必须 | dsp 用户验证 token | fdsfdsf345456s |
| page_size | Number | 必须 | 页大小 | 10 |
| deal_type | Number | 必须 | 交易类型：1 优先采购 2 私有交换 | 1 |
| page | Number | 必须 | 页码 | 1 |
| sign_time | Number | 必须 | 1970 年到现在的时间，毫秒 | 173000343445532 |

2) 返回结果：

| 名称 | 类型 | 是否必须 | 示例值 | 描述 |
|---------|----------------|------|------|--------|
| sucess | Boolean | 否 | true | 查询是否成功 |
| code | Number | 否 | 1 | 查询结果编码 |
| message | String | 否 | OK | 查询结果信息 |
| deals | DealInfoDTO [] | 否 | | 查询交易结果 |

其中，DealInfoDTO 类型定义如下：

| 名称 | 类型 | 示例值 | 描述 |
|-------------------|-----------|---------------------|---------------------------|
| addresses | DicDTO [] | | 交易对应的地域定向信息 |
| deal_id | Number | 28 | 交易对象 id |
| price | Number | 52 | 优先交易的固定价格或者私有竞价的最低价格，单位是分 |
| begin_time | Date | 2000-01-01 00:00:00 | 交易开始时间 |
| end_time | Date | 2000-01-01 00:00:00 | 交易结束时间 |
| adzone_ids | Number [] | 21046113 | 交易对应的广告位 |
| deal_type | Number | 1 | 交易类型：1 优先采购 2 私有交换 |
| seller_site_names | String [] | www.taobao.com | 交易买方对应的网站列表 |
| dsp_ids | Number [] | 17825897 | 交易相关的 dsp ID 列表 |
| advertiser_ids | Number [] | 44 | 交易相关的广告客户 ID |
| interval_ids | String [] | 1 | 交易对应的时间段定向信息 |

其中，DicDTO 类型定义如下：

| 名称 | 类型 | 示例值 | 描述 |
|-------|--------|-----|--------|
| code | Number | 153 | 数据项 ID |
| value | String | abc | 数据项值 |

3) 返回示例

JSON 数据格式：

```
{
  "tanx_deals_get_response": {
    "sucess": true,
    "code": 1,
    "message": "OK",
    "deals": {
      "deal_info_d_t_o": [{
        "addresses": {
          "dic_d_t_o": [{
            "code": 153,
            "value": "abc"
          }]
        }
      }]
    }
  }
}
```

```

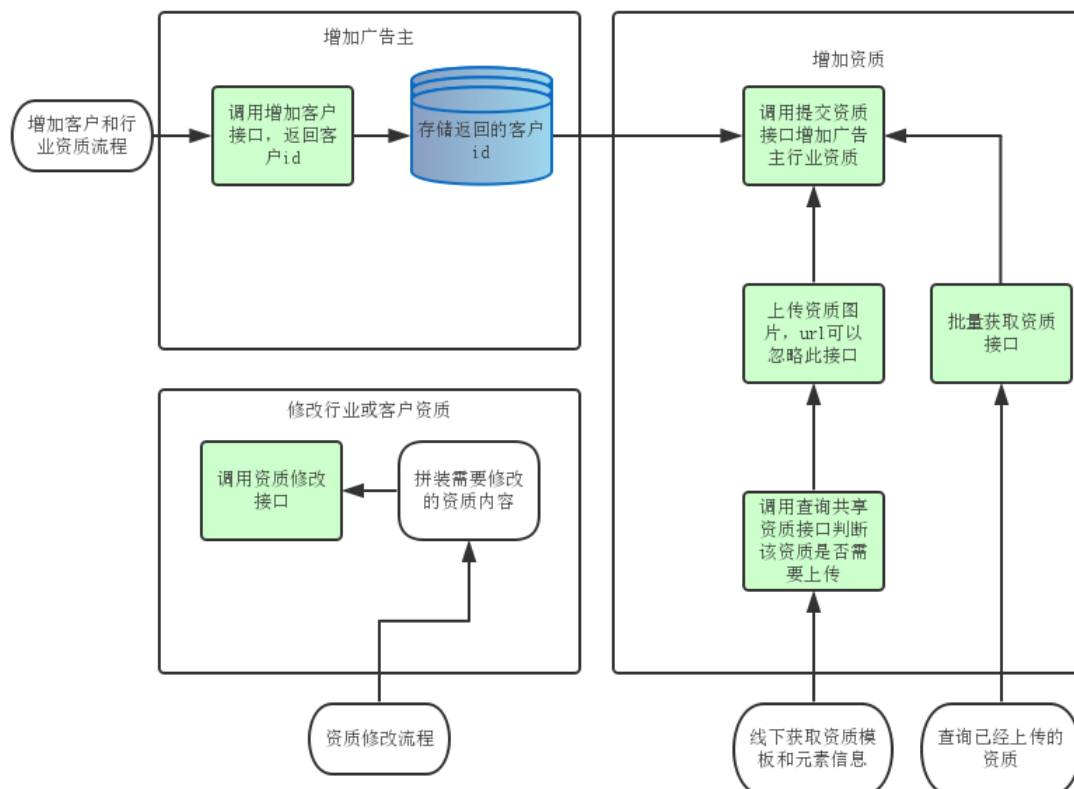
    },
    "deal_id": 28,
    "price": 52,
    "begin_time": "2000-01-01 00:00:00",
    "end_time": "2000-01-01 00:00:00",
    "adzone_ids": {
        "number": [21046113]
    },
    "deal_type": 1,
    "seller_site_names": {
        "string": ["www.taobao.com"]
    },
    "dsp_ids": {
        "number": [17825897]
    },
    "advertiser_ids": {
        "number": [44]
    },
    "interval_ids": {
        "string": ["1"]
    }
  }
}
}
}

```


4.1.3 客户资质信息同步

DSP 可以使用界面提交和接口提交两种方法上传客户资质信息，接口上传方法：

- 1) 增加客户：调用[新增客户接口](#)注册客户，调用成功返回客户 ID，DSP 存储客户 ID 在本地，在提交资质时使用。
- 2) 提交行业资质：客户根据离线的资质规则说明调用[提交资质接口](#)上传资质至 Tanx。特别的，在上传前可调用[查询共享资质接口](#)判断哪些资质不用上传。



4.1.3.1 新增客户接口

该接口用于：DSP 使用本接口新增客户到 Tanx，Tanx 将返回客户 ID 给 DSP。如果客户信息已提交，将返回已存在的客户 ID。

接口定义如下：

- 1) 接口名称：taobao.tanx.qualification.advertiser.add
- 2) 应用级输入参数：

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|----|----|------|----|-----|
|----|----|------|----|-----|

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|-------------|-----------------|------|---|------------|
| advertisers | AdvertiserDTO[] | 必须 | 名为 advertiser 的 json 对象数组, 一次最多上传 20 个 | |
| member_id | Number | 必须 | DSP 的 memberId | 123456 |
| token | String | 必须 | 验证令牌, 其值为: MD5 (tokenBaseKey + sign_time) | |
| sign_time | Number | 必须 | 当前时间戳, 1970-01-01 后的秒数 | 1382529851 |

其中, AdvertiserDTO 类型定义如下:

| 名称 | 类型 | 必须 | 描述 | 示例值 |
|-----------------|--------|----|--|---------|
| user_type | Number | 必须 | 用户类型(客户类型 1-淘系客户 2-非淘系公司, 3-非淘系个人) | 1 |
| advertiser_name | String | 必须 | 客户名称。 淘系客户填写推广店铺的旺旺昵称 非淘系公司客户填写推广域名所有人的营业执照公司名称 非淘系个人填写网站名称 | 耐克官方旗舰店 |

3) 返回结果:

| 名称 | 类型 | 示例值 | 描述 |
|-----------------|-----------------|-----|--------|
| is_success | Boolean | | 是否成功 |
| advertiser_list | AdvertiserDto[] | | 客户对象列表 |

其中, Advertiser 类型定义如下:

| 名称 | 类型 | 示例值 | 描述 |
|-----------------|--------|------|--|
| nick_name | String | 上海宝洁 | 客户昵称, 由 TANX 写入的昵称 |
| advertiser_type | Number | 1 | 用户标签(用二进制存储客户属性 1.品牌客户 2. VIP 客户 4. 世界 500 强客户)。此标签是由 TANX 进行的归类 |
| user_type | Number | 1 | 用户类型(客户类型 0-淘宝, 1-天猫, 2-dsp 公司, 3-dsp 个人)。填写的淘系客户在返回时 TANX 会返回具体是属于淘系的天猫还是淘宝 |

| 名称 | 类型 | 示例值 | 描述 |
|-----------------|--------|---------|---|
| advertiser_name | String | 宝洁天猫旗舰店 | 客户的名称 |
| advertiser_id | Number | 123 | 客户 id。之后 DSP 与 TANX 需要用到客户 ID 交互的地方都用此 ID |

4) 返回示例

JSON 数据格式：

```
{
  "tanx_qualification_advertiser_add_response": {
    "is_success": true,
    "advertiser_list": {
      "advertiser_dto": [
        {
          "nick_name": "上海宝洁",
          "advertiser_type": 1,
          "user_type": 1,
          "advertiser_name": "宝洁天猫旗舰店",
          "advertiser_id": 2755
        }
      ]
    }
  }
}
```

4.1.3.2 查询共享资质接口

该接口用于：本接口返回客户的所有已经存在的资质，这些资质在过期之前不需要再次上传。

接口定义如下：

1) 接口名称：taobao.tanx.qualification.solid.find

2) 应用级输入参数：

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|---------------|----------|------|------------------------------------|-------|
| advertiser_id | Number | 必须 | 客户的 id，单个查询 | 123 |
| element_ids | Number[] | 可选 | 客户资质元素的 id，通过资质元素 id 指定查询的资质元素下的共享 | [1,2] |

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|-----------|--------|------|--|------------|
| | | | 资质，一次最多指定 100 个 | |
| page | Number | 必须 | 起始页 | 1 |
| page_size | Number | 必须 | 分页大小，一次最大 100 条 | 20 |
| member_id | Number | 必须 | DSP 的 memberId | 123456 |
| token | String | 必须 | 验证令牌，其值为：MD5 (tokenBaseKey + sign_time) | |
| sign_time | Number | 必须 | 当前时间戳，1970-01-01 后的秒数 | 1382529851 |

3) 返回结果：

| 名称 | 类型 | 示例值 | 描述 |
|--------------------|---------------------|-----------------|--------|
| is_success | Boolean | | 是否成功 |
| qualification_list | qualificationDTO [] | Tanx 的客户的 id 列表 | |
| count | String | 200 | 返回查询总数 |

其中 qualificationDTO 返回结果为

| 名称 | 类型 | 示例值 | 描述 |
|------------|--------|------------|--|
| name | String | 上海宝洁营业执照资质 | 上次资质时 qualification 中的 name。具体见下面上传资质章节的定义 |
| user_id | Number | 123 | 客户 id。对应前面的 advertiser_id |
| user_name | String | 上海宝洁 | 客户名称。对应前面的 advertiser_name |
| user_type | String | 宝洁天猫旗舰店 | 用户类型(客户类型 0-淘宝，1-天猫，2-dsp 公司，3-dsp 个人) |
| element_id | Number | 123 | 资质元素 id |
| start_time | Date | 123 | 资质生效时间 |
| end_time | Date | 宝洁天猫旗舰店 | 资质过期时间 |
| supplement | String | 小 S 代言某某品牌 | 明星代言、特殊功效或等填写字段，具体见下面上传资质章节的定义 |

4) 返回示例

JSON 数据格式：

```
{
```

```

"tanx_qualification_solid_find_response":{
  "is_success":true,
  "qualification_list":{
    "qualification_dto":[
      {
        "name":"客户资质",
        "user_id":123,
        "user_name":"宝洁天猫旗舰店",
        "user_type":1,
        "element_id":123,
        "start_time":"2014-08-11 23:59:59",
        "end_time":"2016-04-1 23:59:59",
        "supplement":"小S代言某某品牌"
      }
    ]
  },
  "count":"200"
}
}

```

或者

```

{
  "error_response": {
    "code": 15,
    "msg": "Remote service error",
    "sub_code": "403",
    "sub_msg": "token验证失败"
  }
}

```

4.1.3.3 提交资质接口

该接口用于：调用该接口前请确保客户 ID 已存在。

接口定义如下：

- 1) 接口名称：taobao.tanx.qualification.add
- 2) 应用级输入参数：

| 名称 | 类型 | 必须 | 描述 | 示例值 |
|----|----|----|----|-----|
|----|----|----|----|-----|

| 名称 | 类型 | 必须 | 描述 | 示例值 |
|----------------|-----------------|----|--|------------|
| qualifications | qualification[] | 必须 | 名为 qualifications 的 json 对象数组,一次上传上限为 20 个 | |
| member_id | Number | 必须 | DSP 的 memberId | 123456 |
| token | String | 必须 | 验证令牌, 其值为: MD5 (tokenBaseKey + sign_time) | |
| sign_time | Number | 必须 | 当前时间戳, 1970-01-01 后的秒数 | 1382529851 |

其中, qualification 类型定义如下:

| 名称 | 类型 | 必须 | 描述 | 示例值 |
|--------------|----------|----|--|--|
| element_id | Number | 必须 | 客户资质元素的 id, 通过 id 确定上传资质元素 | 1 |
| end_time | Date | 可选 | 资质到期时间 | 2014-12-31 23:59:59 |
| name | String | 可选 | 本次上传资质名称(可自定义),不传系统自动生成 | 骆驼天猫旗舰店营业资质 |
| start_time | Date | 可选 | 资质生效时间 | 2014-11-11 23:59:59 |
| supplement | String | 可选 | 资质特殊属性名称(如特殊功效化妆品名称或明星代言的明星名字), 如何填写请见离线的资质规则 | |
| url_contents | String[] | 必填 | 资质内容, 可能是店铺 url 或者图片地址, 如果是图片请先调用 taobao.tanx.qualification.picture.upload 接口. 图片生成接口详情见 4.1.3.5 | ["http://img04.daily.taobaocdn.net/tfscom/TB1SAeKXXXXXc5XXXXXXXXXX", "http://img02.daily.taobaocdn.net/tfscom/TB1A75KXXXXXa7XXXXXXXXXXXX"] |
| user_id | Number | 必须 | 客户 id | 对应前面的 advertiser_id |

3) 返回结果:

| 名称 | 类型 | 是否必须 | 示例值 | 描述 |
|------------|----------|------|---------|----------------|
| is_success | Boolean | | | 是否成功 |
| id_list | Number[] | | 123,456 | 按照顺序返回插入的资质 id |

4) 返回示例

JSON 数据格式:

```
{
  "tanx_qualification_add_response":{
    "is_success":true,
    "id_list":{
      "number":[
        123,
        345
      ]
    }
  }
}
```

或者

```
{
  "error_response": {
    "code": 15,
    "msg": "Remote service
error",
    "sub_code": "403",
    "sub_msg": "token验证失败"
  }
}
```

4.1.3.4 修改资质接口

该接口用于：修改已上传的审核拒绝或待审核状态的资质，审核通过的不允许修改。

接口定义如下：

1) 接口名称：taobao.tanx.qualification.modify

2) 应用级输入参数：

| 名称 | 类型 | 必须 | 描述 | 示例值 |
|----------------|-----------------|----|---|--------|
| qualifications | Qualification[] | 必须 | 名为 qualifications 的 json 对象数组，一次修改上限 20 条 | |
| member_id | Number | 必须 | DSP 的 memberId | 123456 |

| 名称 | 类型 | 必须 | 描述 | 示例值 |
|-----------|--------|----|---|------------|
| token | String | 必须 | 验证令牌，其值为：MD5 (tokenBaseKey + sign_time) | |
| sign_time | Number | 必须 | 当前时间戳，1970-01-01 后的秒数 | 1382529851 |

其中，qualification 类型定义如下：

| 名称 | 类型 | 必须 | 描述 | 示例值 |
|--------------|----------|----|---|--|
| id | Number | 必须 | 客户资质 id，通过 id 确定修改的资质记录 | 1 |
| end_time | Date | 可选 | 资质到期时间 | 2014-12-31 23:59:59 |
| start_time | Date | 可选 | 资质生效时间 | 2014-11-11 23:59:59 |
| supplement | String | 可选 | 资质特殊属性名称(如特殊功效和明星代言的明星名字) | |
| url_contents | String[] | 可选 | 资质内容，可能是店铺 url 或者图片地址，如果是图片请先调用 taobao.tanx.qualification.picture.upload 接口，图片生成接口详情见 4.1.3.5 | ["http://img04.daily.taobaocdn.net/tfscom/TB1SAeKXXXXXXc5XXXXXXXXXX", "http://img02.daily.taobaocdn.net/tfscom/TB1A75KXXXXXXa7XXXXXX"] |

3) 返回结果：

| 名称 | 类型 | 是否必须 | 示例值 | 描述 |
|------------|---------|------|-----|------|
| is_success | Boolean | | | 是否成功 |

4) 返回示例

JSON 数据格式：

```
{
  "tanx_qualification_industry_add_response": {
    "is_success": true
  }
}
或者
```



```
{
  "error_response": {
    "code": 15,
    "msg": "Remote service error",
    "sub_code": "403",
    "sub_msg": "token验证失败"
  }
}
```

4.1.3.5 上传资质图片接口

该接口用于：使用本接口上传资质图片到 Tanx 服务器，获得图片路径，在提交资质时引用该路径。

接口定义如下：

1) 接口名称：taobao.tanx.qualification.picture.upload

2) 应用级输入参数：

| 名称 | 类型 | 必须 | 描述 | 示例值 |
|-----------|--------|----|---|------------|
| fileByte | byte[] | 必须 | 资质图片的 byte 数组形式，不能大于 1M | |
| member_id | Number | 必须 | DSP 的 memberId | 123456 |
| token | String | 必须 | 验证令牌，其值为：MD5 (tokenBaseKey + sign_time) | |
| sign_time | Number | 必须 | 当前时间戳，1970-01-01 后的秒数 | 1382529851 |

3) 返回结果：

| 名称 | 类型 | 是否必须 | 示例值 | 描述 |
|------------|---------|------|-----|-----------------|
| is_success | Boolean | | | 是否成功 |
| url | String | | | 存在淘宝服务器的 url 地址 |

4) 返回示例

JSON 数据格式:

```
{
  "tanx_qualification_picture_upload_response": {
    "is_success": true,
    "url":
"http://img02.daily.taobaocdn.net/tfscom/TB1h6qKXXXXXXaUXXXXXXXXXXXXX"
  }
}
或者
{
  "error_response": {
    "code": 15,
    "msg": "Remote service error",
    "sub_code": "403",
    "sub_msg": "token验证失败"
  }
}
```

4.1.3.6 批量获取资质接口

该接口用于：批量获取当前用户提交的资质信息。

接口定义如下：

1) 接口名称：taobao.tanx.qualification.find

2) 应用级输入参数：

| 名称 | 类型 | 必须 | 描述 | 示例值 |
|-----------|--------------------|----|---|------------|
| query | QualificationQuery | 可选 | | |
| page | Number | 必须 | 查询页起始 | |
| page_size | Number | 必须 | 分页最大值 | |
| member_id | Number | 必须 | DSP 的 memberId | 123456 |
| token | String | 必须 | 验证令牌，其值为： MD5 (tokenBaseKey + sign_time) | |
| sign_time | Number | 必须 | 当前时间戳， | 1382529851 |

| 名称 | 类型 | 必须 | 描述 | 示例值 |
|----|----|----|-----------------|-----|
| | | | 1970-01-01 后的秒数 | |

其中，QualificationQuery 类型定义如下：

| 名称 | 类型 | 必须 | 描述 | 示例值 |
|-------------------|----------|----|---|-----|
| asc | Number | | 正序还是倒序 1 是正,0 是倒 | 可选 |
| ids | Number[] | | 资质 id | 可选 |
| audit_status | Number | | 审核状态 -1=拒绝, 0=待审核, 1=通过 | 可选 |
| audit_time_begin | Date | | 审核时间左区间 | 可选 |
| audit_time_end | Date | | 审核时间右区间 | 可选 |
| create_time_begin | Date | | 创建时间左区间 | 可选 |
| create_time_end | Date | | 创建时间右区间 | 可选 |
| effective_status | Number | | 资质生效状态(该状态值是根据不能在新增资质时设置)-1=已过期, 0=待生效, 1=生效中, 2=即将过期 | 可选 |
| element_ids | Number[] | | 查询时包含的资质元素 id 列表 | 可选 |
| end_time_begin | Date | | 失效时间右区间 | 可选 |
| end_time_end | Date | | 审核时间右区间 | 可选 |
| order_by | String | | 排序字段。可选：id,name,userid,username,usertype,elementid,dspid,starttime,endtime,createtime,audittime | 可选 |
| start_time_begin | Date | | 生效时间左区间 | 可选 |
| start_time_end | Date | | 生效时间右区间 | 可选 |
| user_ids | Number[] | | 对应前面的 advertiser_id, 查询时和 user_names 选其一 | 可选 |
| user_names | String[] | | 对应前面的 advertiser_name。查询时和 user_ids 请只选其一 | 可选 |
| user_type | Number | | 客户类型 0-淘宝, 1-天猫, 2-非淘系公司客户, 3-非淘系个人客户 | |

3) 返回结果:

| 名称 | 类型 | 描述 |
|--------------------|--------------------|-----------|
| is_success | Boolean | 是否成功 |
| qualification_list | QualificationDto[] | 返回 Dto 列表 |
| count | String | 查询返回总条数 |

其中，QualificationDto类型定义如下:

| 名称 | 类型 | 描述 |
|------------------|----------|-----------------------------------|
| id | Number | 资质 id |
| user_id | Number | 客户 id。对应前面的 advertiser_id |
| name | String | 资质名称 |
| user_name | String | 客户名称。 对应前面的 advertiser_name |
| user_type | Number | 客户类别（0-淘宝，1-天猫，2-dsp 公司，3-dsp 个人） |
| element_id | Number | 资质元素 id |
| content_list | String[] | 资质内容。上传的 URL 或图片地址 |
| effective_status | Number | -1=已过期，0=待生效，1=生效中，2=即将过期 |
| audit_status | Number | 资质审核状态 -1=拒绝，0=待审核，1=通过 |
| start_time | Date | 资质生效时间 |
| end_time | Date | 资质失效时间 |
| create_time | Date | 资质创建时间 |
| update_time | Date | 资质修改时间 |
| audit_time | Date | 资质审核时间 |
| reason | String | 拒绝原因 |
| supplement | String | 用户附加内容（如明星名字,特殊功效化妆品名称） |

返回示例

JSON 数据格式:

```
{
  "tanx_qualification_find_response": {
    "qualification_list": {
```

```

        "qualification_dto": [
            {
                "id": 123,
                "name": "客户资质",
                "user_id": 123,
                "user_name": "宝洁天猫旗舰店",
                "user_type": 1,
                "element_id": 123,
                "content_list": {
                    "string": [
                        "www.taobao.com"
                    ]
                },
                "effective_status": 1,
                "audit_status": 1,
                "dsp_id": 123,
                "start_time": "2000-01-01 00:00:00",
                "end_time": "2000-01-01 00:00:00",
                "create_time": "2000-01-01 00:00:00",
                "update_time": "2000-01-01 00:00:00",
                "audit_time": "2000-01-01 00:00:00",
                "reason": "资质内容和图片不符",
                "supplement": "明星资质"
            }
        ]
    },
    "is_success": true,
    "count": "200"
}
}
或者
{
    "error_response": {
        "code": 15,
        "msg": "Remote service error",
        "sub_code": "403",
        "sub_msg": "token验证失败"
    }
}
}

```

4.2 离线数据

4.2.1 推广位过滤信息

`publisher` 目前可以通过 4 种方式设置过滤内容，分别是：

- ✓ 禁止的敏感类目：对应 `BidRequest` 的 `excluded_sensitive_category` 字段
- ✓ 禁止的广告行业类目：对应 `BidRequest` 的 `excluded_ad_category` 字段
- ✓ 禁止的创意类型：对应 `BidRequest` 的 `excluded_filter` 字段
- ✓ 禁止的目标跳转 URL：对应 `BidRequest` 的 `excluded_click_through_url` 字段

DSP 可以有两种方式获取这些数据：

- 1) 通过 API，即实时竞价请求（`BidRequest`）获取，但对于某些设置较多的媒体来说会有截断的情况。
- 2) 通过离线方式获取。离线获取的数据为全量数据。

推荐使用离线方式获取全量数据，建议每隔 1 小时获取一次此数据。获取方式如下：

- 1) 请求地址：

`http://www.tanx.com/pubfilter/download?logName=abc@xxx.com&token=1f1b2299eb9ae1d047c374149549d1c9&signTime=1401967006`

- 2) 参数解释：

`logName`：DSP 的帐户登录邮箱

`signTime`：触发请求事件时的系统时间(东八区)，约定为 `System.currentTimeMillis()/1000`

`token`：`MD5(tokenBaseKey+signTime)`，`tokenBaseKey` 与 `signTime` 为字符串直接拼接，`tokenBaseKey` 请联系 Tanx 团队获取。

- 3) 准入条件：

- ✓ `logName` 用户存在，并且有下载离线数据的权限。
- ✓ `signTime` 和系统的时间戳不能超过 30s。
- ✓ `token` 必须能校验通过。

- 4) 返回结果：

文件名：`ExcludePubFilteryyyyMMdd.json`。例如：`ExcludePubFilter20120628.json`

文件结束判断：正确完整的文件是以“^D”(即“\u0004”)字符作为最后一行。如果获取到的

文件最后一行不是“^D”(即“\u0004”), 则文件获取失败, 请重新获取。

文件格式: 多行, 每一行 (除最后一行外) 是一个 json encode 后的 object。格式如下:

```
{ "excluded_ad_category": [70101, 70102], "excluded_brand_advertiser": ["abc.com", "www.de
f.com"], "excluded_click_through_url": ["www.123.com", "www.456.com"], "excluded_filter": [
1, 2, 3, 9], "excluded_sensitive_category": [50001, 50002, 50007], "included_adboard_level": ["99
"], "pid": "mm_1_1_1" }
```

字段解释如下:

- ✓ pid: 推广位 ID, 对应于 BidRequest.AdzInfo.pid
- ✓ excluded_ad_category: 该 PID 上禁止的广告行业类目, 对应于 BidRequest 中的 excluded_ad_category 字段。
- ✓ excluded_sensitive_category: 该 PID 上禁止的敏感类目, 对应于 BidRequest 中的 excluded_sensitive_category 字段。
- ✓ excluded_brand_advertiser/excluded_click_through_url: 该 PID 上禁止的目标跳转 url (广告点击跳转的最终页面), 对应于 BidRequest 中的 excluded_click_through_url 字段。
- ✓ excluded_filter: 该 PID 上禁止的广告创意类型。对应于 BidRequest 中的 AdzInfo.excluded_filter 字段。
- ✓ included_adboard_level: 该 PID 上允许的创意等级。DSP 可通过[实时查询接口](#)查询创意的审核结果。

5 附加说明

5.1 字典文件

| 文件 | 描述 | 对应字段 | | |
|--|--------------|-----------------------------|--|--------------------|
| | | BidRequest | BidResponse | MobileCreative |
| Tanx-dict-site-category.txt | 网站分类 | category | - | - |
| Tanx-dict-view-type.txt | 推广位 展现形式 | AdzInfo.view_type | - | view_type |
| Tanx-dict-content-category.xlsx | 网页类目 | ContentCategory.id | - | - |
| Tanx-dict-ad-category-deprecated.txt (即将废弃) | 广告 行业类目 | excluded_ad_category | Ads.category | - |
| Tanx-dict-ad-category.xlsx | 广告 行业类目 | excluded_ad_category | Ads.category | - |
| Tanx-dict-sensitive-category.txt | 敏感类目 | excluded_sensitive_category | Ads.category | - |
| Tanx-dict-creative-type.txt | 创意类型 | AdzInfo.excluded_filter | Ads.creative_type | - |
| Tanx-dict-mobile-creative-attr.xlsx | 广告创意 属性 | - | - | Creative.Attr.name |
| Tanx-dict-api-framework.txt | 交互协议 | AdzInfo. api | | |
| Tanx-dict-landing-type.txt | 广告打开 方式 | Mobile.landing_type | Tanx-dict-landing-type.txt | 广告打开方式 |
| Tanx-dict-pmp-province-code.txt | pmp 省份 字典 | | | |

5.2 相关下载

| 文件 | 说明 | 最后更新时间 |
|--|----------------------------|------------|
| tanx-bidding.proto | 实时竞价接口 API | 2014/03/18 |
| tanx_request.tgz | 离线测试工具，可模拟 Tanx 的竞价请求 | 2013/02/06 |
| query_api.tgz （Java） | 实时查询接口的示例代码 | 2014/03/18 |
| decode_price.tgz （C++） | 结算价格解密的示例代码 | 2012/11/06 |
| decode_device_id.tgz （C++） | 设备 ID 解码的示例代码 | 2014/05/05 |
| mobile_mruid_creative_test.tgz | Mruid 创意测试工具，适用 Android 设备 | 2014/10/24 |

5.3 优化建议

5.3.1 保持长连接

为了减少连接的新建和时延，需要 Tanx 和 DSP 的服务器之间保持连接的状态，即双方均开启长连接。连接保持的空闲超时时间不小于 5 分钟，建议 15 分钟以上。例如 Apache 中，应设置 `KeepAliveTimeout` 为 900 或者以上，同时将 `MaxKeepAliveRequests` 设置为 0（尽可能的复用连接）；在 nginx 中应设置 `keepalive_timeout` 为 900 或者以上。

5.4 特别说明

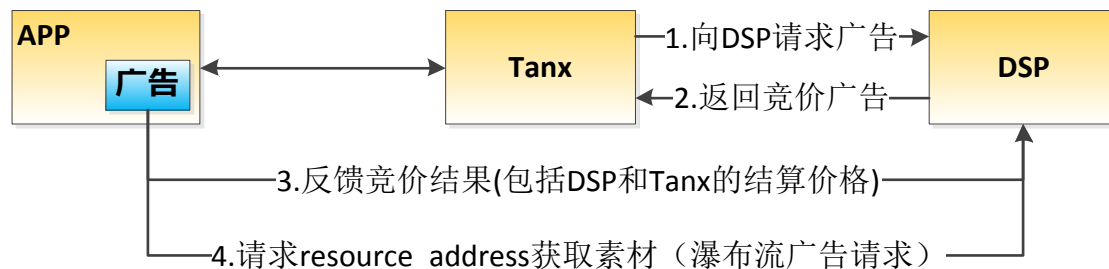
本章节主要描述 DSP 在接入不同平台的不同类型广告时需要注意的事项，帮助 DSP 快速了解需要做出的调整。完整的使用说明，请查看[使用指南部分](#)。

5.4.1 移动设备上的横幅推广位

- 1) DSP 需要在 html_snippet 字段中声明创意。
- 2) DSP 需要在 feedback_address 中包含结算价格宏。对于 APP 流量（Mobile.is_app 为 1），点击宏为可选。
- 3) 对于 APP 中需要使用 mraid 交互的流量，请仔细阅读基于 MRAID 协议的扩展 API 接口文档。

5.4.2 Native

和普通的广告不同，Native 的展现形式，如果在应答中不返回素材，会多一次 resource_address 的请求，且可能伴随着用户的操作（比如下拉刷新）而多次请求 resource_address。如下图：



- 1) 在 Tanx 和 DSP 的实时竞价接口中，Tanx 会通过 BidRequest.Mobile.ad_num 字段告知 DSP 本推广位上最多需要的创意数量。DSP 需要在 resource_address 的请求应答中返回不超过该数量的创意（需要使用 MobileCreative 结构组织多个创意）。
- 2) DSP 需要在 resource_address 字段中声明创意地址。并在该字段中必须包含创意序列宏。创意序列宏会在请求 resource_address 时展开。详细请查看[宏说明的创意序列宏部分](#)。
- 3) 在请求 resource_address 获取素材时，DSP 需要保证返回的素材必须是以创意应答接口格式（即 MobileCreative）进行组织的。详细请查看[创意应答接口部分](#)。

- 4) 对于竞价成功的 DSP，多次请求 resource_address 不会重复计费（仅计费一次）。