

# 百度流量交易服务接入规范

版本：3.0.1

最后更新时间：2016-11-21

此文档仅供 DSP 公司与百度流量交易服务对接使用，不得向第三方传播

## 目录

一、百度流量交易服务接口协议 .....	1
1、百度流量交易服务接口说明 .....	1
2、百度流量交易服务向 DSP 发送的广告请求接口(BID REQUEST).....	2
3、DSP 向百度流量交易服务返回的竞价广告接口（BID RESPONSE） .....	20
4、百度流量交易服务向 DSP 发送的竞价结果接口（WIN NOTICE） .....	27
二、COOKIE MATCHING .....	28
三、示例.....	30
附录 1：扩展广告 SDK 脚本文档 .....	31

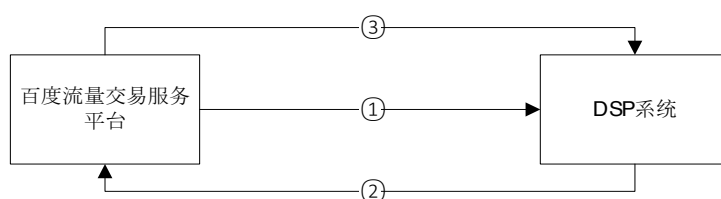
## 一、百度流量交易服务接口协议

### 1、百度流量交易服务接口说明

百度流量交易服务协议包括三个接口：

- ① 百度流量交易服务向 DSP 发送的广告请求接口(Bid Request)
- ② DSP 向百度流量交易服务返回的竞价广告接口 ( Bid Response )
- ③ 百度流量交易服务向 DSP 发送的竞价结果接口 ( Win Notice )

关系如下：



## 2、百度流量交易服务向 DSP 发送的广告请求接口(Bid Request)

广告请求接口使用 http post 方式进行交互，消息内容使用 protobuf 封装，序列化之后传送，访问超时设置为 130ms。Content-Type 设为 application/octet-stream。每个请求进行一个广告位的竞价。

### 广告请求接口组成



### protobuf 接口

```
message BidRequest {  
    // 请求 ID, 唯一标识本次请求, 明文字符串  
    required string id = 1;  
  
    // **** 用户信息 ****  
    // 用户 IP 地址, 点分十进制字符串  
    optional string ip = 2;  
    // User-Agent  
    optional string user_agent = 3;  
    // 用户 ID, 已于 2015 年 4 月 30 日停止使用  
    optional string baidu_user_id = 4;  
    // 用户 ID 版本, 已于 2015 年 4 月 30 日停止使用  
    optional int32 baidu_user_id_version = 5;  
    // 用户 ID
```

```
// 在 2015 年 5 月 1 日后, cookie 的版本号为 2
message BaiduId {
    optional string baidu_user_id = 1;
    optional int32 baidu_user_id_version = 2;
}
repeated BaiduId baidu_id_list = 8;
// DSP 托管到 BES 的 USER ID
optional string buyer_user_id = 32;
// 用户兴趣
repeated int64 user_category = 6;
// 自定义的用户标签
message CustomizedUserTag {
    //DSP 在 AMS 服务中定义的用户标签
    message AmsTag {
        optional uint64 tag_id = 1;
    }
    repeated AmsTag ams_tag_list = 1;
}
optional CustomizedUserTag customized_user_tag = 31;
// 用户性别, 在 2015 年 7 月 10 日停止使用, 性别信息在用户兴趣字段中提供
enum Gender {
    UNKNOWN = 0;
    MALE = 1;
    FEMALE = 2;
}
optional Gender gender = 7;
// 页面语言
optional string detected_language = 9;
// flash 版本
optional string flash_version = 10;

// **** 位置信息 ****
message Geo {
    // 经纬度信息
    message Coordinate {
        // 经纬度坐标标准
        enum Standard {
            // 百度地图的经纬度坐标标准
            BD_09 = 0;
            // 国测局制定的经纬度坐标标准
            GCJ_02 = 1;
            // 国际经纬度坐标标准
            WGS_84 = 2;
        }
    }
}
```

```
// 百度地图的墨卡托坐标标准，以米为单位
BD_09_MC = 3;
}

// 地图坐标标准
optional Standard standard = 1;

// 纬度
optional float latitude = 2;

// 经度
optional float longitude = 3;
}
repeated Coordinate user_coordinate = 1;


// 用户位置信息
message UserLocation {
    // 省份，如北京市
    optional string province = 1;
    // 城市，如北京市
    optional string city = 2;
    // 区县，如海淀区
    optional string district = 3;
    // 街道，如中关村大街
    optional string street = 4;
}
optional UserLocation user_location = 2;
}
optional Geo user_geo_info = 28;


// **** 页面信息 ****
// 当前页面 URL
// 当流量来源是 IOS 的 APP 时，IOS 的应用下载链接会在 URL 字段中传输
optional string url = 11;
// 请求的 referer
optional string referer = 12;
// 网站分类
optional int32 site_category = 13;
// 网站质量类型
optional int32 site_quality = 14;
// 页面类型
optional int32 page_type = 15;
// 页面关键词
repeated string page_keyword = 17;
```

```
// 页面内容质量
optional int32 page_quality = 18;

// 页面分类
optional int32 page_vertical = 21;

// 发布商不允许的广告行业，2015 年 7 月 15 日后信息同时通过 API 中的媒体过滤设置文件
提供，该字段停止使用时间待定
repeated int32 excluded_product_category = 19 [packed=true];

// **** 移动信息 ****
message Mobile {
    // **** 移动设备信息 ****

    // 百度唯一标识移动设备的编号
    optional string DEPRECATED_device_id = 1;

    // 新版移动设备序列号标识字段。允许同时存储多个序列号
    message MobileID {
        // 移动设备 ID 的类型
        enum IDType {
            // 未知
            UNKNOWN = 0;
            // IMEI
            IMEI = 1;
            // MAC 地址
            MAC = 2;
        }

        optional IDType type = 1; // 序列号类型

        optional string id = 2; // 序列号
    }

    repeated MobileID id = 13;

    // 设备类型定义
    enum MobileDeviceType {
        UNKNOWN_DEVICE = 0;
        HIGHEND_PHONE = 1;
        TABLET = 2;
    }

    // 设备类型
    optional MobileDeviceType device_type = 2;

    // 移动平台名，例如 android, iphone 等等
    enum OS {
        UNKNOWN_OS = 0;
        IOS = 1;
        ANDROID = 2;
        WINDOWS_PHONE = 3;
    }
}
```

```
optional OS platform = 3 [default = UNKNOWN_OS];
// 移动操作系统版本号
// 例如 Android 2.1, major, minor 分别是 2,1
// 例如 Iphone 4.2.1, major, minor, micro 分别是 4,2,1
message DeviceOsVersion {
    optional int32 os_version_major = 1;
    optional int32 os_version_minor = 2;
    optional int32 os_version_micro = 3;
}

optional DeviceOsVersion os_version = 4;
// 设备品牌
optional string brand = 5;
// 设备机型
optional string model = 6;
// 设备屏宽
optional int32 screen_width = 7;
// 设备屏高
optional int32 screen_height = 8;
// 设备屏幕像素密度
optional float screen_density = 15;
// 运营商
optional int64 carrier_id = 9;
// 无线网络类型
enum WirelessNetworkType {
    UNKNOWN_NETWORK = 0;
    WIFI = 1;
    MOBILE_2G = 2;
    MOBILE_3G = 3;
    MOBILE_4G = 4;
}
optional WirelessNetworkType wireless_network_type = 10;
// 移动设备上为广告控制提供的 ID
optional string DEPRECATED_for_advertising_id = 11;
// 新版移动设备用户识别号字段，允许多种类型同时存在
// 推荐使用本字段获取用户 ID，原字段会逐步废弃
message ForAdvertisingID {
    enum IDType {
        // 未知
        UNKNOWN = 0;
        // Android 适用
        ANDROID_ID = 4;
        // IOS 适用
```



```
IDFA = 5;
}

optional IDType type = 1; // ID 类型
optional string id = 2; // ID 内容
}
repeated ForAdvertisingID for_advertising_id = 14;

// **** 移动应用信息 ****
message MobileApp {
    // 百度移动联盟为该 App 分配的 app id
    optional string app_id = 1;
    // App 应用的 package 全名
    optional string app_bundle_id = 2;
    // App 应用分类
    optional int32 app_category = 3;
    // App 开发者 ID
    optional int32 app_publisher_id = 4;
    // App 允许的交互类型定义
    // 电话、下载、应用唤醒
    enum AppInteractionType {
        TELEPHONE = 0;
        DOWNLOAD = 1;
        DEEPLINK = 2;
    }
    // App 允许的交互类型
    repeated AppInteractionType app_interaction_type = 5;
}
// 移动应用信息，当流量来自 App 时该字段非空
optional MobileApp mobile_app = 12;
}
// 移动信息，当流量来自移动设备时该字段非空
optional Mobile mobile = 29;

// **** 视频信息 ****
Message video {
    //视频的标题
    optional string title = 1;
    //视频的标签
    repeated string tags = 2;
    //视频的播放时长
    optional int32 content_length = 3;
    //视频的频道 ID
```

```
    repeated int64 channel_id = 4;
}
Optional Video video = 30;

// **** 广告位信息 ****
message AdSlot {
    // 广告位 ID, 全局唯一 ID
    optional uint64 ad_block_key = 1;
    // 当前页面广告位顺序 id, 同一页面从 1 开始
    optional int32 sequence_id = 2;
    // 展示类型
    optional int32 adslot_type = 3;
    // 宽
    optional int32 width = 4;
    // 高
    optional int32 height = 5;
    // 实际宽
    optional int32 actual_width = 18;
    // 实际高
    optional int32 actual_height = 19;
    // 展示位置
    optional int32 slot_visibility = 6;
    // 发布者允许的创意类型
    repeated int32 creative_type = 7 [packed=true];
    // 发布者允许的创意封装和渲染类型
    // 1 静态创意
    // 2 动态创意
    enum CreativeDescType {
        STATIC_CREATIVE = 1;
        DYNAMIC_CREATIVE = 2;
    };
    repeated CreativeDescType creative_desc_type = 22;
    // 发布者不允许的 landing page url
    //2015 年 7 月 15 日后信息同时通过 API 中的媒体过滤设置文件提供, 该字段停止使用时间
    待定
    repeated string excluded_landing_page_url = 8;
    // 媒体过滤设置信息的 ID
    repeated fixed64 publisher_settings_list_id = 14;
    // 发布者设置的底价, 单位分
    optional int32 minimum_cpm = 9;
```

```
// 视频广告的最大时长
// 该字段已于 2015 年 6 月 3 日停止使用，目前使用 VideoInfo 中的字段
optional int32 max_video_duration = 10;
// 视频广告的最小时长
// 该字段已于 2015 年 6 月 3 日停止使用，目前使用 VideoInfo 中的字段
optional int32 min_video_duration = 11;
// 视频广告的播放延时， 0 及正整数表示前贴，-1 表示中贴，-2 表示后贴
// 该字段已于 2015 年 6 月 3 日停止使用，目前使用 VideoInfo 中的字段
optional int32 video_start_delay = 12;

// 视频广告信息
message VideoInfo {
    // 视频广告的最大时长
    optional int32 max_video_duration = 1;
    // 视频广告的最小时长
    optional int32 min_video_duration = 2;
    // 视频广告的播出延时， 0 及正整数表示前贴，-1 表示中贴，-2 表示后贴
    optional int32 video_start_delay = 3;
}
optional VideoInfo video_info = 15;

message LinkUnitInfo {
    // 链接单元广告候选样式
    // 在 request 中，BES 会计算在此次请求中最优的广告样式集合，并将此集合
    // 发送给 DSP，最后使用的样式将根据 DSP 返回的关键词组合进行确定。
    repeated int32 style_type = 1;
    // 链接单元广告可以返回的关键词个数
    optional int32 req_keyword_num = 2;
    // BES 推荐使用的关键词，dsp 可选择使用，也可以使用自己认为更好的关键词
    // 使用 utf-8 编码
    // 推荐使用 keywords 字段，proposed_keyword 字段会逐步废弃
    repeated string proposed_keyword = 3;
    // 替代 proposed_keyword，容纳词明文之外的其它字段
    message ProposedKeyword {
        // BES 推荐使用的关键词，dsp 可选择使用，也可以使用自己认为更好的关键词
        // 使用 utf-8 编码
        // 与原 proposed_keyword 字段一致，做为原字段的替代
        optional string text = 1;
        // 推荐词的点击率预估，供 DSP 参考；单位十亿分之 1
    }
}
```

```
        optional float pctr1 = 2;
    }
    repeated ProposedKeyword keywords = 4;
}
optional LinkUnitInfo link_unit_info = 16;

// 优先交易信息
message PreferredOrderInfo {
    message PreferredOrder {
        // 优先交易 ID
        optional int64 order_id = 1;
        // 该优先交易的交易价格
        optional int64 fixed_cpm = 2;
    }
    // 一次请求可能包含多个优先交易
    repeated PreferredOrder preferred_orders = 1;
    // 允许优先交易的同时，是否允许竞价
    optional bool allow_auction = 2 [default=true];
}
optional PreferredOrderInfo preferred_order_info = 13;

// 包断订单信息
message GuaranteedOrder {
    // 包断订单 id，全局唯一
    optional uint32 order_id = 1;
    // 订单计费信息
    message Charge {
        // 订单价格
        optional int32 price = 1;
        // 计费类型
        enum ChargeType {
            CHARGE_GD_CPM = 2; // 包断订单以 CPM 方式计价
        }
        optional ChargeType charge_mode = 2 [default = CHARGE_GD_CPM];
    }
    optional Charge charge = 2;
}
repeated GuaranteedOrder guaranteed_orders = 20;

// 可扩展创意信息
message ExpandCreativeInfo {
    // 触发动作类型
    enum ActionType {
        // 悬停
        HOVER = 1;
    }
}
```

```
// 点击
CLICK = 2;

// 加载
LOAD = 3;
}
repeated ActionType expand_action_type = 1;
// 扩展方向
enum Direction {
    EXPAND_NONE = 0;
    // 向上扩展
    EXPAND_UP = 1;
    // 向下扩展
    EXPAND_DOWN = 2;
    // 向左扩展
    EXPAND_LEFT = 3;
    // 向右扩展
    EXPAND_RIGHT = 4;
};
optional Direction expand_direction = 2;
// 扩展创意允许的创意类型
repeated int32 expand_creative_type = 3 [packed=true];
// 扩展创意的宽度
optional int32 expand_width = 4;
// 扩展创意的高度
optional int32 expand_height = 5;
// 扩展创意展现时长
optional int32 expand_duration = 6;
}
optional ExpandCreativeInfo expand_creative_info = 17;
// 广告位级别
enum AdSlotLevel {
    UNKNOWN_ADB_LEVEL = 0;
    TOP = 1;    // 优质广告位
    MED = 2;    // 中端广告位
    TAIL = 3;   // 长尾广告位
    LOW = 4;    // 低质广告位
}
optional AdSlotLevel adslot_level = 21 [default=UNKNOWN_ADB_LEVEL];
// 原生广告诉求参数
message NativeAdParam {
    enum Fields {
```

```
// 标题
TITLE = 0x1;
// 内容描述
DESC = 0x2;
// 主题图
IMAGE = 0x4;
// logo 图标
LOGOICON = 0x8;
// 投放时下载的 APP 大小
APPSIZE = 0x10;
}
message ImageEle {
    // 宽
    optional int32 width = 1;
    // 高
    optional int32 height = 2;
    // 形状,
    // 0 没有形状要求,
    // 1 矩形,
    // 2 圆形,
    // 3 半圆形
    optional int32 shape = 3 [default = 0];
}
// 按照位图设置相应的位为 1, 不需要请求的位保持 0
// 示例: 原生广告必须包含标题和图标, 则 required_fields = (0x1 | 0x8) = 9
repeated int64 required_fields = 1;
// 标题最大长度
optional int32 title_max_length = 2;
// 描述最大长度
optional int32 desc_max_length = 3;
// 广告主 logo 或图标的宽高、形状要求
optional ImageEle logo_icon = 4;
// 主题图的宽高、形状要求
optional ImageEle image = 5;
// 主题图数量
optional int32 image_num = 6;
}
// 是否允许返回非原生广告创意,
// 如果为 true, 则对于存在原生诉求的情况下可以返回非原生广告创意,
```

```

// 如果为 false，则必须返回原生创意
optional bool allowed_non_nativead = 23 [default=true];
optional NativeAdParam nativead_param = 24;
// 是否为 HTTPS 请求
// 如果为 true，则所有资源（图片、视频等）必须以 HTTPS 返回
// 注意：url 字段对应协议与 secure 字段的值并无严格对应关系，
// 比如，存在 url 协议为 HTTP 而 secure 为 true 的情况，
// 因此，需要使用 secure 字段来决定是否以 HTTPS 返回资源，而不要依赖 url 字段
optional bool secure = 25 [default=false];
}
// 默认每次请求一个广告位
repeated AdSlot adslot = 20;

// **** 系统使用 ****
optional bool is_test = 26 [default=false];
optional bool is_ping = 27 [default=false];
}

```

#### 广告请求接口说明

- [1] id：请求 ID，明文字符串，唯一标识本次请求。
- [2] ip：用户 ip 地址，点分十进制字符串，utf8 格式。
- [3] user\_agent：用户 user agent 信息，utf8 格式。
- [4] baidu\_user\_id：百度用户 id，用于标记访问的用户。
- [5] baidu\_user\_id\_version：百度用户 ID 版本，用于用户 ID 升级时使用。
- [6] buyer\_user\_id：DSP 托管到 BES 的 USER ID。
- [7] user\_category：用户兴趣信息，按照权重从高到低最多发送 20 个兴趣信息。对应关系见用户兴趣字典（2015 年 4 月 16 日开始使用新人群标签 1.0）。
- [8] tag\_id：DSP 通过 AMS 设置的自定义标签，当流量命中标签时会填写在此字段。
- [9] gender：用户性别。该信息为用户行为分析结果，不保证百分百准确。

值	说明
UNKNOWN	未识别性别
MALE	男
FEMALE	女

- [10] detected\_language：页面语言。
- [11] flash\_version：flash 版本。当前浏览器或者设备上的 flash 播放器版本。
- [12] coordinate：坐标信息。根据能获取的数据提供百度坐标体系数据或国测局标准、国际经纬度坐标标准的数据。

[13] userLocation : 用户位置信息。提供省份、城市、区县、街道 4 级数据。

[14] url : 广告位所在的 url 信息。utf8 格式。当流量来源是 IOS 的 APP 时, IOS 的应用下载链接会在 URL 字段中传输。

[15] referer : referer url 信息。utf8 格式。

[16] site\_category : 网站分类 ID, ID 对应关系在网站分类字典中提供。

[17] site\_quality : 网站质量类型

值	说明
0	未知
1	流量质量较低
2	流量质量较高
3	流量质量很高

[18] page\_type : 页面类型

值	说明
0	未定义
1	正文页
2	索引页
3	视频页
4	论坛帖子
5	图片页
6	漫画页
7	主页
8	介绍页
9	小说页
10	下载页
11	问答页
12	搜索结果页
13	广告页

[19] page\_keyword : 页面关键词。可以提供最多 3 个, 每个词最多 5 个字, utf8 格式。

[20] page\_quality : 页面内容质量。

值	说明
0	正常页面
1	包含低俗内容页面

[21] page\_vertical: 页面分类, ID 对应关系见页面分类字典。



[22] excluded\_product\_category：不允许的广告行业，ID 对应关系见广告行业字典。

[23] device\_id：设备 ID。百度唯一标识移动设备的编号。该字段会逐渐被 MobileID 字段代替。

[24] MobileID：新版的移动设备 ID，可以解决之前 device\_id 只提供 IMEI 或者 MAC 地址并且不说明确定类型的问题，该字段提供设备 ID 的类型及对应的值，并且不同类型可以同时提供。

[25] device\_type：设备类型。

值	说明
0	未识别设备
1	手机
2	平板

[26] platform：操作系统

值	说明
0	其他及未识别
1	iOS
2	Android
3	Windows_Phone

[27] os\_version：移动操作系统版本号。

[28] brand：设备品牌。

[29] model：设备机型。

[30] screen\_width：设备屏宽，物理像素。

[31] screen\_height：设备屏高，物理像素。

[32] screen\_density：设备屏幕像素密度。

[33] carrier\_id：运营商。MCC+MNC:比如 4600x，前三位为 MCC ( Mobile Country Code )，后两位为 MNC ( Mobile Network Code )。

[34] wireless\_network\_type：网络类型。

值	说明
0	未识别
1	WIFI
2	2G
3	3G
4	4G

[35] for\_advertising\_id：操作系统为广告精准投放所提供的 ID。

[36] app\_id：APP ID。百度为该 App 分配的唯一 ID。

[37] app\_bundle\_id：App 应用的 package 全名。

[38] app\_category : APP 应用分类。ID 对应关系见分类词典。

[39] app\_publisher\_id : APP 开发者 ID。百度为开发者分配的唯一 ID。

[40] app\_interaction\_type : APP 允许的交互类型。

值	说明
0	电话直拨
1	点击下载
2	应用唤醒

[41] title : 视频的标题。

[42] tags : 视频的标签。

[43] content\_length : 视频的播放时长。

[44] channel\_id : 视频的频道 ID。

[45] ad\_block\_key : 广告位 ID，对广告位进行唯一标识。

[46] sequence\_id : 广告位顺序 id，为了同一页面多个广告位请求使用，从 1 开始计数。

[47] adslot\_type : 展示类型。

值	说明
0	固定
1	悬浮
8	图+ICON
11	插屏
12	开屏
13	原生
21	视频前贴片
22	视频中贴片
23	视频后贴片
26	视频暂停

[48] width : 广告位宽度，逻辑像素。

[49] height : 广告位高度，逻辑像素。

[50] actual\_width : 广告位实际宽度，逻辑像素。在移动设备上会出现[48]广告位实际宽度和[46]广告位宽度不同的情况，此字段可用 HTML 创意的缩放。

[51] actual\_height : 广告位实际高度，逻辑像素。在移动设备上会出现[49]广告位实际高度和[47]广告位高度不同的情况，此字段可用 HTML 创意的缩放。

[52] slot\_visibility : 广告位位置信息

值	说明
---	----

0	其它屏
1	第一屏
2	第二屏

[53] creative\_type : 允许的创意类型

值	说明
0	文本
1	图片
2	FLASH
4	图文
5	链接单元 ( linkunit )
7	Video

[54] creative\_desc\_type : 允许的创意封装和渲染类型

值	说明
1	静态创意
2	动态创意

注：静态创意：在投放时，通过选择事先上传到 BES 服务器的创意进行投放，这种形式叫静态创意投放。

动态创意：在投放时，通过在 response 中的 html\_snippet 字段中填写 html 代码、JS 代码等拼接各种元素（例如商品图片、商品名称、商品价格）进行投放，这种投放方式叫做动态创意投放。

[55] excluded\_landing\_page\_url : 发布者不允许的 landing page url 列表。

[56] publisher\_settings\_list\_id : 媒体保护设置信息的 ID。媒体保护设置信息通过 API 进行获取，API 规范详见《百度流量交易服务 API 接口规范》。

[57] mininum\_cpm : 发布者设置的底价，单位分。

[58] max\_video\_duration : 视频广告的最大时长，仅在 adslot\_type 为 21、22、23 的时候有值，单位是秒。

[59] min\_video\_duration : 视频广告的最小时长，仅在 adslot\_type 为 21、22、23 的时候有值，单位是秒。

[60] video\_start\_delay : 视频广告的播放延时。0 及正整数表示前贴，-1 表示中贴，-2 表示后贴。仅在 adslot\_type 为 21、22、23 的时候有值，单位是秒。

[61] style\_type : 链接单元广告候选样式。在媒体允许投放链接单元类的广告时( creative\_type 为 5 )，BES 会在 request 中计算在此次请求中最优的广告样式集合，此集合用该字段标识，最后使用的样式将根据 DSP 返回的关键词组合进行确定。

值	说明	示例
0	排行榜样式	
1	配图样式	
2	标签云样式	

[62] req\_keyword\_num : DSP 竞价时可以返回的最大关键词个数。

[63] proposed\_keyword : BES 推荐使用的关键词。DSP 可以选择使用，也可以使用自己认为更好的关键词。使用 UTF-8 编码。

[64] text : BES 推荐使用的关键词。DSP 可以选择使用，也可以使用自己认为更好的关键词。使用 UTF-8 编码；与原有 proposed\_keyword 字段一致，作为原字段的代替。

[65] pctr1 : 推荐词的点击率预估，供 DSP 参考。单位十亿分之一。

[66] order\_id : 优先交易的 ID。

[67] fixed\_cpm : 优先交易的交易价格。

[68] order\_id : 包断订单的 ID。

[69] price : 包断订单的单价，按照 chargetype 中标明的计费模式计算。

[70] chargetype : 订单的计费类型。

值	说明
CHARGE_GD_CPM	包断订单以 CPM 方式计价。

[71] actionType : 触发动作类型，指通可以触发扩展创意的动作。

[72] allow\_auction : 本次请求在允许优先交易的同时，是否允许竞价，默认值为 true。

[73] Direction : 扩展方向，创意扩展时的方向。

[74] expand\_creative\_type : 扩展创意允许的创意类型，可能与原始创意允许的创意类型不一致。

[75] expand\_width : 扩展创意的宽度。

[76] expand\_height : 扩展创意的高度。

[77] expand\_duration : 扩展创意展现时长，以扩展创意完全展开的时间点开始计时。

[78] adslot\_level : 广告位级别

值	说明
0	未知

1	优质广告位
2	中端广告位
3	长尾广告位
4	低质广告位

[79] required\_fields：原生广告需要的元素，按照位图设置相应的位为 1，不需要请求的位保持 0；元素顺序为：标题、内容描述、主题图、图标、投放时下载的 APP 大小。

[80] title\_max\_length：标题最大长度。

[81] desc\_max\_length：描述最大长度。

[82] logo\_icon：广告主 logo 或图标的宽高、形状要求，格式参考 message ImageEle。

[83] image：主题图的宽高、形状要求，格式参考 message ImageEle。

[84] image\_num：主题图数量，返回创意的主题图数量必须与要求的数量一致。

[85] allowed\_non\_nativead：是否能返回非原生创意，当值为 true 时，可参考 creative\_type 字段返回普通创意，当值为 false 时，必须返回原生创意。

[86] secure：是否为 HTTPS 请求。如果为 true，则所有资源（图片、视频等）及展示监测、点击监测、第三方监测等必须返回 HTTPS 协议的链接。请注意：url 字段（广告位所在的 url 信息）对应协议与 secure 字段的值并无严格对应关系，比如，存在 url 协议为 HTTP 而 secure 为 true 的情况，因此，需要使用 secure 字段来决定是否以 HTTPS 返回链接，而不要依赖 url 字段。

[87] adslot：广告位，目前百度流量交易服务只支持一个广告位的竞价请求。

[88] is\_test：是否是测试请求。接入测试使用，正式上线后设置为 false。

[89] is\_ping：是否是心跳请求。接入测试使用，正式上线后设置为 false。

### 3、DSP 向百度流量交易服务返回的竞价广告接口 ( Bid Response )

竞价结果返回接口以 http post 方式进行返回，消息内容使用 protobuf 进行封装，序列化之后返回。响应的 Content-Type 设为 application/octet-stream。不参与竞价，也必须在超时时限内返回 protobuf 格式的响应，超时或者返回信息无法解析被视为访问失败。

竞价结果返回接口有几种方式：Static、HTML Snippet。其中 Static 方式是最基本的接口，所有的 DSP 都可以采用。HTML Snippet 方式使用必须得到百度的确定，并且采用 HTML Snippet 方式返回的 Bid Response 必须小于 30K。

#### protobuf 接口

```
message BidResponse {  
    // 返回 ID，将请求中的 id 复制给返回 id，便于 session trace  
    required string id = 1;  
  
    // **** 竞价广告信息 ****  
    message Ad {  
        // 广告位顺序 ID  
        optional int32 sequence_id = 1;  
        // 创意 ID  
        optional int64 creative_id = 2;  
        // 代码段  
        optional string html_snippet = 7;  
        // 竞价链接单元广告  
        message LinkUnitKeyword {  
            // 竞价链接单元广告的关键词，utf-8 编码  
            optional string keyword = 1;  
            // 关键词竞价的样式  
            repeated int32 selected_style_type = 2;  
        }  
        optional LinkUnitKeyword link_unit_keyword = 18;  
        // 广告主 id  
        optional uint32 advertiser_id = 8;  
        // 创意宽  
        optional int32 width = 9;  
        // 创意高  
        optional int32 height = 10;  
        // 创意所属行业  
        optional int32 category = 11;  
        // 创意类型  
        optional int32 type = 12;  
        // 到达页面
```

```
optional string landing_page = 13;
// 点击链接
repeated string target_url = 14;
// 展示监测
repeated string monitor_urls = 17;
// 扩展创意信息
message ExpandCreativeInfo {
    // 扩展创意宽
    optional int32 creative_width = 1;
    // 扩展创意高
    optional int32 creative_height = 2;
    // 扩展创意着陆页
    optional string creative_landing_page = 3;
    // 扩展创意类型
    optional int32 creative_type = 4;
}
optional ExpandCreativeInfo expand_creative_info = 19;
// 最高竞价，单位分
optional int32 max_cpm = 3;
// 扩展参数
optional string extdata = 5;
// 是否进行 cookie matching
optional bool is_cookie_matching = 6;
// 如果选择优先交易，该交易的 ID
optional int64 preferred_order_id = 15;
// 如果采用包断投放，投放的 id
optional uint32 guaranteed_order_id = 20;
// APP 唤醒信息
// 仅当交互类型包含应用唤醒时才可以填写，DSP 可根据投放需求决定是否填写
message DeeplinkInfo {
    // 应用唤醒打开页面，当使用应用唤醒功能时，该字段为必填
    optional string deeplink_url = 1;
    // 应用版本
    optional uint32 app_version = 2;
    // 应用唤醒退化链接，当使用应用唤醒功能时，该字段为必填
    optional string fallback_url = 3;
    // 应用唤醒退化链接类型，当使用应用唤醒功能时，该字段为必填
    // LANDING_PAGE = 1, DOWNLOAD = 2
    optional uint32 fallback_type = 4;
}
optional DeeplinkInfo deeplink_info = 21;
```

```

}
repeated Ad ad = 2;

// **** 系统使用 ****

// debug 接口
optional string debug_string = 3;

// DSP 处理时间
optional int32 processing_time_ms = 4;
}
    
```

## 广告返回接口说明

- [1] id :返回信息的 ID ,这个 ID 要与 request id 一致 ,表明这个 response 与 request 的对应关系 ,放弃竞价或正常竞价时均必填。
- [2] sequence\_id :广告位顺序 ID ,与请求中的顺序 ID 对应 ,从 1 开始 ,在正常竞价时必填。
- [3] creative\_id :对于将创意上传到百度 ,由百度进行广告渲染的方式 ,creative\_id 是 DSP 内部的广告 ID ,这个 ID 唯一标识 DSP 上传到百度的创意。对于使用 html snippet ,DSP 自己进行广告渲染的方式 ,creative\_id 代表 html snippet 的 ID。当 html snippet 中的创意内容、监测地址以及 landing page 改变时 ,需要生成新的 creative\_id。建议使用 64 位整数 ,utf8 编码 ,在正常竞价时必填。
- [4] html\_snippet :仅在使用 html snippet 方式时使用。该字段中除了点击链接有特殊要求外 ,其他样式无要求。如果 html snippet 中有多个创意时 ,只需要在第一个创意的展示监测中加入%%PRICE%%宏。点击宏的使用方法如下 ,请注意 :在一次返回中 ,只能使用一种点击宏方法 ;在 html snippet 中 http 头部需要填充 content\_length 字段 ,并且使用标准的 http keep-alive 机制进行传输。

宏	说明
%%CLICK_URL_{N}%%	<ul style="list-style-type: none"> <li>使用此宏时必须同时填写 target_url 字段</li> <li>如果 html snippet 中有多个创意时 ,每个创意的点击链接位置仅需要使用宏%%CLICK_URL_{N}%% ,不需要完整的点击地址。完整的点击地址顺序填写到 target_url 字段中。</li> <li>这里{N}为具体的序号 ,代表第 N 个创意 ,N 从 0 开始计数</li> <li>DSP 的 click url 在页面上展现为被转义的形式。字符的具体转义规则 :                             <ul style="list-style-type: none"> <li>a 0~9, a~z, A~Z 不变 ;</li> <li>b 空格被转义为+ ( 加号 ) ;</li> <li>c 所有其他字符被转义为对应的 utf-8 编码字符 ,并以%开头 ;</li> </ul>                             比如原始的 URL :                              http://click.dsp.com?macroA=%A%&amp;macroB=##MACRO_B##&amp;macroC=(MacroC.Value)&amp;refer=\$REFER-URL                         </li> </ul>



	<p>会被转义为：</p> <p>http%3A%2F%2Fclick%2Edsp%2Ecom%3FmacroA%3D%25%25A%25%25%26macroB%3D%23%23MACRO%5FB%23%23%26macroC%3D%28MacroC%2EValue%29%26refer%3D%24REFER%2DURL</p>
%%CLICK_URL_ESC%%	<ul style="list-style-type: none"> <li>使用此宏时不需要填写 target_url 字段</li> <li>只需要返回一个宏即可，DSP 可以将创意的点击链接写为 http://dsp_click_url&amp;url=%%CLICK_URL_ESC%%landing_page_url（要求 landing_page_url 被 encode 两次）。BES 会构造 BES 的 click_url 并 encode 后替换宏。</li> <li>请注意：DSP 在广告渲染时，需要对 BES 的 click_url 进行 decode，并且所有创意的点击链接都要加入 BES 的 click_url，否则 BES 无法准确记录点击数据，会影响 BES 的报表提供和数据反作弊；使用 CLICK_URL_ESC 作为 click_url 宏时，html 代码中请勿出现 CLICK_URL_{N}宏。</li> </ul>

例如：如 html 中包含两个创意，

- 使用%%CLICK\_URL\_{N}%%时，则需要使用两个宏%%CLICK\_URL\_0%%和%%CLICK\_URL\_1%%。并在 target\_url 字段中顺序赋值。

```
<html><body>...<a href="%%CLICK_URL_0%%"/>..<a href="%%CLICK_URL_1%%"/>...</body></html>
```

- 使用%%CLICK\_URL\_ESC%%时，请参照如下方式：

```
<html><body>...<a href="http://click.dsp.com?some_params&&bes_url=%%CLICK_URL_ESC%%encoded_landing_page_url1"/>...<a href="http://click.dsp.com?some_params&&bes_url=%%CLICK_URL_ESC%%encoded_landing_page_url2"/>
```

当 DSP 返回 html\_snippet 时，必须同时设置如下信息：

creative\_id: 创意 id

advertiser\_id: 广告主 id

width：创意的宽

height：创意的高

category: 关键词所属行业

type: 创意的物料类型

landing\_page: 着陆页

target\_url: 点击链接 ( 当使用%%CLICK\_URL\_{N}%%宏时必填 )

max\_cpm: 关键词报价, 单位分

此外在投放 APP 插屏创意时, 必须添加关闭按钮, 由于百度无法关闭整个 iframe, 需要 DSP 来进行处理, 方法如下:

用 css 画一个关闭按钮, 点击关闭的时候, 百度调用了 sdk 的关闭接口, 也就是点击关闭按钮的时候, 调用 sdk 的 onAdPlayEnd 方法。

```
this.sdk = (window.baidu && window.baidu.mobads && window.baidu.mobads.Sdk)
|| (parent && parent.baidu && parent.baidu.mobads && parent.baidu.mobads.Sdk);
this.bind(closeBtn, 'click', this.sdk.onAdPlayEnd);
```

[5] keyword : 竞价链接单元广告的关键词, utf-8 编码。当 DSP 返回关键词竞价时, 必须同时设置如下信息:

creative\_id: 创意 id

selected\_style\_type: 关键词竞价的样式

advertiser\_id: 广告主 id

category: 关键词所属行业

type: 创意的物料类型

landing\_page: 着陆页

target\_url: 点击链接

monitor\_urls: 展示监测

max\_cpm: 关键词报价, 单位分

[6] selected\_style\_type :所竞价关键词所使用的样式。为空表示可以适用于 request 中的所有样式, 如果 DSP 对样式无特别要求, 建议为空, 这样可以提升关键词的胜出率。如果选择, 该样式的值必须是 bid request 中所列明的候选样式( style\_type )中的一种或多种, 不能超出所列明的样式。广告最终的样式由 BES 衡量多重因素从候选样式中选择的最优样式进行展现。

[7] advertiser\_id : 广告主 ID, 当返回 HTML 创意时必填。

[8] width : 创意的宽, 当返回 HTML 创意时必填, 此字段需要与请求中的[46]广告位宽度相对应。

[9] height : 创意的高, 当返回 HTML 创意时必填, 此字段需要与请求中的[47]广告位高度相对应。

[10] category : 创意所属的行业, 必须指定到第 2 级行业。广告分类体系见数据字典, 当返回 HTML 创意时必填。

[11] type : 创意类型, 当返回 HTML 创意时必填。

值	说明
0	文本
1	图片

2	FLASH
4	图文
5	链接单元 ( linkunit )
7	Video

[12] landing\_page : 到达页面, 当返回 HTML 创意时必须填。在 Html snippet 中创意的 LandingPage 必须在同一个域下。LandingPage 只需要提交第一个 “/” 前的域名即可。如 <http://www.abc.com/abc.html> 这样的只需要提交 <http://www.abc.com> 就可以。

[13] target\_url : 包含 302 跳转的完整点击链接。当返回 HTML 创意时, 如果使用 %%CLICK\_URL\_{N}%% 宏, 则必填; 如果使用 %%CLICK\_URL\_ESC%% 宏, 则不需要填写。使用方法说明如下:

- Html\_snippet 中含有多个创意的情况下, 每个创意点击链接的顺序应与 html snippet 中设置的创意的 %%CLICK\_URL\_{N}%% 的顺序一致。如该顺序不正确, 将引发点击的统计偏差。

接上例:

```
ad.add_target_url("http://click.buyer.com?idea=ad0...");
ad.add_target_url("http://click.buyer.com?idea=ad1...");
```

- 当返回关键词时, 只需要填写点击链接即可, 不需要添加点击宏。

点击链接目前支持的宏包括:

宏	说明
%%ID%%	请求 ID
%%EXT_DATA%%	DSP 提供的扩展参数
%%CREATIVE_ID%%	创意 ID
%%STYLE_TYPE%%	链接单元广告展现时的样式 ID
%%INDEX%%	关键词在链接单元广告中的排序

%%CREATIVE\_ID%%、%%STYLE\_TYPE%%、%%INDEX%% 宏仅关键词竞价中支持。

[14] monitor\_urls : 展示监测。

[15] creative\_width : 扩展创意的宽度。

[16] creative\_height : 扩展创意的高度。

[17] creative\_landing\_page : 扩展创意的落地页。

[18] creative\_type : 扩展创意的创意类型。

值	说明
0	文本
1	图片

2	FLASH
4	图文
5	链接单元 ( linkunit )
7	Video

[19] max\_cpm : DSP 提供的广告 CPM 最高竞价信息，当正常竞价时必填。

[20] extdata : DSP 提供的扩展参数，用于 DSP 自定义内容。会替换 win notice url 和 click url 中的宏%%EXT\_DATA%%。如果本次请求需要 cookie matching，该扩展参数会以 ext\_data=的形式附加在 cookie matching 的 url 中。

[21] is\_cookie\_matching : DSP 根据自身设置，表示是否需要进行 cookie matching。

[22] preferred\_order\_id : 如果选择优先交易，该交易的 ID。

[23] guaranteed\_order\_id : 如果为包断订单流量，该订单的 ID。

[24] deeplink\_url : 应用唤醒打开页面的地址。

[25] app\_version : 打开的应用的版本。

[26] fallback\_url : 应用唤醒退化链接的地址。

[27] fallback\_type : 应用唤醒退化链接的类型。

值	说明
1	LANDING_PAGE
2	DOWNLAOD

[28] ad : 如果放弃竞价，返回 0 个 ad。目前只支持一个广告位的竞价请求返回，也就是说目前每次最多返回一个广告。

[29] debug\_string : 联调测试用字段，utf8 编码。

[30] processing\_time\_ms : DSP 内部广告检索时间，单位毫秒。

#### 4、百度流量交易服务向 DSP 发送的竞价结果接口 ( Win Notice )

通过对展示监测链接中特定参数的宏替换，将广告的计费价格发送给赢得竞价的 DSP 平台。百度流量交易服务对竞标成功的 DSP 通过 HTTP GET 发送竞价成功通知。

目前支持的宏：

宏	说明
%%PRICE%%	广告成交价格
%%ID%%	请求 ID
%%EXT_DATA%%	DSP 提供的扩展参数
%%CREATIVE_ID%%	创意 ID
%%STYLE_TYPE%%	链接单元广告展现时的样式 ID
%%INDEX%%	关键词在链接单元广告中的排序

其中 DSP 的展现监测链接中必须加入%%PRICE%%、%%ID%%宏；

%%CREATIVE\_ID%%、%%STYLE\_TYPE%%、%%INDEX%%宏仅关键词竞价中支持。

URL Example：

```
http://www.dsp.com/monitor?reqid=%%ID%%&price=%%PRICE%%&ext_data=%%EXT_DATA%%
```

PRICE 基于 SHA-1 HMAC 算法进行加密，C++ 模块可以直接用 openssl 中的接口实现。DSP 注册时获得 32 字节的加密密钥，用于解密。获取 32 字节的完整性检测密钥，用于价格完整性检查。

## 二、Cookie Matching

Cookie Matching 建立了 Baidu User ID 与 DSP User ID 的映射关系，DSP 负责维护映射关系表。在 RTB 时 DSP 可以通过映射关系获取到 DSP User ID，然后通过 ID 关联的用户属性优化出价决策。

### 1、Cookie Matching 的准备工作

① DSP 需要提供 Cookie Matching 重定向服务器地址，如下例：

```
http://www.dsp.com/cm
```

② 百度提供的 Cookie Matching 重定向服务器代码，如下例：

```

```

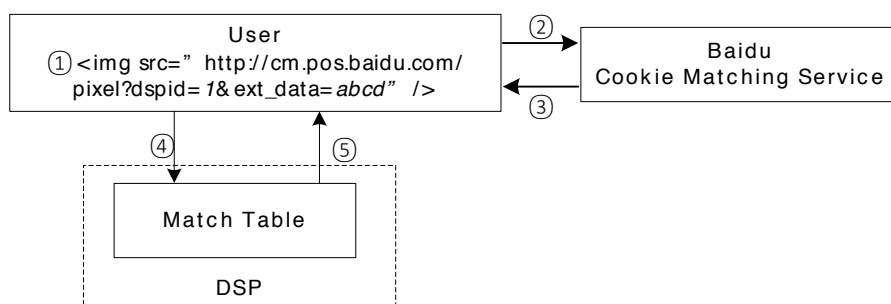
http://cm.pos.baidu.com/pixel 为 Baidu Cookie Matching 服务地址

说明：

- dspid 为 DSP 接入百度流量交易服务的唯一标识
- ext\_data 为 DSP 在 Bid Response 中提供的扩展参数。如果直接使用 cookie matching 服务，该参数可以不提供。

其中，dspid 必须提供，ext\_data 可以为空

### 2、Cookie Matching 流程



① DSP 发起 Cookie Matching 请求（可以通过自有资源将如下代码插入页面，或者通过 Bid Response 告诉百度流量交易服务本次请求需要 cookie matching，百度将如下代码插入广告页面），匹配标记：

```

```

② User 向 Baidu Cookie Matching 服务器发送 GET 请求

③ Baidu Cookie Matching 服务将请求重定向到 DSP 提供的 Cookie Matching 服务

④ 向 DSP Cookie Matching 服务发起请求

```
http://www.dsp.com/cm?baidu_user_id=ID&cookie_version=1&timestamp=TIMESTAMP
&ext_data=abcd
```

- baidu\_user\_id 表示 Baidu User ID
- cookie\_version 为 baidu user id 版本号

- timestamp 为了防止浏览器缓存，嵌入的时间戳
  - ext\_data DSP 在 Bid Response 中提供的扩展参数
- ⑤ DSP 生成 DSP User ID，将 Baidu User ID 与 DSP User ID 的映射关系保存到 Match Table 中，然后返回空白 1\*1 像素图片给用户。

### 3、错误处理

当 Cookie Matching 服务遇到错误，会发起如下重定向请求。

注意：在错误时，DSP 仍需将 1\*1 像素图片正常返回给用户。

`http://www.dsp.com/cm?baidu_error=1&timestamp=TIMESTAMP`

err=1 cookie matching 服务内部错误。

### 三、示例

示例使用 protocol buffer 序列化库 TextFormat(google/protobuf/text\_format.h)序列化而成。示例请见 baidu\_realtime\_bidding.proto 文件。



## 附录 1：扩展广告 SDK 脚本文档

### 1. SDK 使用方式

首先将 ext.js 引入用于广告展示的 iframe 内嵌页面中，

目前支持两个方法：

展开广告：

```
baidudup.expand("u12345", function (err) {  
    // "u12345" 表示需要展开的广告位 slotId  
    if (err) {  
        return; // 如果展开出现问题，则 err 参数则存在  
    }  
});
```

收起广告：

```
baidudup.close("u12345", function (err) {  
  
});
```

允许绑定三类事件：

当鼠标移动到广告容器上时：mouseover

当点击广告时：click

当广告加载完毕时：adloaded

绑定事件：

```
baidudup.on("click", function () {  
});
```

例如，当用户鼠标悬浮广告时展开广告，并且展开后 10 秒自动关闭：

```
baidudup.on('mouseover', function () {  
    baidudup.expand("u12345", function () {  
        setTimeout(function () {  
            baidudup.close("u12345", function () {});  
        }, 1000 * 10);  
    });  
});
```

## 2. SDK 实现原理

### (1) 方法调用:

通过 window.postMessage 方法向父窗口 ( window.top ) 广播消息, 消息格式为

```
var message = {  
  title: "baidudup", //父窗口通过鉴别消息 title 字段来判断是否来自 sdk  
  type: "invokeMethod", // 消息类型, invokeMethod 表示 API 调用  
  method: methodName, // 方法调用名称  
  parameters: parameters, // 调用方法参数  
  guid: guid // 消息唯一标识, 同时与调用该方法的回调函数关联  
};
```

父窗口调用指定 API 后, 将调用结果 ( 或者是否报错 ) 返回给 SDK, 数据结构为:

```
{  
  title: "baidudup", // 同上  
  type: "methodResponse", // 同上  
  result: result, // 方法调用后的结果  
  guid: guid // 保证后端能够找到对应的回调函数  
}
```

### (2) 事件机制

父页面向 SDK 广播事件发生, 数据结构:

```
{  
  title: "baidudup", // 同上  
  type: "eventHappen", // 同上  
  event: "mouseover", // 发生事件的事件名称  
  eventInfo: obj // 和事件相关的信息  
}
```

广告内嵌页面需要在 sdk 中注册事件处理函数

```
baidudup.addEventListener('mouseover', function (event) {  
  //Handler this event  
});
```