

目录

新义 SSP API 接口 2.0 协议	1
文档更新记录	1
对接地址	2
Request	2
protobuf 文件	2
Request 字段信息	7
Ad	7
App	8
Device	8
boot_mark 获取方法	10
update_mark 获取方法	11
device_startup_time 设备启动时间	12
system_update_time 系统更新时间	12
system_init_time 系统初始化时间	12
User	12
Response	13
Response 字段信息	13
Ad 对象	13
image	15
video	15
宏	15
点击坐标打点 click_area_report_url	16
响应中的click_area_report_url示例	16
价格解密	16
AES 测试用例	16
Double Click 测试用例	17
示例	17
请求	17
响应	18

新义 SSP API 接口 2.0 协议

文档更新记录

版本	作者	时间	备注
v2.0.0	崔英杰	2019.05.24	创建 SSP API 2.0 接口文档
v2.0.1	崔英杰	2019.10.17	增加 OAID 支持
v2.0.2	崔英杰	2019.12.11	增加 DeepLink trancckers
v2.0.3	崔英杰, 马秉尧	2020.04.02	增加 request id 和 response id
v2.0.4	马秉尧	2020.07.01	增加 rom_version, sys_compiling_time, ssid 和 wifi_mac
v2.0.5	崔英杰	2020.07.14	移除广点通下载类型
v2.0.6	崔英杰	2020.09.03	支持 oaid_md5
v2.0.7	马秉尧	2021.02.22	增加价格加密传输支持
v2.0.8	马秉尧	2021.03.08	支持 caid
v2.0.9	马秉尧	2021.08.19	增加 boot_mark、update_mark 字段
v2.0.10	马秉尧	2022.04.22	增加 caid_version 字段

版本	作者	时间	备注
v2.0.11	马秉尧	2023.04.12	增加了价格解密测试用例
v2.0.12	马秉尧	2023.04.17	增加了逻辑像素点击宏
v2.0.13	马秉尧	2023.08.31	增加了 privacy_url、permission_url、mini_program_id、mini_program_path 响应字段
v2.0.14	马秉尧	2023.11.14	增加了 paid 请求字段
v2.0.15	马秉尧	2024.03.19	增加了 download_app_desc 响应字段
v2.0.16	戚银	2024.05.17	增加了 device_startup_time、system_update_time、system_init_time 字段
v2.0.17	张勇	2024.05.31	增加了 caid_md5 caid2 caid2_md5 caid2_version 请求字段
v2.0.18	马秉尧	2024.06.24	增加了 ipv6 请求字段
v2.0.19	马秉尧	2024.08.26	增加了 click_area_report_url 响应字段
v2.0.20	马秉尧	2025.03.06	删除了 ratings, likes, downloads 响应字段，修正 download_app_size 描述中的单位
v2.0.21	马秉尧	2025.03.07	增加了 protobuf 支持，增加压缩支持说明

对接地址

url: [https://api.mobrtb.com/ad/xy/\\$媒体token](https://api.mobrtb.com/ad/xy/$媒体token)

其中 `$媒体token` 是一个 32 字节的字符串，例如：`BA2E26E8C87C936B29B58C1A918F5E6D`。该字符串在媒体创建媒体账号之后，在账户->个人信息中可以得到。

Request

方法	格式	编码
HTTP POST	JSON 或 protobuf	UTF-8

请求如果是使用 protobuf 格式，请求的 HTTP header 中的 `Content-Type` 一定要设置为：`application/x-protobuf`，否则，会默认按照 JSON 格式解析请求。

请求和响应均支持压缩，请求数据如果是压缩格式的，请求的 HTTP header 中需要带 `Content-Encoding` 头，值为压缩格式，压缩格式支持：`zstd`，`gzip`，`br`，`compress`，`deflate`。响应如果想要返回压缩格式的数据，请求的 HTTP header 中需要带 `Accept-Encoding` 头，值为支持的压缩格式，如果支持多种压缩格式，返回按照 `Accept-Encoding` 中列举的压缩格式的顺序，取第一个支持的压缩格式返回，实际响应的压缩格式，会在响应的 `Content-Encoding` 头中标明。

protobuf 文件

```
syntax = "proto3";
option go_package = "./xyssp";
package xyssp;

message BidRequest {
```

```
// 请求 id, 每个请求唯一
string id = 1;
// 协议版本, 当前协议版本 2.0.0
string version = 2;
// Ad 数组, 目前只支持传一个 Ad 对象
repeated Ad ads = 3;
// App 对象
App app = 4;
// Device 对象
Device device = 5;
// User 对象
User user = 6;
// 是否需要 https 链接, 默认不需要
bool need_https = 7;
message Ad {
    // 广告位 token
    string ad_unit_token = 1;
    // 广告位宽度
    int64 width = 2;
    // 广告位高度
    int64 height = 3;
    // 底价, 单位为分
    double floor_price = 4;
    // 是否支持将 html 广告放在 webview 里面展示, 默认不支持
    bool support_js = 5;
}
message App {
    // 应用名
    string name = 1;
    // 应用包名
    string bundle = 2;
    // 应用版本
    string version = 3;
    // -1: 不支持 deeplink
    // 0: 在浏览器打开的目标页面中触发 deeplink (默认)
    // 1: 表示可在广告页面触发 deeplink, 失败则打开 target_url
    int64 deeplink_mode = 4;
}
enum DeviceType {
    UNKNOWN = 0; //未知
    PHONE = 1;   //手机
    PAD = 2;     //平板
    TV = 3;      //电视
    PC = 4;      //电脑
}
message Device {
    // IP 地址
    string ip = 1;
    // IPv6 地址
    string ipv6 = 2;
    // WebView User-Agent
    string user_agent = 3;
    // 生产厂商, 例如:"Samsung"
    string make = 4;
    // 手机品牌, 例如:"MI4"
    string brand = 5;
    // 手机型号
    string model = 6;
    // 操作系统, 取值范围: "android", "ios"
    string os = 7;
    // 操作系统版本
```

```
string os_version = 8;
// 网络连接类型, 取值范围: "wifi", "2g", "3g", "4g", "5g"
string connection_type = 9;
// 屏幕方向, 取值范围: "landscape", "portrait"
string orientation = 10;
// 移动网络运营商, 取值范围: "46000", "46002", "46001", "46003"
// 46000: 中国移动
// 46002: 中国移动
// 46001: 中国联通
// 46003: 中国电信
string plmn = 11;
// Mac 地址
string mac = 12;
// Mac 地址的 md5 值
string mac_md5 = 13;
// 系统语言
string language = 14;
// 屏幕宽度, 单位: 像素
int64 screen_width = 15;
// 屏幕高度, 单位: 像素
int64 screen_height = 16;
// 屏幕像素密度
int64 screen_dpi = 17;
// 屏幕物理像素密度
double screen_pxratio = 18;
// 地理位置经度
double geo_longitude = 19;
// 地理位置纬度
double geo_latitude = 20;
// 用户设备安装包名列表, 用逗号分隔
string installed_apps = 21;
// 用户设备类型
DeviceType device_type = 22;
// 无线网 SSID
string ssid = 23;
// WIFI 路由器 Mac 地址
string wifi_mac = 24;
// 设备 IMEI, 不传会影响填充, CDMA 手机请传 MEID 码
string imei = 25;
// 设备 IMEI 的 MD5 值, IMEI 获取不到时请传这个字段
string imei_md5 = 26;
// Android ID
string android_id = 27;
// Android ID 的 md5 值
string android_id_md5 = 28;
// 设备 IMSI
string imsi = 29;
// Android 广告 ID
string android_advertising_id = 30;
// 匿名设备标识符 OAID
// 对于 Android 设备, 当 oaid 不填时, imei(或 imei_md5) 和 android_id(或
// android_id_md5) 必填
string oaid = 31;
// OAID 的 md5 值
string oaid_md5 = 32;
// 手机 ROM 的版本号 (例如 MIUI 11.0.6, EMUI 10.1 等)
string rom_version = 33;
// 系统编译时间 (时间戳)
int64 sys_compiling_time = 34;
// IDFA, 不传会影响填充
string idfa = 35;
```

```
// IDFA 的 md5 值, IDFA 获取不到时请传这个字段
string idfa_md5 = 36;
// IDfv
string idfv = 37;
// 中广协推出的 iOS 设备 ID, 支持同时传两个版本
string caid = 38;
// CAID 的 md5 值
string caid_md5 = 39;
// CAID 的版本号
string caid_version = 40;
string caid2 = 41;
string caid2_md5 = 42;
string caid2_version = 43;
string openudid = 44;
string boot_mark = 45;
string update_mark = 46;
// 拼多多设备指纹
string paid = 47;
// 设备启动时间
string device_startup_time = 48;
// 系统更新时间
string system_update_time = 49;
// 系统初始化时间
string system_init_time = 50;
}
message User {
    // 年龄
    int64 age = 1;
    // 性别, 取值范围: "男", "女", "未知"
    string gender = 2;
    // 用户感兴趣关键字
    repeated string keywords = 3;
}
}

message BidResponse {
    // 回填 Request 中的 id
    string id = 1;
    // Ad 对象数组, 目前只支持返回一个 Ad 对象
    repeated Ad ads = 2;
    message Ad {
        // 广告宽度
        int64 width = 1;
        // 广告高度
        int64 height = 2;
        // 广告 ID
        string ad_id = 3;
        // 素材 ID
        string creative_id = 4;
        // 广告价格, 单位 分
        double price = 5;

        // 标题
        string title = 6;
        // 副标题
        string subtitle = 7;
        // 描述
        string description = 8;
        // 广告主名称
        string advertiser_name = 9;
        // 按钮文字
```

```
string button_text = 10;
// 图片数组
repeated Image images = 11;
// 图标
Image icon = 12;
// Logo
Image logo = 13;
// 视频
Video video = 14;
// 视频封面图
Image video_cover = 15;
// html 代码片段, 在 webview 中加载
string html_snippet = 16;
// html 代码地址, 在 webview 中加载
string html_url = 17;

// 广告动作类型
// 1: 在应用内用 webview 打开 target_url
// 2: 在系统浏览器打开 target_url
// 6: 下载应用
// 7: 打开 deeplink_url
// 8: 打开小程序
int64 action = 18;
// 目标地址, 里面的宏需要替换
string target_url = 19;
// 下载应用包名
string download_app_bundle = 20;
// 下载应用名称
string download_app_name = 21;
// 下载应用版本
string download_app_version = 22;
// 下载应用大小, 单位 Bytes
int64 download_app_size = 23;
// 下载应用描述
string download_app_desc = 24;
// 下载应用隐私协议地址
string privacy_url = 25;
// 下载应用权限列表地址
string permission_url = 26;
// 小程序原始id
string mini_program_id = 27;
// 小程序页面路径
string mini_program_path = 28;
// Deeplink 或 Universal Link, 里面的宏需要替换
string deeplink_url = 29;

// 竞价获胜上报地址
string win_notice_tracker = 30;
// 曝光上报地址, 里面的宏需要替换
repeated string impression_trackers = 31;
// 点击上报地址, 里面的宏需要替换
repeated string click_trackers = 32;
// 应用下载开始上报地址, 里面的宏需要替换
repeated string download_begin_trackers = 33;
// 应用下载完成上报地址, 里面的宏需要替换
repeated string download_ended_trackers = 34;
// 应用下载完成上报地址, 里面的宏需要替换
repeated string install_ended_trackers = 35;
// 视频播放开始上报地址, 里面的宏需要替换
repeated string video_play_begin_trackers = 36;
// 视频播放中止上报地址, 里面的宏需要替换
```

```

repeated string video_play_break_trackers = 37;
// 视频播放结束上报地址, 里面的宏需要替换
repeated string video_play_ended_trackers = 38;
// deeplink 唤醒时, 发现应用没有安装的上报地址
repeated string deeplink_app_not_installed_trackers = 39;
// deeplink 唤醒时, 发现应用有安装的上报地址
repeated string deeplink_app_installed_trackers = 40;
// deeplink 唤醒时, 应用有安装, 但唤起失败的上报地址
repeated string deeplink_app_invoke_failed_trackers = 41;
// deeplink 唤醒时, 应用唤起成功上报地址
repeated string deeplink_app_invoke_success_trackers = 42;
// 汇川预算 点击坐标打点 (具体细节, 参考文档)
string click_area_report_url = 43;
}
message Image {
// 图片地址
string url = 1;
// 图片宽度
int64 width = 2;
// 图片高度
int64 height = 3;
}
message Video {
// 视频地址
string url = 1;
// 视频时长, 单位 秒
int64 duration = 2;
}
}

```

Request 字段信息

字段名称	类型	必须	描述
id	string	否	请求 id, 每个请求唯一
version	string	是	协议版本, 当前协议版本 2.0.0
ads	[]Ad	是	Ad 数组
app	App	是	App
device	Device	是	Device
user	User	否	User
need_https	bool	否	是否需要 https 链接, 默认不需要

Ad

字段名称	类型	必须	描述
ad_unit_token	string	是	广告位 token, 我们提供测试广告位 token 列表供测试用
width	int	是	广告位宽度
height	int	是	广告位高度
floor_price	float	否	底价, 单位为分

字段名称	类型	必须	描述
support_js	bool	否	是否支持将 html 广告放在 webview 里面展示, 默认不支持

App

字段名称	类型	必须	描述
name	string	是	应用名
bundle	string	是	应用包名
version	string	否	应用版本
deeplink_mode	int	否	- -1 : 不支持 deeplink - 0 : 在浏览器打开的目标页面中触发 deeplink (默认) - 1 : 表示可在广告页面触发 deeplink, 失败则打开 target_url

Device

字段名称	类型	必须	描述
ip	string	是	ip
ipv6	string	否	ipv6
user_agent	string	是	webview user agent
make	string	是	生产厂商, 例如: “Samsung”
brand	string	是	手机品牌, 例如: “MI4”
model	string	是	型号
os	string	是	操作系统 - android - ios
os_version	string	是	操作系统版本
connection_type	string	是	网络连接类型 - wifi - 2g - 3g - 4g - 5g
orientation	string	是	设备方向 - landscape

字段名称	类型	必须	描述
plmn	string	否	- portrait 国家运营商编号 - 46000 : 中国移动 - 46002 : 中国移动 - 46001 : 中国联通 - 46003 : 中国电信
mac	string	否	MAC 地址
mac_md5	string	否	mac 获取不到时请传这个字段
language	string	否	系统语言
screen_width	int	否	屏幕宽度, 单位: 像素
screen_height	int	否	屏幕高度, 单位: 像素
screen_dpi	int	否	屏幕像素密度, 参考
screen_pxratio	float	否	屏幕物理像素密度, 参考
geo_longitude	float	否	经度
geo_latitude	float	否	纬度
installed_apps	string	否	用户设备安装包名列表, 用逗号分隔 com.android.calendar,com.android.chrome
device_type	int	否	用户设备类型 - 0 : 未知 - 1 : 手机 - 2 : 平板 - 3 : 电视 - 4 : PC
ssid	string	否	无线网 SSID
wifi_mac	string	否	WIFI 路由器 Mac 地址
以下字段仅在 Android 系统上可获取到			
imei	string	否	不传会影响填充 cdma 手机请传 meid 码
imei_md5	string	否	imei 获取不到时请传这个字段
android_id	string	否	
android_id_md5	string	否	android_id 获取不到时请传这个字段
imsi	string	否	
android_advertising_id	string	否	

字段名称	类型	必须	描述
oaid	string	否	匿名设备标识符。 对于 Android 设备，当 oaid 不填时， imei(或 imei_md5) 和 android_id(或 android_id_md5) 必填
oaid_md5	string	否	oaid md5
rom_version	string	否	手机 ROM 的版本号（例如 MIUI 11.0.6, EMUI 10.1 等）
sys_compiling_time	long	否	系统编译时间（时间戳）
以下字段仅在 iOS 系统上可获取到			
idfa	string	否	不传会影响填充
idfa_md5	string	否	idfa 获取不到时请传这个字段
idfv	string	否	
caid	string	否	中广协推出的iOS设备ID，同时支持传两个版本，详情 http://www.china-caa.org/digital/caid/intro
caid_md5	string	否	中广协推出的iOS设备ID MD5，详情 http://www.china-caa.org/digital/caid/intro
caid_version	string	否	中广协推出的iOS设备ID 的版本号，详情 http://www.china-caa.org/digital/caid/intro
caid2	string	否	中广协推出的iOS设备ID，详情 http://www.china-caa.org/digital/caid/intro
caid2_md5	string	否	中广协推出的iOS设备ID MD5，详情 http://www.china-caa.org/digital/caid/intro
caid2_version	string	否	中广协推出的iOS设备ID 的版本号，详情 http://www.china-caa.org/digital/caid/intro
openudid	string	否	
boot_mark	string	否	参考 boot_mark 获取方法
update_mark	string	否	参考 update_mark 获取方法
paid	string	否	拼多多设备指纹
device_startup_time	string	否	设备启动时间
system_update_time	string	否	系统更新时间
system_init_time	string	否	系统初始化时间

boot_mark 获取方法

iOS参考代码：

```
- (NSString *)getBoot
{
    NSString *timeString=nil;
    int mib[MIB_SIZE];
    size_t size;
    struct timeval boottime;
    mib[0] = CTL_KERN;
```

```

    mib[1] = KERN_BOOTTIME;
    size = sizeof(boottime);
    if (sysctl(mib, MIB_SIZE, &boottime, &size, NULL, 0) != -1)
    {
        timeString = [NSString stringWithFormat:@"%d.%d", (int) boottime.tv_sec, (int) boottime.tv_usec];
    }
    return timeString?:@"";
}

```

Android 参考代码:

```

//获取boot
static jstring getBoot(JNIEnv *env, jclass) {
    FILE *fp = fopen("/proc/sys/kernel/random/boot_id", "r");
    const int TID1_LEN = 37;
    char boot[TID1_LEN];
    if (fp == NULL) {
        //获取失败
    } else {
        unsigned char c;
        int i = 0;
        while (i < TID1_LEN) {
            c = fgetc(fp);
            boot[i] = c;
            i = i + 1;
        }
        if (ferror(fp)) {
            //获取失败
        }
    }
    // fgets(boot,sizeof(boot),fp);
    std::string sboot = boot;
    return env->NewStringUTF(sboot.c_str());
}

```

update_mark 获取方法

iOS参考代码:

```

- (NSString *)getUpdate
{
    NSString *timeString = nil;
    struct stat sb;
    NSString *enCodePath = @"L3Zhci9tb2JpbGU=";
    NSData *data=[[NSData alloc] initWithBase64EncodedString:enCodePath options:0];
    NSString *dataString = [[NSString alloc] initWithData:data encoding:NSUTF8StringEncoding];
    const char* dePath = [dataString cStringUsingEncoding:NSUTF8StringEncoding];
    if (stat(dePath, &sb) != -1) {
        timeString = [NSString stringWithFormat:@"%d.%d",
            (int)sb.st_ctimespec.tv_sec, (int)sb.st_ctimespec.tv_nsec];
    } else {
        timeString = @"0.0";
    }
    return timeString?:@"";
}

```

Android 参考代码:

```

//获取update
static jstring getUpdate(JNIEnv *env, jclass) {
    struct stat sb;
    int updates, updatens;
    if (stat("/data/data", &sb) == -1) {

```

```

        //获取失败
    } else {
        updatens = (int) sb.st_atim.tv_nsec;
        updates = (int) sb.st_atim.tv_sec;
    }
    std::string idRes = std::to_string(updates) + "." + std::to_string(updatens);
    return env->NewStringUTF(idRes.c_str());
}

```

device_startup_time 设备启动时间

```

static time_t bootSecTime() {
    struct timeval boottime;
    size_t len = sizeof(boottime);
    int mib[2] = { CTL_KERN, KERN_BOOTTIME };
    if( sysctl(mib, 2, &boottime, &len, NULL, 0) < 0 )
    {
        return 0;
    }
    return boottime.tv_sec;
}
+ (NSString *) bootTimeInSec {
    return [NSString stringWithFormat:@"%ld",bootSecTime()];
}

```

system_update_time 系统更新时间

```

+ (NSString *) getSysU {
    NSString *result = nil;
    NSString *information = @"L3Zhci9tb2JpbGUvTGlicmFyeS9Vc2VyQ29uZmJndXJhdGlvbGByb2ZpbGVzL1B1YmV0luZm8vTUNN";
    NSData *data=[[NSData alloc] initWithBase64EncodedString:information options:0];
    NSString *dataString = [[NSString alloc] initWithData:data encoding:NSUTF8StringEncoding];
    NSError *error = nil;
    NSDictionary *fileAttributes = [[NSFileManager defaultManager] attributesOfItemAtPath:dataString error:&error];
    if (fileAttributes) {
        id singleAttribute = [fileAttributes objectForKey:NSFileCreationDate];
        if ([singleAttribute isKindOfClass:[NSDate class]]) {
            NSDate *dataDate = singleAttribute;
            result = [NSString stringWithFormat:@"%f",[dataDate timeIntervalSince1970]];
        }
    }
    return result;
}

```

system_init_time 系统初始化时间

```

static NSString *getFileTime() {
    struct stat info;
    int result = stat("/var/mobile", &info);
    if (result != 0) {
        return @"";
    }
    struct timespec time = info.st_birthtimespec;
    return [NSString stringWithFormat:@"%ld.%09ld", time.tv_sec, time.tv_nsec];
}

```

User

字段名称	类型	必须	描述
age	int	否	年龄
gender	string	否	性别 男 女 未知
keywords	[]string	否	用户感兴趣关键字

Response

http status code	描述
200	正常, 有广告返回
204	正常, 无广告返回
400	参数错误, response body 里面有错误原因, 请复制出来给我们解释
404	广告位 token 错误, 找不到相应广告位

Response 字段信息

字段名称	类型	必须	描述
id	string	否	回填 Request 中的 id
ads	[]Ad	是	Ad 对象数组

Ad 对象

字段名称	类型	必须	描述
width	int	是	广告宽度
height	int	是	广告高度
ad_id	string	是	广告 id
creative_id	string	是	素材 id
price	float	否	广告价格, 单位 分
用于广告展示的素材			
title	string	否	标题
subtitle	string	否	副标题
description	string	否	描述
advertiser_name	string	否	广告主名称
button_text	string	否	按钮文字
images	[]image	否	图片数组

字段名称	类型	必须	描述
icon	image	否	图标
logo	image	否	logo
video	video	否	视频
video_cover	image	否	视频封面图
html_snippet	string	否	html 代码片段, 在 webview 中加载, 里面的宏需要替换
html_url	string	否	html 代码地址, 在 webview 中加载, 里面的宏需要替换
用于广告点击处理			
action	int	是	广告动作类型 - 1 : 在应用内用 webview 打开 target_url - 2 : 在系统浏览器打开 target_url - 6 : 下载应用 - 7 : 打开 deeplink_url, 如果失败则打开 target_url - 8 : 打开小程序
target_url	string	是	目标地址, 里面的宏需要替换
download_app_bundle	string	否	下载应用包名
download_app_name	string	否	下载应用名称
download_app_version	string	否	下载应用版本
download_app_size	int	否	下载应用大小, 单位 Bytes
download_app_desc	string	否	下载应用描述
privacy_url	string	否	下载应用隐私协议地址
permission_url	string	否	下载应用权限列表地址
mini_program_id	string	否	小程序原始id
mini_program_path	string	否	小程序页面路径
deeplink_url	string	否	里面的宏需要替换
上报地址			
win_notice_tracker	string	是	竞价获胜
impression_trackers	[]string	是	曝光, 里面的宏需要替换
click_trackers	[]string	是	点击, 里面的宏需要替换
download_begin_trackers	[]string	否	应用下载开始, 里面的宏需要替换
download_ended_trackers	[]string	否	应用下载完成, 里面的宏需要替换
install_begin_trackers	[]string	否	应用安装开始, 里面的宏需要替换
install_ended_trackers	[]string	否	应用安装完成, 里面的宏需要替换

字段名称	类型	必须	描述
video_play_begin_trackers	[]string	否	视频播放开始, 里面的宏需要替换
video_play_break_trackers	[]string	否	视频播放中止, 里面的宏需要替换
video_play_ended_trackers	[]string	否	视频播放结束, 里面的宏需要替换
deeplink_app_not_installed_trackers	[]string	否	deeplink 唤醒时, 发现应用没有安装
deeplink_app_installed_trackers	[]string	否	deeplink 唤醒时, 发现应用有安装
deeplink_app_invoke_failed_trackers	[]string	否	deeplink 唤醒时, 应用有安装, 但唤起失败
deeplink_app_invoke_success_trackers	[]string	否	deeplink 唤醒时, 应用唤起成功
click_area_report_url	string	否	汇川预算 点击坐标打点 click_area_report_url

image

字段名称	类型	必须	描述
url	string	是	图片地址
width	int	否	图片宽度
height	int	否	图片高度

video

字段名称	类型	必须	描述
url	string	是	视频地址
duration	int	否	视频时长, 单位 秒

宏

字段	含义
{XY_CLICK_DOWN_X}	手指点击时相对于素材的 X 坐标（物理像素）, 能替换则替换, 不能替换请全部替换为-999
{XY_CLICK_DOWN_Y}	手指点击时相对于素材的 Y 坐标（物理像素）, 能替换则替换, 不能替换请全部替换为-999
{XY_CLICK_UP_X}	手指抬起时相对于素材的 X 坐标（物理像素）, 能替换则替换, 不能替换请全部替换为-999
{XY_CLICK_UP_Y}	手指抬起时相对于素材的 Y 坐标（物理像素）, 能替换则替换, 不能替换请全部替换为-999
{XY_CLICK_DOWN_LX}	手指点击时相对于素材的 X 坐标（逻辑像素）, 能替换则替换, 不能替换请全部替换为-999
{XY_CLICK_DOWN_LY}	手指点击时相对于素材的 Y 坐标（逻辑像素）, 能替换则替换, 不能替换请全部替换为-999
{XY_CLICK_UP_LX}	手指抬起时相对于素材的 X 坐标（逻辑像素）, 能替换则替换, 不能替换请全部替换为-999
{XY_CLICK_UP_LY}	手指抬起时相对于素材的 Y 坐标（逻辑像素）, 能替换则替换, 不能替换请全部替换为-999

字段	含义
{XY_PRICE}	结算价格, 单位 分

备注：逻辑像素(dp) = 物理像素(px) / 屏幕密度(dpi)，使用逻辑像素作为单位可消除 Android 端不同分辨率手机屏幕导致的坐标差异，逻辑像素 Android 端需使用 dp 作为单位，iOS 端需使用 pt 作为单位。

点击坐标打点 click_area_report_url

1. 点击坐标反馈：该接口用于汇川广告在媒体侧产生点击事件时，由媒体回传点击坐标信息，用于反作弊。
2. 触发条件：如汇川广告在媒体侧产生点击事件，则构造post请求，并使用规定json串为body
3. 业务流程：
 1. 汇川检索在click_area_report_url上追加sid和creative_id，并下发给媒体
 2. 汇川广告在媒体侧产生点击事件时，媒体构造post请求，body为json串：
 1. 客户端将click_area_report_url中的sid和creative_id解析后追加到json串中的“sid”和“creative_id”
 2. 追加相应的点击元素信息到“click_element”，如点击标题click_element:title、点击图片click_element:image等。
 3. json串kv均为字符串形式
 3. 请注意区分sld 和 sid：前者为点击方式，后者为汇川侧请求id
 4. 不强制要求客户端上报，可以由媒体服务端拼body转发
4. 点击坐标说明及示例：
 1. ios端单位为物理像素px；android端单位为逻辑像素dp，逻辑像素dp=物理像素px/屏幕密度dpi；
 2. 原点为广告位左上角，坐标是相对于实际广告位（即完整广告容器）左上角的坐标；

响应中的click_area_report_url示例

```
"click_area_report_url": "https://test.huichuan.sm.cn/redirect/click_area?sid=12345&creative_id=456",
//拼接参数示例
{
  sld:必填，string，实际的点击方式，枚举值：0 - 常规触屏点击；1 - 滑动点击；2 - 摇一摇；3 - 自定义手势；4 - 点击或滑动；5 - 扭一扭；6 - 擦除触发点击；7 - 横竖屏转换。仅sld=0,1,3,4,6时，需要媒体上报点击坐标；
  width:必填，string，实际广告位的宽，单位为像素；
  height:必填，string，实际广告位的高，单位为像素；
  down_x:必填，string，用户手指按下时的横坐标；
  down_y:必填，string，用户手指按下时的纵坐标；
  up_x:必填，string，用户手指离开屏幕时的横坐标；
  up_y:必填，string，用户手指离开屏幕时的纵坐标；
  click_element:string，点击元素信息，如标题等。枚举值：title - 标题、sub_title - 副标题、author_icon - 广告头像、button - 按钮、image - 图片、video - 视频、other - 其他；
  sid:必填，string，媒体端上从click_area_report_url中解析的sid；
  creative_id:必填，string，媒体端上从click_area_report_url中解析的creative_id；
}
```

价格解密

价格宏的加密是可选的，可以直接传数字明文。

如果需要加密传输，加密算法可以采用 AES/ECB/PKCS5 算法，加密后的价格以 RawURLEncoding (un-padded alternate base64) 编码传递。所需秘钥在对接时由我方单独提供。

AES 测试用例

密钥	加密前数据	加密后数据
123456789abcdefghijklmnopqrstuvwxyz	100	agFVCc6ZpMRQGW8-mUtzRA
123456789abcdefghijklmnopqrstuvwxyz	500	8RNzQbVj6VvM0a_hRuzy3w
123456789abcdefghijklmnopqrstuvwxyz	1001	Kiiv5UT0xLVha19mPLKT6g

加密算法也可以采取 Google 的 Double Click 加密算法，加密方式链接：<https://developers.google.com/authorized-buyers/rtb/response-guide/decrypt-price#encryption-scheme>，所需密钥 e_key, i_key 在对接时由我方单独提供。

Double Click 测试用例

e_key	i_key	加密前数据	加密后数据
8f1dd415a672c54c1dd295201cb630a4b74ad404e5c8ba961ec009af01	100		AAABh3NrNQsw6ra4kzTreGX0UjS-qtQVwK7w-v
8f1dd415a672c54c1dd295201cb630a4b74ad404e5c8ba961ec009af01	500		AAABh3NrNQsTzyTNN1G42Wbwpreesy63ZPSU00
8f1dd415a672c54c1dd295201cb630a4b74ad404e5c8ba961ec009af01	1001		AAABh3NrNQsJm4-5rwyTYPED8M4B_TIERhj7Jv

注意：因为 Google 的 Double Click 加密算法中包含一个由时间戳和随机数的 iv，所以每次加密后的数据都是不一样的，验证算法实现是否正确时，只需要能够将加密后的数据解密成加密前的数据即可。

示例

请求

```
{
  "id": "bptcvhm8cv6t0nsoh6eg",
  "version": "2.0.0",
  "ads": [
    {
      "ad_unit_token": "209A03F87BA3B4EB82BEC9E5F8B41383",
      "width": 640,
      "height": 100,
      "support_js": true
    }
  ],
  "app": {
    "name": "今日头条",
    "version": "3.0.1",
    "bundle": "com.xinyi.toutiao",
    "deeplink_mode": 1
  },
  "device": {
    "ip": "114.251.228.90",
    "user_agent": "Mozilla/5.0 (Linux; Android 8.0.0; MIX 2 Build/OPR1.170623.027; wv)
    ↪ AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/70.0.3538.110 Mobile Safari/
    ↪ 537.36 News/195 Android/8.0.0 NewsApp/195 SDK/26 VERSION/7.1.5dongqiudiClientApp",
    "make": "Xiaomi",
    "brand": "Xiaomi",
  }
}
```

```

    "model": "MIX 2",
    "os": "android",
    "os_version": "7.0.1",
    "connection_type": "wifi",
    "orientation": "portrait",
    "plmn": "46000",
    "imei": "865736038728823",
    "android_id": "1f45138911dba981"
  }
}

```

响应

```

{
  "id": "bptcvhm8cv6t0nsoh6eg",
  "ads": [
    {
      "width": 640,
      "height": 100,
      "ad_id": "unique-ad-id",
      "creative_id": "unique-creative-id",
      "images": [
        {
          "url": "https://cdn.mobrtb.com/images/back_ad/banner/640_100/640-100-5.png"
        }
      ],
      "action": 7,
      "target_url": "https://m.cudaojia.com?
        ↪ appKey-d7950cd55bb34a1a89e1e09ab776a3bf&appType-h5&appEntrance-7&business-money",
      "deeplink_url": "tbopen://m.taobao.com/tbopen/index.html?
        ↪ source-auto&action-ali.open.nav&module-h5&bootImage-0&h5Url-
        ↪ https%3A%2F%2Fh5.m.taobao.com%2Fbcec%2Fdahanghai-
        ↪ jump.html%3Fspm%3D2014.ugdhh.3915747229.1208-
        ↪ 293%26bc_fl_srcgrowth_dhh_3915747229_1208-293&spm-2014.ugdhh.3915747229.1208-
        ↪ 293&bc_fl_src-growth_dhh_3915747229_1208-293&materialid-1208",
      "impression_trackers": [
        "https://api.mobrtb.com/tracker?request_id-bq3fm5aghlvo0f9abrd0&log_event_type-7"
      ],
      "click_trackers": [
        "https://api.mobrtb.com/tracker?request_id-bq3fm5aghlvo0f9abrd0&log_event_type-8"
      ]
    }
  ]
}

```