

# API-JSON 对接说明和规范

( 适用 API v5.5.0 )

## 1 API 接入技术说明

- 本规范需结合 API 接口文件 ( ksmobads\_api\_5.5.0 接口协议 ) , 请注意各字段的数据类型及取值
- 服务器地址 ( 线上 : <http://120.92.16.234/api/json> , 测试 : <http://120.92.44.245/api/json> )
- API 采用 json 对请求与返回内容进行封装。

协议	HTTP 协议, 支持 POST 方法
content-type	application/json
charset	UTF-8

## 2 广告请求字段说明

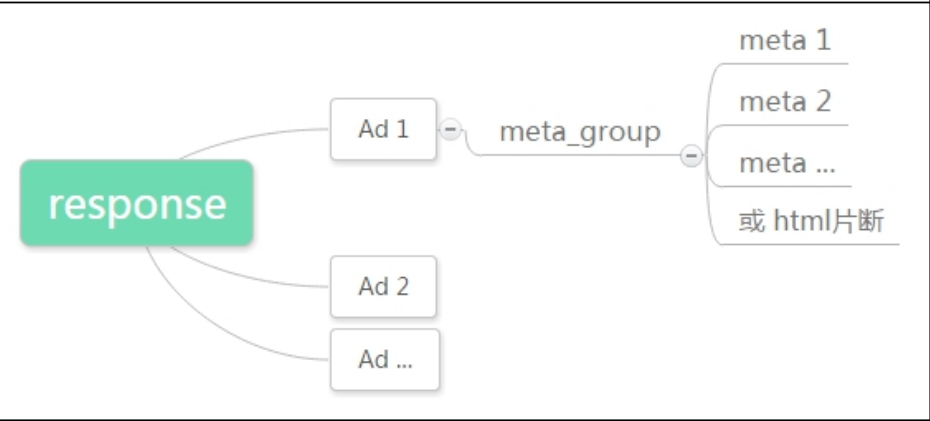
广告请求 ( MobadsRequest ) 参数是对一次请求的详细描述, 回传参数的丰富和准确直接影响流量变现效果。

参数	是否必填	参数定义	参数填写说明
<strong>1.1 基础参数</strong>			
request_id	必填	广告请求 ID	由接入方自定义, 在发起请求时填写, 需确保 request_id 32 位且唯一。推荐使用 app_id+adslot_id+若干位随机 Key 构成
api_version	必填	API 接口版本号	按照当前接入所参照的 API 文档版本赋值, 影响所有后续逻辑。当前版本 5.5.0, 填写错误将导致请求错误
channel_id	选填	发布渠道 ID	渠道接入方必填。渠道聚合等涉及三方分成, 需要填写渠道 ID。渠道 ID 获取请联系业务经理。独立开发者不填
<strong>1.2 应用参数</strong> : 需保证广告请求的应用与 MSSP/Union 登记的应用信息一致			
app_id	必填	应用 ID	在 Mobile SSP ( 以下简称 MSSP ) 提交应用, 可获得应用 ID。此 ID 格式样例为: 12abe6c0
app_version	必填	应用版本	格式参照 api_verison。请正确填写, 影响优选策略
app_package	选填	应用包名	应用包名, 确保与提交的应用一一对应, 建议媒体方填写
<strong>1.3 代码位参数</strong> : 代码位创建和设置在 MSSP 完成			
adslot_id	必填	代码位 id	代码位在 MSSP 进行设置, 平台提供各种属性选择; 设置实时生效, 请谨慎操作。
adslot_size	必填	代码位尺寸	用于填写当前设备可展现广告区域的尺寸。说明: 如代码位请求图片物料, 图片尺寸是在 MSSP 进行设置。
adslot_type	必填	广告位类型	用于填写对应广告位类型。说明: Banner=1, 插屏=2, 开屏=4, 原生=8, 奖励视频=9
ads	必填	广告数量	返回广告条数, 目前仅支持信息流。
video	选填	视频贴片	以 VAST 标准接入视频贴片, 必填。包括视频标准、长度、版权。

1.4 设备参数：获取并回传正确的设备信息，有帮变现效果			
Device type	必填	设备类型	描述设备分类的可选值集合。包括手机（含 iTouch）和平板
OS	必填	操作系统	描述操作系统的可选值集合。包括 Android 和 iOS
os_version	必填	操作系统版本	格式参照 api_verison，注意至少需要填写主版本号 major 和副版本号 minor
vendor	必填	设备厂商	设备厂商名称，中文需要 UTF-8 编码。样例：MEIXU
model	必填	设备型号	设备型号，中文需要 UTF-8 编码。样例：MX5
screen_size	必填	设备屏幕尺寸	广告检索需要字段；字段参见 proto 文件“二维尺寸信息”
1.5 设备唯一识别码：请优先使用明文，务必填入真实信息，否则无法保证变现效果			
idfa	必填	iOS 设备唯一标识码	广告定向依赖 ID
imei	必填	Android 设备唯一标识码	广告定向依赖 ID
android_id	必填	Android 设备系统 ID	要求必填，如有问题请联系运营人员
mac	必填	设备 WiFi 网卡 MAC 地址	
idfa_md5	可替补 idfa	idfa 经过 MD5 方式加密	建议优先传明文，否则影响用户定向精准度
imei_md5	可替补 imei	imei 经过 MD5 方式加密	建议优先传明文，否则影响用户定向精准度
androidid_md5	可替补 android_id	MD5 加密 android_id	建议优先传明文，否则影响用户定向精准度
1.6 移动网络参数：用于广告系统针对性匹配广告和选择交互方式			
IPv4	必填	IPv4 地址	公网 IPv4 地址，服务器对接必填。确保填写的内容为用户设备的公网出口 IP 地址
Connection_type	必填	网络类型	移动设备网络连接方式的可选值集合。选项参见 proto 文件
operator_id	必填	运营商 ID	当前系统可以接受的移动运营商分类可选值集合。选项参见 proto 文件
cellular_id	选填	基站 ID	当前连接的运营商基站 ID，用于辅助用户定位
1.7 Wi-Fi Aps 网络参数：强烈建议填写，可填写多个，包括当前连接热点和其它周边热点			
ap_mac	选填	热点 mac 地址	重要参数，建议填写
rssi	选填	热点信号强度	重要参数，建议填写
ap_name	选填	热点名称	可用于识别用户所处场所，以精准定向
is_connected	选填	是否当前连接热点	该参数配合热点名称
1.8 GPS 参数：强烈建议填写。描述当前用户设备所处实时位置，用于触发 LBS 广告。			
CoordinateType	选填	GPS 坐标类型	
longitude	选填	GPS 坐标经度	
latitude	选填	GPS 坐标纬度	
timestamp	选填	GPS 时间戳信息	

### 3 广告应答字段解析

广告应答（**MobadsResponse**）包含了返回的广告信息和常见错误码。在收到应答后，需要先检查是否存在错误码，然后再解析广告结构。广告应答字段结构如下：



参数	参数定义	参数填写说明
1.1 基础参数		
request_id	广告请求 ID	由接入方自定义，在发起请求时填写，需确保 request_id 的唯一性。推荐使用 app_id+adslot_id+若干位随机 Key 构成
error_code	错误码	请求响应出错时的错误接入码，用于问题排查。错误码的具体成因可以参见《错误码对照表》
adslot_id	广告位 ID	对应请求时填写的广告位 ID
2.2 广告清单和元数据组：描述广告的基本内容和结构		
Ad ads	应答广告清单	在无错误码的的应答中，有若干广告描述信息。注意：信息流广告请求可以返回一个或多个广告，需要逐个解析（具体返回广告个数在 MSSP 广告位处设置）
expiration_time	广告清单过期时间戳	单位秒。清单过期后展现无效
ad_key	广告签名	当前返回广告创意的签名，是广告创意的唯一标识
meta_group	广告元数据组	支持返回的广告包含多个元数据，建议对接方优先解析此字段获取广告元数据信息。关于 meta_group 的详细说明请见 proto 文件。
total_num	广告元数据个数	每条广告对应元素组中元数据总数
html_snippet	HTML 片段	“返回 html 渲染模板”将在后续版本中支持。暂时只返回广告元数据信息，由对接方渲染；数据采用 base64 编码，获取后需进行解码。
2.3 广告物料元数据信息：提供单个广告创意的基本属性、素材、交互等信息。		
title	推广标题	中文需要 UTF-8 编码
brand_name	广告品牌名称	下载类则为 app 名称（如“小米商城”），非下载类则为品牌名称（如“小米”）
description	广告描述	默认只有一个元素，暂时约定最多只有两个元素，具体情况已线上广告实际返回结果为准，中文需要 UTF-8 编码

<b>image_src</b>	广告图片地址	<p>image_src 返回的图片尺寸为 MSSP 配置的当前广告位需要的图片尺寸（目前非严格按照此尺寸，广告系统会按近似的宽高比进行物料匹配）。</p> <p>如请求的是信息流视频，请取该字段的图片作为视频预览图，在视频播放前和播放完成后展示图片；如设置为 Wi-Fi 环境下自动播放视频，则只需在播放结束后展示图片。</p>
<b>material_width</b>	物料的宽度	如果是图片,表示图片的宽度;如果是视频(含有视频截图),则为视频宽度;如果是图文或文本,则不会填充此字段
<b>material_height</b>	物料的高度	如果是图片,表示图片的高度;如果是视频(含有视频截图),则为视频高度;如果是图文或文本,则不会填充此字段
<b>icon_src</b>	广告图标地址	注意：单个广告可能有多张图标返回。icon_src 返回的图标尺寸为标准尺寸为 90*90 像素
<b>creative_type</b>	创意类型	<p>当前 API 接口可以支持的广告创意类型集合，包括以下类型（具体选项可参考 Proto 文件“创意类型”部分）：</p> <p>①无创意类型（NO_TYPE），主要针对原生自定义素材广告，不再制定返回广告的创意类型，根据广告位设置对返回字段进行读取即可；</p> <p>②纯文字广告（TEXT），一般由 title、description 构成</p> <p>③纯图片（IMAGE），一般由单张或多张 image_src 构成；</p> <p>④图文混合广告（TEXT_ICON）。一般由单张 icon_src 和 title、description 构成；</p> <p>⑤视频广告（VIDEO），一般由视频 URL 和视频时长构成；</p> <p>⑥奖励视频广告（VIDEO_HTML），一般由视频 URL/视频时长和一个落地页面构成</p>
<b>interaction_type</b>	交互类型	<p>当前 API 接口可以支持的广告交互类型集合，包括以下类型（具体选项可参考 Proto 文件“交互类型”部分）：</p> <p>①无动作（NO_INTERACTION）。即广告广告点击后无需进行任何响应</p> <p>②打开网页（SURFING）。使用 Webview 或浏览器打开网页，目标页即 landing page，一般由广告主定义；建议使用应用内 Webview 打开，以免跳出应用；用户可能进行点击或其它操作，建议在 Webview 中设置关闭广告或返回按钮；</p> <p>③下载应用（DOWNLOAD）。Android 可设置为直接下载或让用户进行确认；iOS 可通过添加 Store Kit 框架，实现在应用内打开 app store 应用详情页（仅 iOS 6.0 以上设备支持），以免直接跳出应用。应用下载使用 iOS store kit 需要传入 itunesid，开发者可根据 iTunes 链接自行解析 itunesid（最新了解到，APPLE 官方对 storekit 接口维护有异常，请谨慎使用，可通过测试确认）。</p> <p>④可选动作(Optional)。此动作针对 app 使用。开发者可根据自己的需求选择是打开详情页，或是直接下载 app，或是获取相关数据并进行后续的转化上报。此动作时，click_url 中会增加选择宏\${ACCT_TYPE}，开发者收到 click_url 后需对该宏进行替换。</p> <p>（上报类型推荐选择 1，以获取最佳广告效果）</p> <p>❖ 0: 表示跳转到详情页；</p> <p>❖ 35: 表示跳转到应用下载地址，直接开始下载；</p> <p>返回 click_url 示例:</p>

		<a down_x":51,"down_y":69,"up_x":54,"up_y":26}&amp;acttype='\${ACCT_TYPE}"' href="http://www.trackingurl.com/tracking.php?i=1&amp;os=2&amp;s={">http://www.trackingurl.com/tracking.php?i=1&amp;os=2&amp;s={"down_x":51,"down_y":69,"up_x":54,"up_y":26}&amp;acttype=\${ACCT_TYPE}</a> 如需要下载,替换后的 click_url 示例: <a down_x":51,"down_y":69,"up_x":54,"up_y":26}&amp;acttype='35"' href="http://www.trackingurl.com/tracking.php?i=1&amp;os=2&amp;s={">http://www.trackingurl.com/tracking.php?i=1&amp;os=2&amp;s={"down_x":51,"down_y":69,"up_x":54,"up_y":26}&amp;acttype=35</a> ❖ 1: 表示开发者需要处理点击上报响应信息来获取 app 下载地址, 以及后续的转化上报数据 ( <b>仅在此情况下替换宏\${CLICKID}, 并上报转化</b> )。此时, 点击上报响应头中 status code 为 200 表示上报成功, 移动应用类广告的点击响应内容是 json map 格式, 具体字段如下: <pre>{   "ret":0, // 返回码, 0 表示成功, 1 表示失败   "data": // app 下载信息和转化上报信息   {     "dstlink":"http://xxx", // 该 app 的下载地址     "clickid":"xxx" // clickid 需缓存下来, 用于后续转化上报   } }</pre> 开发者需要用返回的 json map 中的 clickid 替换 ad_tracking 中相应事件 ( 包括 APP 开始下载、完成下载、安装 ) 的 tracking_url 的\${CLICKID}宏, 即可通过替换后的 url 进行相应的转化上报。
click_url	点击行为地址	用户点击后, 在客户端进行响应, 会经过多次 302 跳转最终到达目标地址 ( <b>采用 HTTP GET 方式上报</b> )。后续会将 click_url 迁移至 meta_tracking 或 ad_tracking 中, 请优先响应 ad_tracking 中的广告展现事件对应的汇报串, 替换 URL 中的点击坐标宏, \${DOWN_X}, \${DOWN_Y}, \${UP_X}, \${UP_Y}, 分别表示按下 X 坐标、按下 Y 坐标、弹起 X 坐标、弹起 Y 坐标, 若不能获取可替换为-999 ( <b>坐标参考原点为广告位左上角</b> )。例如: 替换前 <a href="http://www.trackingurl.com?viewid=7QQvisAJ4hwqaj4Ze3HNjSXxjyOqNsjs61BjLiKQqaA3KfcSZwObxPztkamQ&amp;s=%7B%22down_x%22%3A%22%22%22down_y%22%3A%22%22%22up_x%22%3A%22%22%22up_y%22%3A%22%22%22%7D">http://www.trackingurl.com?viewid=7QQvisAJ4hwqaj4Ze3HNjSXxjyOqNsjs61BjLiKQqaA3KfcSZwObxPztkamQ&amp;s=%7B%22down_x%22%3A%22%22%22down_y%22%3A%22%22%22up_x%22%3A%22%22%22up_y%22%3A%22%22%22%7D</a> 替换后 <a href="http://www.trackingurl.com?viewid=7QQvisAJ4hwqaj4Ze3HNjSXxjyOqNsjs61BjLiKQqaA3KfcSZwObxPztkamQ&amp;s=%7B%22down_x%22%3A%22-999%22%22down_y%22%3A%22-999%22%22up_x%22%3A%22-999%22%22up_y%22%3A%22-999%22%22%7D">http://www.trackingurl.com?viewid=7QQvisAJ4hwqaj4Ze3HNjSXxjyOqNsjs61BjLiKQqaA3KfcSZwObxPztkamQ&amp;s=%7B%22down_x%22%3A%22-999%22%22down_y%22%3A%22-999%22%22up_x%22%3A%22-999%22%22up_y%22%3A%22-999%22%22%7D</a>
app_package	下载类广告包名	对于下载类广告, 可以使用此字段判断当前设备是否已安装此应用, 如已安装, 在用户对当前广告发生点击行为后, 可以执行打开应用操作; 并在后台通过 click_url 汇报此次点击, 不再进行 302 跳转
app_size	应用文件大小	下载类广告 ( 应用 ) 文件大小
video_url	广告视频物料地址	当返回的广告是视频物料时, 此字段表示视频广告的地址, 支持流媒体播放。
video_duration	广告视频物料时长	当返回的广告是视频物料时, 此字段表示视频广告的时长, 单位秒

<b>meta_index</b>	元数据索引结构	<p>当前元数据在广告元数据组中的索引结构，包括当前广告的元数据总数和当前元数据所在索引</p> <p>改成广告创意元组，由创意数据+创意交互信息构成。创意交互信息描述某些创意特有的一些效果跟踪信息。</p> <p>当特定事件发生时，优先执行创意层级的操作，其次执行广告层级的操作。</p>
<b>2.4 广告效果跟踪信息</b>		
<b>win_notice_url</b>	-	<p>在广告曝光后由客户端逐个上报完，不能遗漏，否则会影响数据和收入。曝光同时还需上报 ad_tracking 中 AD_EXPOSURE 事件进行上报（采用 HTTP GET 方式上报）</p>
<b>ad_tracking</b>	广告效果跟踪信息	<p>用于监控广告展示过程中的事件，包括事件类型和相应的事件监控 URL，对接方可在具体事件到达时发送相应的事件监控 URL。（采用 HTTP GET 方式上报）</p> <p>目前支持的监控事件包括三大类：</p> <p>1）广告展示过程中的共性事件：广告被点击（同 click_url 需做点击坐标宏替换）、广告被展现和广告被关闭（如用户主动关闭广告或用户退出应用界面等）。比如对接方在用户点击广告时需要及时发送相应的 URL，这将会有助于提升流量的变现能力；</p> <p>2）视频类广告展示过程事件：视频开始播放、视频全屏播放、视频播放结束（包括播放完成/未播放完）、点击预览图播放视频等。API 接口支持信息流视频物料，对于要接入信息流视频的用户必须实现相应的监控接口，对于视频类事件，对接方除了发送相应的监控 URL，还需要对 URL 里面的 \${PROGRESS} 宏进行替换，该宏表示触发该事件时的广告播放进度，单位毫秒，再发送这个 URL。示例：如当前播放 7.5 秒，格式为 http://www.trackingurl.com/tracking.php?c=xxx&amp;event=xxx&amp;progress=7500；【奖励视频 tracking 详见独立说明文档-《广告_KSMobads_API_5.5.0（奖励视频）接口参数规范.pdf》】</p> <p>3）下载类广告推广 APP 相关事件：下载 APP、安装 APP 与激活 APP（如无法监测下载和激活事件，则可以不上报）；</p> <p>Android APP 推荐上报下载 APP 和安装 APP 事件；</p> <p>值得说明，如果对接方收到相应的监控事件，却没有对应的事件监控 URL，此时对接方可忽略对该事件的处理。更详细的广告效果监控信息见 proto 文件。</p>

## 4 请求/反馈举例

### 4.1 请求 JSON 举例(测试环境)：

```
{
  "request_id": "464251180c6faa2785d91e1f5b1c829c",
  "api_version": {
    "major": 5,
    "minor": 5,
    "micro": 0
  },
  "adslot": {
    "adslot_id": "ganuxntb",
    "adslot_size": {
      "width": 300,
```

```

    "height": 250
  },
  "app": {
    "app_id": "6bq3bpop",
    "app_version": {
      "major": 3,
      "minor": 4,
      "micro": 0
    },
    "app_package": "com.example.ANDROIDdebug"
  },
  "device": {
    "device_type": 1,
    "os_type": 1,
    "os_version": {
      "major": 3,
      "minor": 1,
      "micro": 0
    },
    "vendor": "XiaoMi",
    "model": "2S",
    "udid": {
      "imei": "866926020248380",
      "mac": "d8:55:a3:ce:e4:40",
      "android_id": "5b7e9e4f42a6635f"
    },
    "screen_size": {
      "width": 1080,
      "height": 1920
    }
  },
  "network": {
    "ipv4": "114.255.44.132",
    "connection_type": 100,
    "operator_type": 1
  },
  "gps": {
    "coordinate_type": 1,
    "longitude": 74.0059,
    "latitude": 40.7127,
    "timestamp": 123456
  },
  "is_debug": true,
  "request_protocol_type": 2
}

```

#### 4.2 反馈 JSON 举例：

[illegible]



[illegible]



```
"tracking_event": 102012,
"tracking_url": [

"http://track.ksyun.com/track/sd?e=102012&sd=Mwk2NzE2YmE4MWQ5ZDI4ZjJiYmU0MGQxZWU5MzcyZTU5ZQk2NzE2YmE4MWQ5ZDI4ZjJiYmU0MGQxZWU5MzcyZTU5ZTAJeDd1cm1mOXUJYngwdG81d3EJZ2FudXhudGIJMQk4NjY5MjYwMjYyNDgzODA="
],
{
  "tracking_event": 0,
  "tracking_url": [

"http://track.ksyun.com/track/ck?c=Mwk2NzE2YmE4MWQ5ZDI4ZjJiYmU0MGQxZWU5MzcyZTU5ZTAJeDd1cm1mOXUJYngwdG81d3EJZ2FudXhudGIJMQk4NjY5MjYwMjYyNDgzODAJaHR0cDovL2R1b2tvby5iYWlkdS5jb20vZ2FtZS8_cGFnZWlkPUhka2ljc3NwJnBfdGFnPTE3NzM5MzQ=&s=%7B%22down_x%22%3A%22${DOWN_X}%22%2C%22down_y%22%3A%22${DOWN_Y}%22%2C%22up_x%22%3A%22${UP_X}%22%2C%22up_y%22%3A%22${UP_Y}%22%7D"
],
},
{
  "tracking_event": 101000,
  "tracking_url": [

"http://track.ksyun.com/track/vs?e=101000&progress=${PROGRESS}&c=Mwk2NzE2YmE4MWQ5ZDI4ZjJiYmU0MGQxZWU5MzcyZTU5ZQk2NzE2YmE4MWQ5ZDI4ZjJiYmU0MGQxZWU5MzcyZTU5ZTAJeDd1cm1mOXUJYngwdG81d3EJZ2FudXhudGIJUTN0UHVDN1Z6QWcxbHloaFNeeUVWb3hwYld2VmhhNaU9LTVYyNkRMcGJ0US0JMQk4NjY5MjYwMjYyNDgzODA="
],
},
{
  "tracking_event": 101001,
  "tracking_url": [

"http://track.ksyun.com/track/vs?e=101001&progress=${PROGRESS}&c=Mwk2NzE2YmE4MWQ5ZDI4ZjJiYmU0MGQxZWU5MzcyZTU5ZQk2NzE2YmE4MWQ5ZDI4ZjJiYmU0MGQxZWU5MzcyZTU5ZTAJeDd1cm1mOXUJYngwdG81d3EJZ2FudXhudGIJUTN0UHVDN1Z6QWcxbHloaFNeeUVWb3hwYld2VmhhNaU9LTVYyNkRMcGJ0US0JMQk4NjY5MjYwMjYyNDgzODA="
],
},
{
  "tracking_event": 101002,
  "tracking_url": [

"http://track.ksyun.com/track/vs?e=101002&progress=${PROGRESS}&c=Mwk2NzE2YmE4MWQ5ZDI4ZjJiYmU0MGQxZWU5MzcyZTU5ZQk2NzE2YmE4MWQ5ZDI4ZjJiYmU0MGQxZWU5MzcyZTU5ZTAJeDd1cm1mOXUJYngwdG81d3EJZ2FudXhudGIJUTN0UHVDN1Z6QWcxbHloaFNeeUVWb3hwYld2VmhhNaU9LTVYyNkRMcGJ0US0JMQk4NjY5MjYwMjYyNDgzODA="
],
},
{
  "tracking_event": 101003,
  "tracking_url": [

"http://track.ksyun.com/track/vs?e=101003&progress=${PROGRESS}&c=Mwk2NzE2YmE4MWQ5ZDI4ZjJiYmU0MGQxZWU5MzcyZTU5ZQk2NzE2YmE4MWQ5ZDI4ZjJiYmU0MGQxZWU5MzcyZTU5ZTAJeDd1cm1mOXUJYngwdG81d3EJZ2FudXhudGIJUTN0UHVDN1Z6QWcxbHloaFNeeUVWb3hwYld2VmhhNaU9LTVYyNkRMcGJ0US0JMQk4NjY5MjYwMjYyNDgzODA="
],
},
{
  "tracking_event": 101009,
  "tracking_url": [

"http://track.ksyun.com/track/vs?e=101009&progress=${PROGRESS}&c=Mwk2NzE2YmE4MWQ5ZDI4ZjJiYmU0MGQxZWU5MzcyZTU5ZQk2NzE2YmE4MWQ5ZDI4ZjJiYmU0MGQxZWU5MzcyZTU5ZTAJeDd1cm1mOXUJYngwdG81d3EJZ2FudXhudGIJUTN0UHVDN1Z6QWcxbHloaFNeeUVWb3hwYld2VmhhNaU9LTVYyNkRMcGJ0US0JMQk4NjY5MjYwMjYyNDgzODA="
],
},
{
  "tracking_event": 101052,
  "tracking_url": [

"http://track.ksyun.com/track/vs?e=101052&progress=${PROGRESS}&c=Mwk2NzE2YmE4MWQ5ZDI4ZjJiYmU0MGQxZWU5MzcyZTU5ZQk2NzE2YmE4MWQ5ZDI4ZjJiYmU0MGQxZWU5MzcyZTU5ZTAJeDd1cm1mOXUJYngwdG81d3EJZ2FudXhudGIJUTN0UHVDN1Z6QWcxbHloaFNeeUVWb3hwYld2VmhhNaU9LTVYyNkRMcGJ0US0JMQk4NjY5MjYwMjYyNDgzODA="
],
},
{
  "tracking_event": 1,
  "tracking_url": [

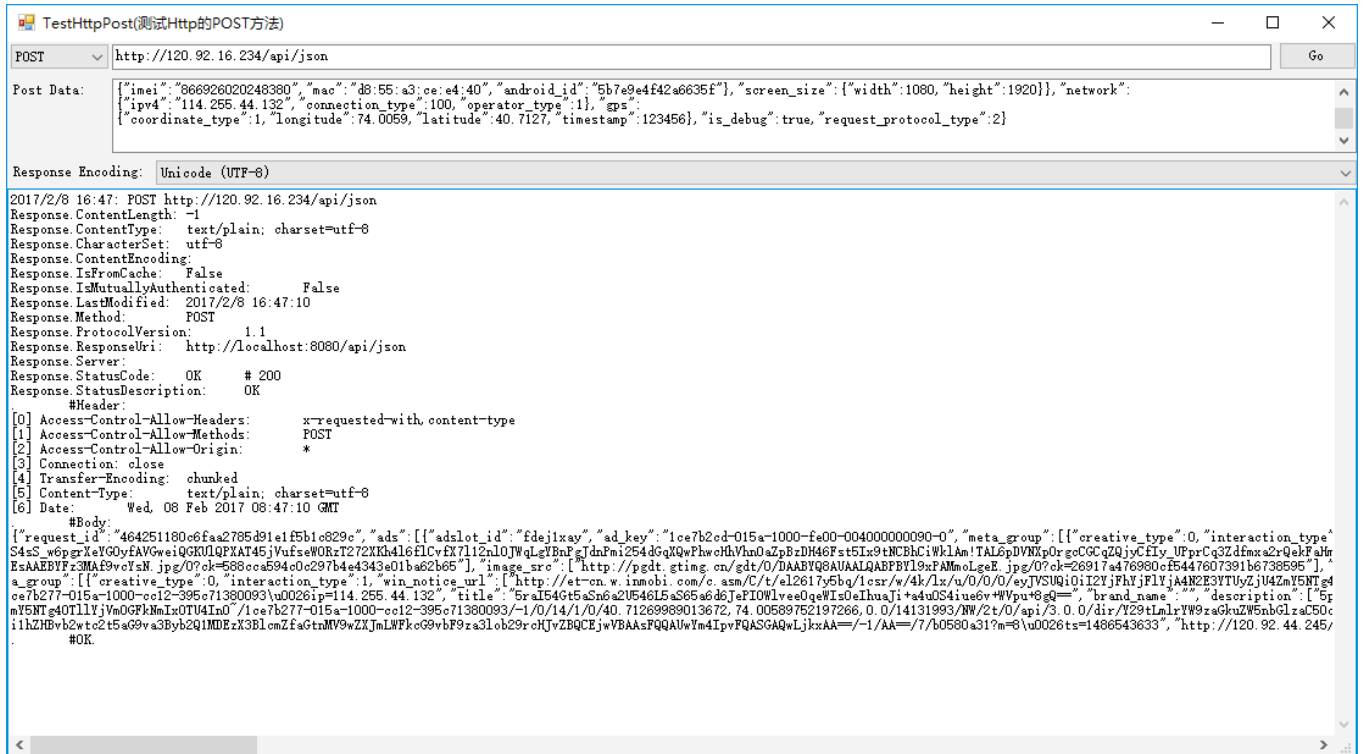
"http://track.ksyun.com/track/imp?v=Mwk2NzE2YmE4MWQ5ZDI4ZjJiYmU0MGQxZWU5MzcyZTU5ZQk2NzE2YmE4MWQ5ZDI4ZjJiYmU0MGQxZWU5MzcyZTU5ZTAJeDd1cm1mOXUJYngwdG81d3EJZ2FudXhudGIJUTN0UHVDN1Z6QWcxbHloaFNeeUVWb3hwYld2VmhhNaU9LTVYyNkRMcGJ0US0JMQk4NjY5MjYwMjYyNDgzODA="
],
}
```

```
],
"ad_key": "bx0to5wqx7urmf9u",
"meta_group": [
  {
    "creative_type": 9,
    "interaction_type": 2,
    "click_url": "http://duokoo.baidu.com/game/?pageid=Hdkicssp&p_tag=1773934",
    "description": [
      "5Lmw5oi/6LS177yf5Zyo6L+Z6YeM5Lmw5oi/5bCx5YOP5Lmw55m96l+c77yB"
    ],
    "icon_src": [
      "http://kscdn.ksyun.com/res/img/201702/116a446a-bc1c-4afe-a544-eaf21b4d239f.png"
    ],
    "image_src": [
      "http://kscdn.ksyun.com/res/img/201702/a4e1169f-5f2e-4f70-a9ed-66f97ca14cb9.jpg"
    ],
    "app_package": "da",
    "app_size": 0,
    "video_url": "http://kscdn.ksyun.com/res/video/201702/ea2e85ed-6d1c-469c-b0e6-bfc8bce04dde.mp4",
    "video_duration": 30,
    "material_width": 0,
    "material_height": 0
  }
],
"vid": "6716ba81d9d28f2bbe40d1ee9372e59e0",
"open_mode": 1
}
],
"expiration_time": 1513861609,
"request_time_s": 0.001756744
}
```

## 5 模拟测试

采用提供的模拟发送程序 TestHttpPost，按步骤进行：

- 1.发送方式选择 **POST**;
  - 2.地址栏中填入**生产或线上**环境的 **URL**;
  - 3.Post Data 中填入请求 json , 测试时可以采用上一部分中的**请求 JSON 举例**;
  - 4.点击【**Go**】按钮
- 即可模拟测试数据请求与反馈



## 6 数据结构定义参考(golang 版)

### 6.1 请求结构体

```
type MobadsRequest struct {
    RequestId      *string      `json:"request_id,omitempty"`
    ApiVersion     *Version     `json:"api_version,omitempty"`
    Adslot         *AdSlot      `json:"adslot,omitempty"`
    App            *App         `json:"app,omitempty"`
    Device         *Device      `json:"device,omitempty"`
    Network        *Network     `json:"network,omitempty"`
    Gps            *Gps         `json:"gps,omitempty"`
    IsDebug        *bool        `json:"is_debug,omitempty"`
    RequestProtocolType *MobadsRequest_RequestProtocolType `json:"request_protocol_type,omitempty"`
    SearchKey      *string      `json:"search_key,omitempty"`
    MediaType      *uint32      `json:"media_type,omitempty"`
}

type Version struct {
    Major *uint32 `json:"major,omitempty"`
    Minor *uint32 `json:"minor,omitempty"`
    Micro *uint32 `json:"micro,omitempty"`
}

type App struct {
    AppId      *string `json:"app_id,omitempty"`
    ChannelId  *string `json:"channel_id,omitempty"`
    AppVersion *Version `json:"app_version,omitempty"`
    AppPackage *string `json:"app_package,omitempty"`
}

type AdSlot struct {
    AdslotId *string `json:"adslot_id,omitempty"`
    AdslotSize *Size `json:"adslot_size,omitempty"`
    Video     *Video `json:"video,omitempty"`
    AdslotType *uint32 `json:"adslot_type,omitempty"`
    Ads       *uint32 `json:"ads,omitempty"`
}

type Udid struct {
    Idfa *string `json:"idfa,omitempty"`
    Imei *string `json:"imei,omitempty"`
    Mac  *string `json:"mac,omitempty"`
    AndroidId *string `json:"android_id,omitempty"`
    IdfaMd5 *string `json:"idfa_md5,omitempty"`
    ImeiMd5 *string `json:"imei_md5,omitempty"`
    AndroididMd5 *string `json:"androidid_md5,omitempty"`
    Ksid *string `json:"ksid,omitempty"`
}

type Size struct {
    Width *uint32 `json:"width,omitempty"`
    Height *uint32 `json:"height,omitempty"`
}

type Device struct {
    DeviceType *Device_DeviceType `json:"device_type,omitempty"`
    OsType     *Device_OsType     `json:"os_type,omitempty"`
    OsVersion  *Version            `json:"os_version,omitempty"`
    Vendor     *string             `json:"vendor,omitempty"`
    Model      *string             `json:"model,omitempty"`
    Udid       *Udid               `json:"udid,omitempty"`
    ScreenSize *Size               `json:"screen_size,omitempty"`
    BrowserType *int32              `json:"browser_type,omitempty"`
    Ua         *string             `json:"ua,omitempty"`
    Referer    *string             `json:"referer,omitempty"`
}

type Network struct {
    Ipv4 *string `json:"ipv4,omitempty"`
    ConnectionType *Network_ConnectionType `json:"connection_type,omitempty"`
    OperatorType *Network_OperatorType `json:"operator_type,omitempty"`
    CellularId *string `json:"cellular_id,omitempty"`
    WifiAps []*WifiAp `json:"wifi_aps,omitempty"`
}

type WifiAp struct {
    ApMac *string `json:"ap_mac,omitempty"`
    Rssi *int32 `json:"rssi,omitempty"`
    ApName *string `json:"ap_name,omitempty"`
    IsConnected *bool `json:"is_connected,omitempty"`
}

type Gps struct {
    CoordinateType *Gps_CoordinateType `json:"coordinate_type,omitempty"`
    Longitude *float64 `json:"longitude,omitempty"`
    Latitude *float64 `json:"latitude,omitempty"`
    Timestamp *uint32 `json:"timestamp,omitempty"`
}
```

```

type Video struct {
    Title          *string          `json:"title,omitempty"`
    ContentLength  *uint32           `json:"content_length,omitempty"`
    Copyright      *Video_CopyRight `json:"copyright,omitempty"`
}
type Video_CopyRight int32
type Device_DeviceType int32
type Device_OsType int32
type Network_ConnectionType int32
type Network_OperatorType int32
type Gps_CoordinateType int32
type MobadsRequest_RequestProtocolType int32

```

## 6.2 反馈结构体

```

type MobadsResponse struct {
    RequestId      *string          `json:"request_id,omitempty"`
    ErrorCode      *uint64          `json:"error_code,omitempty"`
    Ads            []*Ad            `json:"ads,omitempty"`
    ExpirationTime *uint32          `json:"expiration_time,omitempty"`
    AutomaticPro   *bool            `json:"automatic_pro,omitempty"`
    AdStrategy     []*AdStrategy    `json:"ad_strategy,omitempty"`
    TurnType       *MobadsResponse_TurnType `json:"turn_type,omitempty"`
    TurnDuration   *uint32          `json:"turn_duration,omitempty"`
    RequestTimeS   *float64         `json:"request_time_s,omitempty"`
}
type MetaIndex struct {
    TotalNum      *uint32 `json:"total_num,omitempty"`
    CurrentIndex  *uint32 `json:"current_index,omitempty"`
}
type Ad struct {
    AdslotId      *string          `json:"adslot_id,omitempty"`
    AdKey         *string          `json:"ad_key,omitempty"`
    MetaGroup     []*MaterialMeta  `json:"meta_group,omitempty"`
    HtmlSnippet   *string          `json:"html_snippet,omitempty"`
    AdTracking    []*Tracking      `json:"ad_tracking,omitempty"`
    MobAdtext     *string          `json:"mob_adtext,omitempty"`
    MobAdlogo     *string          `json:"mob_adlogo,omitempty"`
    Vid           *string          `json:"vid,omitempty"`
    OpenMode      *int32           `json:"open_mode,omitempty"`
}
type MaterialMeta struct {
    CreativeType      *MaterialMeta_CreativeType `json:"creative_type,omitempty"`
    InteractionType   *MaterialMeta_InteractionType `json:"interaction_type,omitempty"`
    WinNoticeUrl     []string                   `json:"win_notice_url,omitempty"`
    ClickUrl         *string                   `json:"click_url,omitempty"`
    Title            []byte                    `json:"title,omitempty"`
    BrandName        *string                   `json:"brand_name,omitempty"`
    Description       [][]byte                  `json:"description,omitempty"`
    IconSrc          []string                  `json:"icon_src,omitempty"`
    ImageSrc         []string                  `json:"image_src,omitempty"`
    AppPackage       *string                   `json:"app_package,omitempty"`
    AppSize          *uint32                   `json:"app_size,omitempty"`
    VideoUrl         *string                   `json:"video_url,omitempty"`
    VideoDuration    *uint32                   `json:"video_duration,omitempty"`
    MetaIndex        *MaterialMeta_IndexType   `json:"meta_index,omitempty"`
    MaterialWidth     *uint32                   `json:"material_width,omitempty"`
    MaterialHeight    *uint32                   `json:"material_height,omitempty"`
}
type Tracking struct {
    TrackingEvent *Tracking_TrackingEvent `json:"tracking_event,omitempty"`
    TrackingUrl   []string                 `json:"tracking_url,omitempty"`
}
type AdStrategy struct {
    AppId      *string          `json:"app_id,omitempty"`
    AdslotId   *string          `json:"adslot_id,omitempty"`
    ChannelType *AdStrategy_ChannelType `json:"Channel_type,omitempty"`
    Weight     *uint32          `json:"weight,omitempty"`
}
type MaterialMeta_CreativeType int32
type MaterialMeta_InteractionType int32
type Tracking_TrackingEvent int32
type AdStrategy_ChannelType int32
type MobadsResponse_TurnType int32

const (
    // 广告展示过程共性事件
    Tracking_AD_CLICK      Tracking_TrackingEvent = 0
    Tracking_AD_EXPOSURE   Tracking_TrackingEvent = 1
    Tracking_AD_CLOSE      Tracking_TrackingEvent = 2
    // 视频类广告展示过程事件
    Tracking_VIDEO_AD_START      Tracking_TrackingEvent = 101000
    Tracking_VIDEO_AD_FULL_SCREEN Tracking_TrackingEvent = 101001
    Tracking_VIDEO_AD_END        Tracking_TrackingEvent = 101002

```

```
Tracking_VIDEO_AD_START_CARD_CLICK Tracking_TrackingEvent = 101003
Tracking_VIDEO_AD_CLOSE Tracking_TrackingEvent = 101009
Tracking_VIDEO_AD_RESTART Tracking_TrackingEvent = 101052
// 下载类广告推广APP 相关事件
Tracking_APP_AD_DOWNLOAD Tracking_TrackingEvent = 102000
Tracking_APP_AD_INSTALL Tracking_TrackingEvent = 102001
Tracking_APP_AD_ACTIVE Tracking_TrackingEvent = 102002
Tracking_APP_AD_START_DOWNLOAD Tracking_TrackingEvent = 102009
Tracking_APP_AD_START_MID_DOWNLOAD Tracking_TrackingEvent= 102012
)
```

# FAQ

## 1. 建议的对接步骤？

本地测试阶段：根据所采用的编程语言，参照 6 中的数据结构定义，对请求与反馈数据包进行定义，生成对应的 json 串，同所给示例中的 json 进行比对，确认无误后，可采用所给的模拟请求软件 TestHttpPost 进行测试，同样，对反馈 json 可先采用所给示例进行模拟解析；

测试环境对接阶段：联系运营人员，获取测试环境的广告位 id 及应用 id 并进行替换。最后，使用 post 方式向测试服务器 <http://120.92.44.245/api/json> 发送和接收 json 数据；

生产环境对接阶段：在测试环境下测试通过，并核对业务数据没有问题，即可联系运营人员分配正式的 id，进行替换进行跑量。

## 2. 收到的 json 反馈数据同协议所给多出一些信息？

在不影响数据解析的基础上，根据实际需要协议会进行升级，一般会保持向前兼容，如果出现一些多余信息可直接忽略，也可联系运营人员获取最新的协议文档，并进行相应升级。

## 3. 收到的 json 反馈数据解析失败？

排查时，首先采用 json 检测工具进行检查，如：<http://www.bejson.com/>，然后对反馈数据包的数据结构定义进行检查，如：数据类型，变量名称等。

## 4. 反馈数据不正常，或总获取不到广告？

对接过程中，请先自查所请求的 json 串格式和内容上是否合法，符合协议规定，可对比所给的示例请求 json 串。对于有错误码返回的，请结合《错误码对照表》进行参数的检查，确认没有问题可将请求串提交给对接支持人员协查原因。



## 修订记录

编号	修订日期	修订内容	修订作者
1	2017.02.08	V1.0.0 初稿	ChenBintao
2	2017.02.08	V1.0.1 增加对 html_snippet 字段的数据类型及 base64 说明	ChenBintao
3	2017.02.09	V1.0.2 增加 FAQ	ChenBintao
4	2017.02.16	V1.0.4 增加点击上报 click_url 中点击位置的填充	ChenBintao
5	2017.02.22	V1.0.5 增加点击坐标上传示例	ChenBintao
6	2017.02.24	V1.0.6 修改请求中部分[]byte 参数的数据类型为 string	ChenBintao
7	2017.02.28	V1.0.8 增加新事件类型，对应 protobuf 协议 5.4.2	ChenBintao
8	2017.09.25	V5.5.0 展示上报由 win_notice，移动至 Tracking	ChenBintao
9	2017.12.22	V5.5.0 增加最新请求反馈示例	ChenBintao
10			
11			
12			
13			
14			
15			