上海爱奇艺文化传媒有限公司 |2016年8月

爱奇艺程序化购买技术对接文档

V2.3

目录

[1. 广告主上传接口 3](#_Toc459302512)

[1.1接口描述 3](#_Toc459302513)

[1.2接口定义 3](#_Toc459302514)

[1.2.1接口地址 3](#_Toc459302515)

[1.2.2上传及更新接口参数 3](#_Toc459302516)

[1.2.3上传接口返回 4](#_Toc459302517)

[1.2.4状态查询参数 4](#_Toc459302518)

[1.2.5状态查询返回 4](#_Toc459302519)

[1.3调用示例 4](#_Toc459302520)

[1.3.1上传及更新示例 4](#_Toc459302521)

[1.3.2上传返回 5](#_Toc459302522)

[1.3.3状态查询示例 5](#_Toc459302523)

[1.3.4状态查询返回 6](#_Toc459302524)

[1.3.5状态批量查询示例 6](#_Toc459302525)

[1.3.6批量查询返回 6](#_Toc459302526)

[2.素材上传接口 7](#_Toc459302527)

[2.1接口描述 7](#_Toc459302528)

[2.2接口定义 7](#_Toc459302529)

[2.2.1接口地址 7](#_Toc459302530)

[2.2.2上传接口参数 7](#_Toc459302531)

[2.2.3上传接口返回 9](#_Toc459302532)

[2.2.4查询接口参数 9](#_Toc459302533)

[2.2.5查询接口返回 9](#_Toc459302534)

[2.3调用示例 10](#_Toc459302535)

[2.3.1上传示例 10](#_Toc459302536)

[2.3.2上传返回 11](#_Toc459302537)

[2.3.3查询示例 11](#_Toc459302538)

[2.3.4查询返回 11](#_Toc459302539)

[2.3.5批量查询示例 12](#_Toc459302540)

[2.3.6批量查询返回 12](#_Toc459302541)

[2.3.7批量查询示例 13](#_Toc459302542)

[2.3.8按状态查询返回 13](#_Toc459302543)

[3.竞价API 13](#_Toc459302544)

[3.1概述 13](#_Toc459302545)

[3.2账户要求 13](#_Toc459302546)

[3.3准备工作 13](#_Toc459302547)

[3.4 QAX请求DSP的竞价接口（BidRequest） 14](#_Toc459302548)

[3.4.1 接口参数说明 14](#_Toc459302549)

[3.4.2 BidRequest示例 15](#_Toc459302550)

[3.5 DSP返回给QAX的竞价结果接口(Bid Response) 17](#_Toc459302551)

[3.5.1 接口参数说明 17](#_Toc459302552)

[3.5.2 BidResponse示例 17](#_Toc459302553)

[3.5.3 vast示例 18](#_Toc459302554)

[3.5.4vast相关说明 19](#_Toc459302555)

[3.5.5 曝光监测接口与解密方式 23](#_Toc459302556)

[4. Cookie Matching Service API 26](#_Toc459302557)

[4.1 Base URL 26](#_Toc459302558)

[4.2 Redirect URL 26](#_Toc459302559)

[附录一：【user id】命名规则 27](#_Toc459302560)

[附录二：第三方监测说明补充信息 28](#_Toc459302561)

# 1. 广告主上传接口

## 1.1接口描述

使用接口进行广告主上传需要先获取由爱奇艺Ad Exchange分配的dsp\_token，这个dsp\_token同样适用于素材上传接口。目前提供两个接口：广告主上传接口、广告主状态查询接口。

**【广告主上传接口】**DSP通过http post请求广告主上传接口，进行信息及文件上传。广告主上传信息参数包括广告主ID，广告主名称，行业信息及资质相关附件，因为附件传输要占用post body，因此广告主信息相关参数均设置在http header。

**【查询接口】**通过http get请求进行查询。广告主上传成功后即可通过该广告主ID进行审核状态查询，审核通过后即可使用该广告主进行素材上传。

## 1.2接口定义

### 1.2.1接口地址

广告主上传及更新接口：<http://220.181.184.220/upload/advertiser>

广告主审核状态查询：<http://220.181.184.220/upload/api/advertiser>

广告主状态批量查询：<http://220.181.184.220/upload/api/batchAdvertiser> 注:广告主上传及审核状态查询均不提供测试环境，通过正式环境测试。

### 1.2.2上传及更新接口参数

* dsp\_token：字符串类型。用来唯一标识合作方。
* ad\_id：必填项，数字类型。广告主ID。
* name：字符串类型。广告主名称。统一编码方式:UTF-8.
* industry：字符串类型。行业信息，可以为空。
* op：

字符串类型。操作类型，update或者create。默认为create，表示新建。update表示对已经上传的广告主信息进行更新，已经审核通过的广告主不能进行更新操作。原来审核不通过的广告主更新后会重新变为待审核状态。

* file\_name：

统一编码方式:UTF-8.字符串类型。附件名称，当附件为空时该项可以为空。名称要求有正确的后缀标示，否则资质审核工作人员可能无法正确打开文件，带来不便。文件格式支持：ZIP、RAR、JPG、JPEG、PNG、BMP。 多个资质使用压缩包上传。仅支持一个资质/一个压缩文件上传。

### 1.2.3上传接口返回

接口以json格式返回。

* code：状态码，定义如下：

0 上传成功

1001 认证错误（token错误）

4001 参数错误

5001 服务端错误

5002 用户同时上传超过限制,并行上传限制5。

5003 应用请求数超过限制

* desc：错误信息，只有code为4001时才返回该项。

### 1.2.4状态查询参数

* dsp\_token：字符串类型，用来唯一标识合作方。
* ad\_id：数字类型，上传成功的广告主ID。
* batch：批量查询参数，多个ad\_id，用英文逗号分隔。

### 1.2.5状态查询返回

接口以json格式返回。

* code：状态码，定义同上传接口。
* status：审核状态，分别为：

WAIT：待审核

PASS：审核通过

UNPASS：审核不通过

* reason：审核不通过原因，仅当审核状态为UNPASS时返回。

## 1.3调用示例

### 1.3.1上传及更新示例

Java实现方式如下：

//上传接口地址

String uploadUrl = “http://127.0.0.1/post/”;

String dspToken = “testToken”;

Long adId = 1234567;

String name = “测试广告主”;

String industry = “其他”;

String file\_name = "资质.rar";

String op = “update”

//上传附件内容

byte[] content = {};

DefaultHttpClient client = new DefaultHttpClient();

try {

HttpPost post = new HttpPost(uploadUrl);

post.addHeader(“dsp\_token”, dspToken);

post.addHeader(“ad\_id”, String.valueOf(adId));

post.addHeader(“name”, URLEncoder.encode(name, “UTF-8”));

post.addHeader(“industry”, URLEncoder.encode(industry, “UTF-8”));

post.addHeader(“file\_name”, URLEncoder.encode(fileName, “UTF-8”));

post.addHeader(“op”, op);

try {

post.setEntity(new ByteArrayEntity(content));

HttpResponse response = client.execute(post);

} catch (ClientProtocolException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

} finally {

post.releaseConnection();

}

} catch (Exception e) {

e.printStackTrace();

}

### 1.3.2上传返回

* 成功:

{

“code”: “0”

}

* 失败:

{

“code”: “4001”

}

### 1.3.3状态查询示例

http://220.181.184.220/upload/api/advertiser?dsp\_token=xxxxxxx&ad\_id=12345678

### 1.3.4状态查询返回

* 审核通过时:

{

“code”: 0,

“status”: “PASS”

}

* 审核不通过时:

{

“code”: 0,

“status”: “UNPASS”,

“reason”: “未知”

}

### 1.3.5状态批量查询示例

http://220.181.184.220/upload/api/batchAdvertiser?dsp\_token=xxxxxxx&batch=1137,1138,1139

### 1.3.6批量查询返回

{

“code”: 0,

“results”: [

{

“ad\_id”: “1137”,

“status”: “WAIT”

},

{

“ad\_id”: “1138”,

“status”: “UNPASS”,

“reason”: “资质不规范”

},

{

“ad\_id”: “1139”,

“status”: “PASS”

}

]

}

# 2.素材上传接口

## 2.1接口描述

使用接口进行广告素材上传需要先获取由爱奇艺Ad Exchange分配的dsp\_token，与广告主上传接口使用的dsp\_token一致。目前提供三个接口：素材上传接口、素材状态查询接口、状态批量查询接口。

**【素材上传接口】**通过http post请求素材上传接口，进行文件上传。参数包括合作方标识、 文件名称、创意类型、平台类型、跳转地址、广告主ID、创意视频ID以及视频文件内容。因为文件内容要占用post body，所以其他参数均以header形式发送。

**【素材状态查询接口、状态批量查询接口】**

文件上传成功或失败接口会返回对应状态码及信息(详见下述)。素材文件上传完毕，服务端会返回一个m\_id，后续可结合dsp\_token和m\_id调用素材状态查询接口进行状态查询。当查询到素材状态为COMPLETE即完成时，服务端会同时返回一个tv\_id，根据tv\_id可应用该素材展开竞价及投放等工作（请注意：BidResponse中涉及的cr\_id为此处获取的tv\_id）。

查询接口通过http get请求进行查询。因为素材上传后转码及分发等处理比较耗时， 而且处理完毕有人工审核的环节，不一定可及时跟进，因此建议查询素材状态的轮询间隔可以放大一些。

## 2.2接口定义

### 2.2.1接口地址

上传接口：<http://220.181.184.220/upload/post>

查询接口：<http://220.181.184.220/upload/api/query>

批量查询：<http://220.181.184.220/upload/api/batchQuery>

批量查询：[http://220.181.184.220/upload/api/ queryByStatus](http://220.181.184.220/upload/api/batchQuery)

注:素材上传及审核状态查询均不提供测试环境，通过正式环境测试。

### 2.2.2上传接口参数

* dsp\_token：字符串类型，用来唯一标识合作方。
* click\_url：字符串类型，点击广告后着陆地址(landing page)，必填。这里只用来审核使用。
* ad\_id：数字类型，通过广告主上传接口成功上传的广告主ID。
* video\_id：

数字或者字符串类型，创意视频或者动画等素材的ID，相同ID的素材在服务端只留存一份。上传创意时若对应素材已有过上传记录，那么只需传该素材ID即可，无需再上传该素材文件。

* ad\_type：

上传创意类型，可为空。目前支持四种创意类型，1为贴片创意、2为暂停创意4为角标、9为overlay。为空时，默认为贴片创意。

* file\_name：

字符串类型，文件名称，必填。名称必须带有效后缀，目前支持的贴片素材格式有：mp4、flv、mov、avi、mpeg、jpg、jpeg，其中jpg和jpeg图片格式只支持5秒贴片；暂停广告素材格式有：jpg、jpeg、swf、png；角标、overlay广告素材格式有：swf、jpg、jpeg。

* platform：

广告投放平台，必填。1为pc端、2为移动端。当创意类型为贴片时，该字段PC端和移动端通用。

* duration：

创意时长，数字类型。上传视频格式创意时可不填，当创意类型为角标时，该字段必须填写，角标时长为120。

* dpi：

素材分辨率，字符串类型。只有上传overlay广告创意且素材格式为swf时，该字段必须填写，且分辨率为728x90、 480x70、468x60、450x50其中之一。

* position：

广告展示位置，字符串类型。只有当上传角标或overlay广告创意时，可填写该字段。角标创意备选项为left、right，默认为left；overlay创意备选项为top、bottom，默认为bottom。

* end\_date：

创意截止有效时间，字符串类型，格式为yyyyMMdd。默认为3个月，从上传日算最多不超过一年。过期后创意视频、图片等素材将被下架。

* creative\_mode：
* 广告创意模式：选填。1为Normal, 2为Interactive。该参数仅在PCWeb端暂停广告/Overlay需在素材内实现点击跳转功能的情况下有效。
* apple\_id：

应用ID:iPhone客户端APP在苹果应用商店的APP id。如果投放iPhone端一键点击后跳转到APP Store的贴片广告时，属于必填字段（否则会出现点击后无反应的情况）。

* tag：
* 素材标签，数字类型，选填但建议填写，这样方便针对不同类型素材进行相关统计。tag对照如下：

100 餐饮美食

110 房地产

120 职业

130 婚恋

140 互联网与电信

150 3C

160 计算机

170 家居

180 教育

190 金融与投资

200 旅游

210 汽车

220 奢侈品

230 时尚与服饰

240 体育

250 医疗与健康

260 新闻与咨询

270 游戏

280 幼儿育婴

290 娱乐与休闲

300 生活

### 2.2.3上传接口返回

接口以json格式返回。

* code：状态码，定义如下：

0 上传成功

1001 认证错误（token错误）

4001 参数错误

5001 服务端错误

5002 上传素材数量超过限制（每日上传限制：200，并发上传限制：5）

5003 应用请求超过限制

* m\_id：上传成功的素材id，只有code为0即上传成功时才返回该项。
* desc：错误信息，只有code为4001时才返回该项。

### 2.2.4查询接口参数

* dsp\_token：字符串类型，用来唯一标识合作方，必填。
* m\_id：上传成功时返回的素材id，查询单个素材时使用，必填。
* batch：多个m\_id，用逗号分隔，批量查询素材时使用，必填。
* status：素材状态，按素材状态查询素材列表时使用，必填。
* start\_date：状态修改的起始时间，按素材状态查询素材列表时使用，格式为yyyy-MM-dd，选填。
* end\_date：状态修改的截止时间，按素材状态查询素材列表时使用，格式为yyyy-MM-dd，选填。

### 2.2.5查询接口返回

接口以json格式返回。

* code：状态码，定义同上传接口。
* status：素材处理状态，定义如下：

INIT 上传成功，处理中

AUDIT\_WAIT 处理成功，等待审核

AUDIT\_UNPASS 审核未通过

COMPLETE 审核通过，可以使用

OFF 投放下线

* reason：审核未通过原因，当status为AUDIT\_UNPASS或OFF时返回该项。
* tv\_id：爱奇艺奇谱id，只有status为COMPLETE时才返回该项。tv\_id作为后续广告竞价的crid参数值。
* ad\_attr: 素材标签，只有status为COMPLETE时才返回该项
* industry\_type：广告主行业类型，只有status为COMPLETE时才返回该项。

## 2.3调用示例

### 2.3.1上传示例

Java实现方式如下：

//上传接口地址

String uploadUrl = “http://127.0.0.1/post/”;

String dspToken = “testToken”;

String fileName = “test.mp4”;

Integer adType = 2;

String platform = “2”;

String clickUrl = “http://127.0.0.1/test/”;

Long adId = 1234567;

String videoId = “123456789”;

String tag = “270”

String endDate = “20150618”;

String appleId = “123456789”

Integer creativeMode = 2 //Interactive

//上传文件内容

byte[] content = {};

DefaultHttpClient client = new DefaultHttpClient();

try {

HttpPost post = new HttpPost(uploadUrl);

post.addHeader(“dsp\_token”, dspToken);

post.addHeader(“file\_name”, URLEncoder.encode(fileName, “UTF-8”));

post.addHeader(“ad\_type”, String.valueOf(adType));

post.addHeader(“apple\_id”, appleId);

post.addHeader(“creative\_mode”, String.valueOf(creativeMode));

post.addHeader(“ad\_type”, String.valueOf(adType));

post.addHeader(“platform”, platform);

post.addHeader(“click\_url”, URLEncoder.encode(clickUrl, “UTF-8”));

if (adId > 0) {

post.addHeader(“ad\_id”, String.valueOf(adId));

}

post.addHeader(“video\_id”, videoId);

post.addHeader(“tag”, tag);

post.addHeader(“end\_date”, endDate);

try {

post.setEntity(new ByteArrayEntity(content));

HttpResponse response = client.execute(post);

} catch (ClientProtocolException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

} finally {

post.releaseConnection();

}

} catch (Exception e) {

e.printStackTrace();

}

### 2.3.2上传返回

* 成功:

{

“code”: “0”,

“m\_id”: “710000003”

}

* 失败:

{

“code”: “1001”

}

### 2.3.3查询示例

http://220.181.184.220/upload/api/query?dsp\_token=xxxxxxx&m\_id=210710000003

### 2.3.4查询返回

* 待审核时:

{

“code”: “0”,

“status”: “AUDIT\_WAIT”

}

* 处理结束可使用时:

{

“code”: “0”,

“status”: “COMPLETE”,

“tv\_id”: “13005709”,

"ad\_attr": [

"少儿不宜"

],

"industry\_type": "游戏"

}

### 2.3.5批量查询示例

http://220.181.184.220/upload/api/batchQuery?dsp\_token=xxxxxxx&batch=1005,1006

### 2.3.6批量查询返回

{

“code”: 0,

“results”: [

{

“m\_id”: “1005”,

“status”: “AUDIT\_WAIT”

},

{

“m\_id”: “1006”,

“status”: “COMPLETE”,

“tv\_id”: “100002709，

"ad\_attr": [

"少儿不宜"

],

"industry\_type": "游戏"

}

]

}

### 2.3.7批量查询示例

http://220.181.184.220/upload/api/queryByStatus?dsp\_token=xxxxxxx&status=COMPLETE&start\_date=2014-12-10

### 2.3.8按状态查询返回

返回符合条件的mid数组，如：

{

“code”: 0,

“results”: [

710000003,

710000005

]

}

# 3.竞价API

## 3.1概述

QAX(iqiyi ad exchange) 是一套供第三方广告提供商和DSP(Demand-side Platforms,广告需求平台)使用的实时竞价广告平台。它可以公开透明的进行广告竞价，流量提供商和广告商可以选择并统计竞价广告的投递情况。

## 3.2账户要求

**加入QAX的DSP需要提供以下属性**

* DSP实时竞价服务URL，例如: [http://www.thirdDSP.com/bid](http://www.thirddsp.com/bid)
* DSP的cookie mapping服务URL（即跳转地址），例如:

<http://www.thirdparty.com>

* DSP竞价服务所能承受的的最QPS(Queries Per Second)
* 竞价类型(目前默认只支持CPM)

## 3.3准备工作

QAX使用标准HTTP POST方法请求DSP的竞价服务，protobuf作为HTTP body数据格式，Content-type设置为"application/x-protobuf"。

如果在一次请求中，DSP不参与竞价，则直接返回http 204即可，不必返回BidResponse。

## 3.4 QAX请求DSP的竞价接口（BidRequest）

### 3.4.1 接口参数说明

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **参数说明** |
| id | int | request id，即爱奇艺请求id，此id值不唯一，存在重复可能性 |
| use id | string | 用户id，详细命名规则见附录一 |
| site | object | 站点信息 |
| content | object | 网站内容 |
| title | string | 标题 |
| url | string | 网址 |
| keyword | string | 关键字 |
| len | int | 视频时长 |
| album\_id | int | 剧目信息 |
| channel\_id | int | 频道信息，channel\_id字典表见excel文档 |
| device | object | 设备信息 |
| ua | string | User agent |
| ip | string | 设备ip地址，爱奇艺使用广协ip库 |
| geo | object | 地理位置信息，爱奇艺使用广协ip库 |
| country | int | 国家信息，爱奇艺使用广协ip库 |
| metro | int | 省份信息，爱奇艺使用广协ip库 |
| city | int | 城市信息，爱奇艺使用广协ip库 |
| connection\_type | int | 网络接入字段，connection\_type字典表见excel文档 |
| platform\_id | int | 平台信息，platform\_id字典表见excel文档 |
| android\_ia | Int | Android设备的专用id信息，非必须发送字段 |
| Imp | object | 广告返回时的Bid.impid应该对应Imp.id（一次session中唯一） |
| video | object | 视频内容 |
| ad\_zone\_id | int | 广告位id，ad\_zone\_id字典表见excel文档 |
| linearity | int | 是否是串行在正片中的广告，1表示是，2不是 |
| minduration | int | 广告最小时长 |
| maxduration | int | 广告最大时长 |
| protocol | int | 协议版本号 |
| startdelay | int | 本次广告自第几秒开始 |
| ad\_type | int | 广告类型，ad\_type字典表见excel文档 |
| bidfloor |  | CPM底价，单位：分 |
| campaign\_id | int | 交易id，DSP和QAX间通常会有几个campaign |
| is\_pmp | bool | 用于区分PDB和RTB流量，仅在PDB流量中有此字段，true为PDB流量 |

### 3.4.2 BidRequest示例

The bid request for testdsp campaign 61000000 is

id: "133"

user {

id: "bd98f6deb05f6256ca4028799fb42710"

}

site {

id: 1

content {

title: "花千骨第10集"

url: "www.iqiyi.com"

keyword: "内地"

keyword: "言情剧"

keyword: "古装剧"

keyword: "偶像剧"

keyword: "地区"

keyword: "类型"

keyword: "配音语种"

keyword: "国语"

keyword: "剧情"

keyword: "题材"

keyword: "虐恋"

keyword: "师生恋"

keyword: "仙侠"

keyword: "网络小说改编"

keyword: "内地剧场"

keyword: "华语剧场"

keyword: "最新更新"

keyword: "一周最热节目单"

keyword: "高颜值开启返古狂潮"

keyword: "内地"

keyword: "言情剧"

keyword: "古装剧"

keyword: "偶像剧"

keyword: "地区"

keyword: "类型"

keyword: "配音语种"

keyword: "国语"

keyword: "剧情"

keyword: "题材"

keyword: "虐恋"

keyword: "师生恋"

keyword: "仙侠"

keyword: "网络小说改编"

keyword: "内地剧场"

keyword: "华语剧场"

keyword: "最新更新"

keyword: "一周最热节目单"

keyword: "高颜值开启返古狂潮"

len: 2214

album\_id: 202321601

channel\_id: 2

}

}

device {

ua: "WinHttpClient"

ip: "10.1.243.163"

geo {

country: 1

metro: 100

city: 10000

}

connection\_type: 0

platform\_id: 11

android\_id: "879beada72645b29"

}

imp {

id: "17"

video {

ad\_zone\_id: 1000000000381

linearity: 1

minduration: 15

maxduration: 60

protocol: 3

startdelay: 5

ad\_type: 1

}

bidfloor: 800

campaign\_id: 61000000

is\_pmp=false

}

## 3.5 DSP返回给QAX的竞价结果接口(Bid Response)

### 3.5.1 接口参数说明

Impid：与request中的imp.id对应，且保持一致；

price：价格，单位分/CPM；

crid：即tv\_id,素材审核通过后获取,请务必不要使用m\_id。

### 3.5.2 BidResponse示例

id: "133"

seatbid {

bid {

id: "0"

impid: "17"

price: 800

adm: "<VAST xmlns:MM=\"http://www.iqiyi.com/cupid\" version=\"3.0\"><Ad id=\"1\"><InLine><AdSystem><![CDATA[]]></AdSystem><AdTitle><![CDATA[VAST Linear\_1]]></AdTitle><Description><![CDATA[VAST Linear\_1]]></Description><Impression id=\"Impression0\"><![CDATA[http://pmptrack-iqiyi.gentags.net/iqiyi/win?&win=${SETTLEMENT}&BIDID=133&dealid=61000000&pid=1000000000381&ad\_type=1&ad\_id=1&crt\_id=1&UniquID=bd98f6deb05f6256ca4028799fb42710&region=1000000000&city=1000000000]]></Impression><Impression id=\"Impression1\"><![CDATA[http://www.hylinkad.com/imp?test=1]]></Impression><Creatives><Creative id=\"1\" AdID=\"1\" sequence=\"1\"><Linear><Duration>00:00:15</Duration><TrackingEvents></TrackingEvents><VideoClicks><ClickThrough type=\"0\"><![CDATA[http://www.hylinkad.com]]></ClickThrough><ClickTracking><![CDATA[http://pmptrack-iqiyi.gentags.net/iqiyi/ck?&BIDID=133&dealid=61000000&pid=1000000000381&ad\_type=1&ad\_id=1&crt\_id=1&UniquID=bd98f6deb05f6256ca4028799fb42710&region=1000000000&city=1000000000]]></ClickTracking><ClickTracking><![CDATA[http://www.hylinkad.com/clk?test=1]]></ClickTracking></VideoClicks><Icons><Icon><StaticResource><![CDATA[*http://www.iqiyi.com*]]></StaticResource></Icon></Icons></Linear></Creative></Creatives></InLine></Ad></VAST>"

crid: "100075648"

}

}

### 3.5.3 vast示例

<VAST version="3.0">

<Ad>

<InLine>

<!--

-->

<Impression>

<![CDATA[]]>

</Impression>

<Creatives>

<Creative>

<Linear>

<VideoClicks>

<!--

-->

<ClickThrough type="0">

<![CDATA[]]>

</ClickThrough>

<!--

-->

<ClickTracking>

<![CDATA[]]>

</ClickTracking>

</VideoClicks>

<!--

-->

<TrackingEvents>

<Tracking event="start">

<![CDATA[]]>

</Tracking>

<Tracking event="firstQuartile">

<![CDATA[]]>

</Tracking>

<Tracking event="midpoint">

<![CDATA[]]>

</Tracking>

<Tracking event="thirdQuartile">

<![CDATA[]]>

</Tracking>

<Tracking event="complete">

<![CDATA[]]>

</Tracking>

</TrackingEvents>

<Icons>

<!--

-->

<Icon>

<StaticResource>

<![CDATA[*http://www.iqiyi.com*]]>

</StaticResource>

</Icon>

</Icons>

</Linear>

</Creative>

</Creatives>

</InLine>

</Ad>

</VAST>

### 3.5.4vast相关说明

#### 3.5.4.1 Impression（曝光监测） 节点使用规则：

1. 支持添加多个**Impression 节点**，即支持多条曝光监测 URL，条数建议5条内，每条长度不超过1050个字符；
2. **曝光监测 URL** 请填写到 CDATA 里
3. 若**没有曝光监测需求**，可以没有 Impression 节点或 CDATA 里内容为空
4. **HTTPS 协议的监测链接**：所有平台均不支持

5.**302 重定向**

1. PC 端、H5 支持 302 重定向
2. 移动客户端、TV 端不支持302 重定向
3. **移动客户端 MMA SDK 监测**：只要监测链接域名包含附录二-2中任意一家，都会使用 MMA SDK 发送监测，即链接域名决定了是否使用 MMA SDK，不存在额外的开启 SDK 操作，其他域名不会使用 MMA SDK 发送
4. **用户 IP 回传**：PC 端、H5、移动客户端、TV 端都支持，需要在监测链接里加上宏：\_\_IP\_\_，该宏会被自动替换成用户 IP 值

8.**移动客户端、TV 端操作系统信息回传**

1. 移动端：若监测链接可以使用 MMA SDK，则 SDK 发送监测时会自动带上操作系统的标志值（见附录二-3）
2. 移动端使用 API 监测，需要在监测链接里加上宏：\_\_OS\_\_，该宏会被自动替换成操作系统的标志值（见附录二-3）
3. TV 端使用 API 监测，需要在监测链接里加上宏：\_\_OS\_\_，该宏会被自动替换成操作系统的标志值（见附录二-3）

9.**移动客户端、TV 端设备 id 回传**

1. 移动端：若监测链接可以使用 MMA SDK，则 SDK 发送监测时会自动带上设备 id
2. 移动端使用 API 监测，需要在监测链接里加上宏：\_\_UDID\_\_，该宏会被自动替换成设备 id，取值规则见附录二-1
3. TV 端使用 API 监测，需要在监测链接里加上宏：\_\_UDID\_\_，该宏会被自动替换成设备 id，取值规则见附录二-1

10.**移动客户端是否支持 MMA API 监测宏替换标准**：支持，具体的宏和替换值见附录二-4

#### 3.5.4.2 ClickThrough（点击跳转）节点使用规则（TV端不支持点击跳转）：

1. 一次竞价返回里最多只能有一个 **ClickThrough 节点**
2. **点击跳转 URL** 请填写到 CDATA 里
3. 若**没有点击跳转需求**，可以没有 ClickThrough 节点或 CDATA 里内容为空，TV端不支持点击跳转；
4. 既支持 `点击跳转` 和 `点击监测` 串成一条 URL 填写到 ClickThrough 节点上；也支持分开填写，即 `点击跳转` 填写到 ClickThrough 节点上，`点击监测` 填写到 ClickTracking 节点上；但移动端广告并且点击监测需要由 MMA SDK 发送，请务必分开填写

5 **HTTPS 协议**的跳转链接:iOS 上跳转 AppStore下载页面要求使用 HTTPS 链接外，其他场景均不支持。

6 **302重定向** –各端均支持。

7. **iOS 上跳转 AppStore 下载**页面

1. H5 以下链接格式都支持；
2. 移动客户端以下链接格式在最新版本（APP 7.3，目前占比70%左右）的爱奇艺客户端上都是可以正常跳转的，但建议使用格式a（<https://itunes.apple.com/cn/app/id1071517766?mt=8>），后面3种格式在老版本的爱奇艺客户端上无法正常跳转
   * 1. <https://itunes.apple.com/cn/app/id1071517766?mt=8>
     2. <https://itunes.apple.com/cn/app/id1071517766>
     3. <https://itunes.apple.com/cn/app/jian-yu-mo-fa-3dmmo-lang-man/id1071517766?mt=8>
     4. <https://itunes.apple.com/cn/app/jian-yu-mo-fa-3dmmo-lang-man/id1071517766>

8.**移动客户端 MMA SDK 监测** ：点击跳转是打开广告落地页或下载 APP，没有 SDK 监测这一说

9**.用户 IP 回传** ：PC 端、H5、移动客户端都支持，需要在点击跳转链接里加上宏：\_\_IP\_\_，该宏会被自动替换成用户 IP 值，例如替换前：[http://a.b.c/t?ip=\_\_IP\_\_，替换后：http://a.b.c/t?ip=1.2.3.4](http://a.b.c/t?ip=__IP__%EF%BC%8C%E6%9B%BF%E6%8D%A2%E5%90%8E%EF%BC%9Ahttp://a.b.c/t?ip=1.2.3.4)

10.**移动客户端操作系统信息回传**：支持，需要在点击跳转链接里加上宏：\_\_OS\_\_，该宏会被自动替换成操作系统的标志值（见附录二-3），例如替换前：[http://a.b.c/t?os=\_\_OS\_\_，替换后：http://a.b.c/t?os=1](http://a.b.c/t?os=__OS__%EF%BC%8C%E6%9B%BF%E6%8D%A2%E5%90%8E%EF%BC%9Ahttp://a.b.c/t?os=1)

**11.移动客户端设备 id 回传**：支持，需要在点击跳转链接里加上宏：\_\_UDID\_\_，该宏会被自动替换成设备 id，取值规则见附录二-1，例如替换前：[http://a.b.c/t?udid=\_\_UDID\_\_，替换后：http://a.b.c/t?udid=1E2DFA89-496A-47FD-9941-DF1FC4E6484A](http://a.b.c/t?udid=__UDID__%EF%BC%8C%E6%9B%BF%E6%8D%A2%E5%90%8E%EF%BC%9Ahttp://a.b.c/t?udid=1E2DFA89-496A-47FD-9941-DF1FC4E6484A)

**12.移动客户端是否支持 MMA API 监测宏替换标准**：支持，具体的宏和替换值见附录二-4

#### 3.5.4.3 type（点击跳转类型） 属性使用规则：

1. type 属性只对移动端广告有效

1. 若对于点击跳转类型没有要求，可以没有 type 属性，或者 type 属性取默认值"0"

3. type 属性各个取值定义如下：

1. "0"：默认类型，点击后通过内置 WebView 打开落地页，绝大部分广告都使用这个类型
2. "4"：APP 下载，点击后弹出对话框询问用户是否下载 APP，Android 用户选择下载则开始下载 APP，iOS 用户选择下载则跳转到 App Store，适用于游戏应用广告（推荐IOS端 APP下载推广使用）
3. "11"：APP 直接下载。

安卓端直接下载 APP，处理逻辑和类型"4"相同，唯一的区别是不会弹出对话框询问用户是否下载 APP，同样适用于游戏应用广告。注意：Android APP 下载链接在投放上限制只能用 .apk 地址，不能带第三方监测，若一定要使用监测，需知晓会存在重复记录点击请求的问题。

iPhone直接跳转至APP Store下载页面。注意：由于点击跳转是依据APP id字段拉取的APP Store下载页面，不调用素材落地链接，故包在落地链接的点击监测失效，如需数据监测只能将点击监测地址放在Vast ClickTracking CDATA中。

#### 3.5.4.4 ClickTracking（点击监测）节点使用规则（TV端不支持点击监测）：

1. 支持添加多个 **ClickTracking 节点**，即支持多条点击监测 URL，条数建议5条内，每条长度不超过1050个字符；

1. **点击监测 URL** 请填写到 CDATA 里；
2. 若**没有点击监测需求**，可以没有 ClickTracking 节点或 CDATA 里内容为空

4. miaozhen.com、admaster.com.cn、cr-nielsen.com、mma.ctrmi.com 的点击监测都默认由 MMA SDK 发送

注意：需要由 MMA SDK 发送的点击监测必须填写在 ClickTracking 节点上；

**5. HTTPS 协议的监测链接** ：PC 端、H5、移动客户端均不支持

6.**302 重定向**

1. PC 端、H5 都支持 302 重定向;
2. 移动客户端不支持302 重定向.

7.**移动客户端 MMA SDK 监测**：只要监测链接域名包含附录二-2中任意一家，都会使用 MMA SDK 发送监测，即链接域名决定了是否使用 MMA SDK，不存在额外的开启 SDK 操作，其他域名不会使用 MMA SDK 发送

8**.用户 IP 回传** ：PC 端、H5、移动客户端都支持，需要在监测链接里加上宏：\_\_IP\_\_，该宏会被自动替换成用户 IP 值

9.**移动客户端操作系统信息回传**

1. 若监测链接可以使用 MMA SDK，则 SDK 发送监测时会自动带上操作系统的标志值（见附录二-3）
2. 否则，使用 API 监测，需要在监测链接里加上宏：\_\_OS\_\_，该宏会被自动替换成操作系统的标志值（见附录二-3）

10.**移动客户端设备 id 回传**

1. 若监测链接可以使用 MMA SDK，则 SDK 发送监测时会自动带上设备 id
2. 否则，使用 API 监测，需要在监测链接里加上宏：\_\_UDID\_\_，该宏会被自动替换成设备 id，取值规则见附录二-1

11.**移动客户端是否支持 MMA API 监测宏替换标准**：支持，具体的宏和替换值见附录二-4

#### 3.5.4.5 TrackingEvents（监测事件）节点使用规则：

1. 支持 start、firstQuartile、midpoint、thirdQuartile、complete 这5个监测事件，只对前中后贴片广告有效

2. 每个事件支持多个 Tracking 节点，比如：若需要对 start 事件添加多条监测 URL，可以通过添加多个 event="start" 的 Tracking 节点实现，条数没有上限

3. 相应事件的监测 URL 请填写到 CDATA 里

4. 请按照需要添加相应的监测事件节点，比如：若只需要监测 start 事件，则只添加 event="start" 的 Tracking 节点即可

5. 若没有任何事件监测需求，可以没有 TrackingEvents 节点

6. miaozhen.com、admaster.com.cn、cr-nielsen.com、mma.ctrmi.com 的点击监测都默认由MMA SDK发送

#### 3.5.4.6 Icon（DSP Logo）节点使用规则：

1. 一次竞价返回里最多只能有一个 Icon 节点。

2. 请在 CDATA 里填写 DSP Logo 的图片地址，DSP需将符合要求的logo图片给到QAX，由QAX反馈图片地址。（详见response示例及vast示例标黄部分）

LOGO要求：25\*25尺寸的JPEG或PNG、图片清晰、识别度高

3. DSP Logo 会在广告左下角和广告一同展现

4. 此项为必填项，若返回为空或url错误，则DSP本次竞价失败

### 3.5.5 曝光监测接口与解密方式

此信息由广告展示页面返回给DSP，DSP可以灵活的选择竞价结果的接受方式, 只要使用到反馈宏并且不影响广告展现即可,基本规则如下:

1. 此反馈结果只有在广告展现成功后才会返回
2. 当前只返回一个竞价成功的反馈结果
3. 每个DSP有一个唯一的解密密钥, 密钥由十六进制字符构成的长度为32个字节的字符串，使用时请将其转化为16字节的二进制数组。

加密算法是 aes 128位的ecb模式，padding方式为 PKCS。测试用例如下：  
密钥（token）: 46356afe55fa3cea9cbe73ad442cad47 （使用时，请将hex string转化为对应的128位二进制序列）  
加密后的价格密文: 40dd88e115aab34ffa949dfb245e8e97 (bytes, 此处转换成hex string仅供显示）  
加密前的价格明文: 1234567890

必须明确的一点是，加密/解密是针对price来进行的，解密示例函数详见附件[解密价格函数示例.docx](http://wiki.qiyi.domain/download/attachments/19170095/%E8%A7%A3%E5%AF%86%E4%BB%B7%E6%A0%BC%E5%87%BD%E6%95%B0%E7%A4%BA%E4%BE%8B.docx?version=2&modificationDate=1409047430000)。

返回参数中必须带有 **${SETTLEMENT}**宏。

下面表格的message是settlement的proto定义，其中price字段由加密后的价格密文填充。

在竞价成功后， **${SETTLEMENT}**宏将被settlement序列化再经过base64编码后的字符串替换。

|  |
| --- |
| message Settlement {  // Algorithm for price and auth.  optional uint32 version = 1;   // Encrypted price.  required bytes price = 2;   // Authentication information for bidid and price.  optional bytes auth = 3;  } |

**注**：我们使用的**base64编码字典表是自己定制的**，不要误认为跟其他常用的编码表是一致的。字典表如下：

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789\_- （填充字符是!，不是=）。

一个经过base64编码过的Settlement的例子是，CPPiKxIQQN2I4RWqs0-6lJ37JF6Olw!!

解码出来的message中，price的值是，40dd88e115aab34ffa949dfb245e8e97，此值再经过解密可得价格明文。

auth字段（32个字符组成的字符串）是一个校验值，它是由三个字段拼接起来的二进制块，做标准MD5运算得到的。

这三个字段分别是：Bid.id，Settlement.price和token（用于解密的那个）。

1. 解密函数示例

const std::string DecryptPrice(const std::string& cipher, const std::string& key) {

char buffer[kPriceCipherSize];

int price\_length;

EVP\_CIPHER\_CTX ctx;

EVP\_DecryptInit(&ctx,

EVP\_aes\_128\_ecb(),

reinterpret\_cast<const uint8\_t\*>(key.data()),

nullptr);

EVP\_DecryptUpdate(&ctx,

reinterpret\_cast<uint8\_t\*>(buffer),

&price\_length,

reinterpret\_cast<const uint8\_t\*>(cipher.data()),

kPriceCipherSize);

EVP\_DecryptFinal(&ctx,

reinterpret\_cast<uint8\_t\*>(buffer),

&price\_length);

EVP\_CIPHER\_CTX\_cleanup(&ctx);

return std::string(buffer, price\_length);

}

注：cipher是加密后的价格密文，key是相应的token转化而成的16字节的二进制数组。

kPriceCipherSize为相应加密价格密文的长度，当前的策略下是16。

函数返回值是解密后的价格明文。

# 4. Cookie Matching Service API

## 4.1 Base URL

**base url:** <http://ckm.iqiyi.com/pixel>  
**Paramerter:**

|  |
| --- |
| **Parameter** |
| **Description** |
| qiyi\_nid | Network ID of DSP , （该ID由QAX负责分配） |
| qiyi\_sc | 1. 当没有iqiyi.com域下的cookie（即爱奇艺无法识别的用户）时，设置cookie。  2.value被忽略无论有值与否。 |

## 4.2 Redirect URL

|  |
| --- |
| **Parameter** |
| **Description** |
| qiyi\_uid | qiyi User ID.  处理成功后，会在DSP的cookie mapping服务URL后添加此参数，并做302跳转。 |

# 附录一：【user id】命名规则

Android:  
【android id】: ANDROID\_ID是Android系统第一次启动时产生的一个64bit数，示例： 63cfb1479595ab56, 注意：获取此值时需要动态检查一下权限  
【mac address】:加密, *加密方式：去冒号分隔符，转大写，然后MD5。*MAC地址共48位（6个字节），十六进制，示例： 34:23:BA:96:D3:E9， 注意：Android 6.0 以后返回的都是02:00:00:00:00:00  
【imei】加密，移动设备国际识别码: 15位数字或32位字母+数字，使用MD5方式加密。

IOS:

【OpenUDID】通用唯一识别码Unique Device Identifier: 苹果IOS设备的唯一识别码，由40个字母和数字的组合组成，示例： 0d943976b24c85900c764dd9f75ce054dc5986ff

【idfa】: 广告标识符 ，128位（16个字节）， 示例: 1E2DFA89-496A-47FD-9941-DF1FC4E6484A

【mac\_address】: 加密, *加密方式：去冒号分隔符，转大写，然后MD5。*MAC地址共48位（ 6个字节），ios7 以后禁用， ios7及以后的版本调用接口返回的都是02:00:00:00:00:00

移动端 user id 取值规则：

* 1. iOS 上，先取 IDFA，若 IDFA 不可用，则取 openUDID，若 openUDID 不可用，则取 Mac 地址
  2. Android 上，先取 IMEI，若 IMEI 不可用，则取 AndroidID，若 AndroidID 不可用，则取 Mac 地址

# 附录二：第三方监测说明补充信息

1. 移动客户端回传设备 id 取值规则：
   1. iOS 上，先取 IDFA，若 IDFA 不可用，则取 openUDID，若 openUDID 不可用，则取 Mac 地址
   2. Android 上，先取 IMEI，若 IMEI 不可用，则取 AndroidID，若 AndroidID 不可用，则取 Mac 地址
   3. TV 上，取 Mac 地址
2. 移动客户端可以使用 MMA SDK 监测的域名标识
   1. [miaozhen.com](http://miaozhen.com/)
   2. [admaster.com.cn](http://admaster.com.cn/)
   3. [cr-nielsen.com](http://cr-nielsen.com/)
   4. [mma.ctrmi.com](http://mma.ctrmi.com/)
3. 移动客户端操作系统标志值
   1. 0表示Android
   2. 1表示iOS
   3. 2表示Windows
   4. 3表示其他
4. MMA API 监测宏替换定义：
   1. \_\_OS\_\_：操作系统，替换值见附录3
   2. \_\_IMEI\_\_：MD5 加密
   3. \_\_MAC\_\_：去分隔符“:”，转大写，然后 MD5 加密
   4. \_\_MAC1\_\_：保留分隔符“:”，转大写，然后 MD5 加密
   5. \_\_IDFA\_\_：取原值
   6. \_\_OPENUDID\_\_：取原值
   7. \_\_ANDROIDID\_\_：MD5 加密
   8. \_\_IP\_\_：取原值
   9. \_\_TS\_\_：UTC 时间戳，自1970年起的毫秒数