

爱奇艺程序化购买 技术对接文档

V2.5

目录

1. 广告主上传接口	4
1.1 接口描述	4
1.2 接口定义	4
1.2.1 接口地址	4
1.2.2 上传及更新接口参数	4
1.2.3 上传接口返回	5
1.2.4 状态查询参数	5
1.2.5 状态查询返回	5
1.3 调用示例	5
1.3.1 上传及更新示例	5
1.3.2 上传返回	6
1.3.3 状态查询示例	7
1.3.4 状态查询返回	7
1.3.5 状态批量查询示例	7
1.3.6 批量查询返回	7
2. 素材上传接口	9
2.1 接口描述	9
2.2 接口定义	9
2.2.1 接口地址	9
2.2.2 上传接口参数	9
2.2.3 上传接口返回	11
2.2.4 查询接口参数	11
2.2.5 查询接口返回	12
2.3 调用示例	12
2.3.1 上传示例	12
2.3.2 上传返回	13
2.3.3 查询示例	13
2.3.4 查询返回	14
2.3.5 批量查询示例	14
2.3.6 批量查询返回	14
2.3.7 批量查询示例	15
2.3.8 按状态查询返回	15
3. 竞价 API	15
3.1 概述	15
3.2 账户要求	15
3.3 准备工作	15
3.4 QAX 请求 DSP 的竞价接口 (BidRequest)	16
3.4.1 接口参数说明	16
3.4.2 BidRequest 示例	17
3.4.3 机型解密规则	19

3.5 DSP 返回给 QAX 的竞价结果接口(Bid Response)	20
3.5.1 接口参数说明	20
3.5.2 BidResponse 示例	20
3.5.3 vast 示例	21
3.5.4vast 相关说明	22
3.5.5 曝光监测接口与解密方式	26
4. Cookie Matching Service API	29
4.1 Base URL.....	29
4.2 Redirect URL.....	29
附录一：【user id】命名规则.....	30
附录二：第三方监测说明补充信息.....	31

版本信息

版本/状态	备注
V2.4	<ol style="list-style-type: none">1. 更换广告主上传及素材上传服务器地址；详见 1.2.12. 接口参数说明中增加了字段 model, os 和 osv；详见 3.4.13. 增加了机型解密规则；详见 3.4.34. 增加 floor_price 字段按不同行业给 DSP 方返回底价；详见 3.4.15. 增加点击跳转类型 type=11 的单线程下载方式；详见 3.5.4.3

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://220.181.109.203/upload/advertiser>

广告主审核状态查询：<http://220.181.109.203/upload/api/advertiser>

广告主状态批量查询：<http://220.181.109.203/upload/api/batchAdvertiser>

注 1: 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。

注 2: 原服务器 220.181.184.220 会并行至年底，2017 年 1 月 1 日起正式停用。

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://220.181.109.203/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://220.181.109.203/upload/api/batchAdvertiser?dsp_token=xxxxx
xx&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://220.181.109.203/upload/post>

查询接口: <http://220.181.109.203/upload/api/query>

批量查询: <http://220.181.109.203/upload/api/batchQuery>

批量查询: <http://220.181.109.203/upload/api/queryByStatus>

注 1: 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。

注 2: 原服务器 220.181.184.220 会并行至年底，2017 年 1 月 1 日起正式停用。

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://220.181.109.203/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://220.181.109.203/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://220.181.109.203/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`
- ✓ 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
model	string	型号，如“vivoX7”，部分型号有加密，具体可见 3.4.3 机型解密规则
os	string	操作系统，取值如下：ios, windows, android
osv	string	操作系统版本号，如“5.1.1”
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_id	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。
floor_price		不同行业设置不同底价。当在 floor_price 找不到所需要行业的底价的时候，就用 bidfloor 作为当前的流量的底价
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"
  model: " iPhone9,1"
  os: " android "
  osv: "5.1.1"
  geo {
    country: 1
    metro: 100
    city: 10000
  }
  connection_type: 0
  platform_id: 11
  android_id: "879beada72645b29"
}
imp {
  id: "17"
  video {
    ad_zone_id: 10000000000381
    linearity: 1
    minduration: 15
    maxduration: 60
    protocol: 3

```

```

        startdelay: 5
        ad_type: 1
    }
    bidfloor: 800
    floor_price:[
        {
            industry:400000000,
            price:1200
        },
        {
            industry:100000000,
            price:400
        },
        {
            Industry:200000000,
            Price:1000
        }
    ]
    campaign_id: 61000000
    is_pmp:ture
}

```

3.4.3 机型解密规则

ios 端和 android 端的机型信息都做了 URI Encode，所以这两个端的机型信息都需要做下 URI Decode 才行。

1. Android 端所有广告类型，机型信息的取值规则是取的：
android.os.Build.Model
2. iOS 端存在两种形式：
 - (1)一种是直接取的： DeviceModel()
 - (2)一种是拼接的： QIYIVideo/ app 版本号 (iOS;包名;iOS 系统版本;设备型号) Corejar， 其中括号内最后的设备型号就是 DeviceModel()， 需要自行分解。

例子：（URI decode 之后的结果）

Android: “vivo Y51”

ios: “iPhone9,1”

“QIYIVideo/7.9.1 (iOS;com.qiyi.iphone;iOS9.0.2;iPhone6,1) Corejar”

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><![CDAT
A[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=61
000000&pid=1000000000381&ad_type=1&ad_id=1&crt_id=1&UniquID=bd98f6
deb05f6256ca4028799fb42710&region=1000000000&city=1000000000]]></I
mpression><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><TrackingEvent
s></TrackingEvents><VideoClicks><ClickThrough
type=\"0\"><![CDATA[http://www.hylinkad.com]]></ClickThrough><Clic
kTracking><![CDATA[http://pmptrack-
iqiyi.gentags.net/iqiyi/ck?&BIDID=133&dealid=610000000&pid=10000000
00381&ad_type=1&ad_id=1&crt_id=1&UniquID=bd98f6deb05f6256ca4028799
fb42710&region=1000000000&city=1000000000]]></ClickTracking><Click
Tracking><![CDATA[http://www.hylinkad.com/clk?test=1]]></ClickTrac
king></VideoClicks><Icons><Icon><StaticResource><![CDATA[http://ww
w.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>
          </Linear>
        </Creative>
      </Creatives>
    </InLine>
  </Ad>
</VAST>
```

```

        </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
6. **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
7. **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

7. 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）

8. 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

9. 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

- a) <https://itunes.apple.com/cn/app/id1071517766?mt=8>
- b) <https://itunes.apple.com/cn/app/id1071517766>
- c) <https://itunes.apple.com/cn/app/jian-yu-mo-fa-3dm-mo-lang-man/id1071517766?mt=8>
- d) <https://itunes.apple.com/cn/app/jian-yu-mo-fa-3dm-mo-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>

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>

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>

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
- 2. 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。**目前 android 下载管理器默认为多线程，每个线程失败后会进行重新尝试，故会造成多次请求。如需要单线程下载，可在*.apk 后加上参数“&iqiyidl=1”。**

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；
2. 点击监测 URL 请填写到 CDATA 里；
3. 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. miaozen.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 像素尺寸的 PNG 格式文件，背景色透明、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

1. 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 P27 “解密函数示例”。

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，CPPiKxIQQN2I4RWqs0-6lJ37JF60lw!!

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）。

2. 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>

Parameter:

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：
 - 3) iOS 上，先取 IDFA，若 IDFA 不可用，则取 openUDID，若 openUDID 不可用，则取 Mac 地址
 - 4) Android 上，先取 IMEI，若 IMEI 不可用，则取 AndroidID，若 AndroidID 不可用，则取 Mac 地址
 - 5) 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) miaozen.com
 - 2) admaster.com.cn
 - 3) cr-nielsen.com
 - 4) 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