

汽车之家广告交易平台接入规范

changelist	
v2.4.2	2.2.2 节增加点击宏说明，竞价应答接口素材 id 字段说明；
v2.4.3	2.3.1 增加价格加密说明，2.1.5 增加 app 新模板形式；

目录

1 总体说明	2
1.1 概述	2
1.2 接入准备	2
2 实时竞价	2
2.1 实时竞价接口	2
2.1.1 请求接口	2
2.1.2 请求格式示例	6
2.1.3 应答接口	6
2.1.4 应答格式示例	8
2.1.5 广告形式说明	9
2.2 宏配置说明	15
2.2.1 结算价格宏	15
2.2.2 点击宏	15
2.3 信息处理	15
2.3.1 价格加密	15
2.3.2 设备 ID 加密	16
3 cookie mapping	18
3.1 概述	18
3.2 服务地址	18
3.3 参数说明	18
3.4 示例	18
4 信息同步	19
4.1 报表查询接口	19
4.2 素材审核接口	19
4.2.1 获取素材 API	19
4.2.2 获取素材审核信息 API	22
4.2.3 验签算法	24
5 附录	26

5.1 数据字典	26
----------------	----

1 总体说明

1.1 概述

汽车之家 ADX 是一套供 DSP(Demand Side Platform, 需求方平台) 使用的实时竞价营销平台。通过在该系统中的竞价, DSP 可以根据自身需求实时和自由的选择流量, 并统计投放情况。目前只支持按 CPM 计费。

1.2 接入准备

- ✓ 创建 DSP 账户
- ✓ 获取相关技术文档
- ✓ 获取接入相关的密钥数据
- ✓ 提供竞价 url、提供支持的最大 QPS、cookie mapping 地址
- ✓ 联系运营团队进行素材审核

2 实时竞价

ADX 和 DSP 之间的通信协议使用 HTTP, 请求类型为 POST, 数据格式为 JSON, 请求头中需要设置 Content-Type 为 application/json。

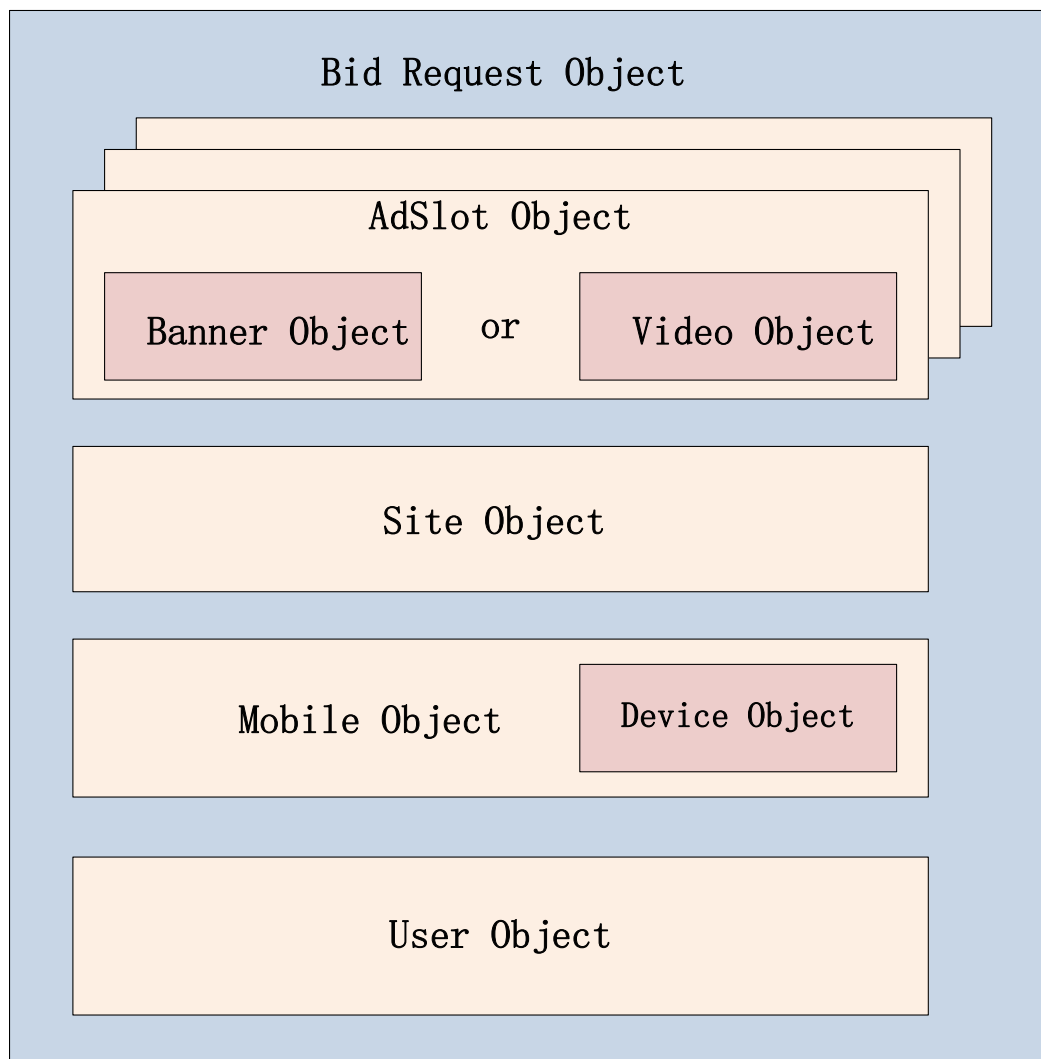
注意: ADX 与 DSP 超时时间为 120ms, 超时比率应该限制在 15%以下

由于 iOS 计划 2017 年实施的传输协议安全政策升级因此对 app 端的广告投放约束如下: 素材地址、曝光监控地址、点击监控地址、落地页地址必须使用 https 协议

2.1 实时竞价接口

2.1.1 请求接口

Bid Request 结构如下图所示:



1. Bid Request Object 说明

Field	Scope	Type	Description
id	required	string	竞价 ID
version	required	string	竞价协议版本
adSlot	required	object	广告位信息
site	optional	object	站点信息
mobile	optional	object	设备信息
user	optional	object	用户信息
is_test	required	bool	是否为测试请求 (true 是 false 否) 如果为测试请求广告不会展现 也不会计费

2. AdSlot Object 说明

Field	Scope	Type	Description
-------	-------	------	-------------

id	required	string	曝光 id (标识每个广告位每一次曝光)
slotid	required	string	广告位 ID
min_cpm_price	required	uint	广告位底价 (分/CPM)
deal_type	required	string	交易类型 (PDB PD PA RTB)
slot_visibility	required	uint	广告位相对位置 0 : 无数据 ; 1-5 : 第 1-5 屏 ; 6 : 第 6 屏及以外
excluded_ad_category	optional	array of uint	禁止的行业类目
banner	optional	object	Banner 类型广告位
video	optional	object	Video 类型广告位

3. Banner Object 说明

Field	Scope	Type	Description
width	required	uint	广告位宽
height	required	uint	广告位高
view_type	required	array of uint	广告位展现形式 5.1 数据字典
templateId	required	array of uint	广告位支持的模板 2.1.5 广告形式说明

4. Video Object 说明 (预留字段)

Field	Scope	Type	Description
-------	-------	------	-------------

5. Site Object 说明

Field	Scope	Type	Description
url	optional	string	当前页面 url
ref	optional	string	Referrer url

6. Mobile Object 说明

Field	Scope	Type	Description
is_app	optional	bool	标识该次广告请求是否来自 app
pkgname	optional	string	包名
device	optional	object	设备信息

7. Device Object

Field	Scope	Type	Description
devicebrand	optional	string	设备品牌
devicemodel	optional	string	设备型号
pm	optional	uint	1-ios,2-android,3-other
os_version	optional	string	操作系统版本
conn	optional	uint	联网方式 0-未知 1-wifi 2:-2g 3-3g 4-4g
networkid	optional	uint	0-代表获取不到运营商 7012 - 中国移动 70121-中国电信 70123-中国联通
lng	optional	uint	经度×1000,000
lat	optional	uint	纬度×1000,000
screen_width	optional	uint	屏幕宽,取设备物理像素
screen_high	optional	uint	屏幕高。取设备物理像素，高度始终大于宽度
screen_orientation	optional	uint	设备横竖屏。可能取值：0， iphone 4s 屏幕正对自己，home 键靠下。90，顺时针旋转 90 度。 180 270；
screen_density	optional	float	屏幕密度，一个逻辑像素等于 几个实际像素；(浮点数)
deviceid	optional	string	设备 id， ios：优先 idfa 如果获取不到就 用 openudid android:imei 加密后的值加密算法见 2.3.2 设备 ID 加密

8. User Object 说明

Field	Scope	Type	Description
id	optional	string	ADX 侧用户 ID
ip	optional	string	用户 ip
user_agent	optional	string	

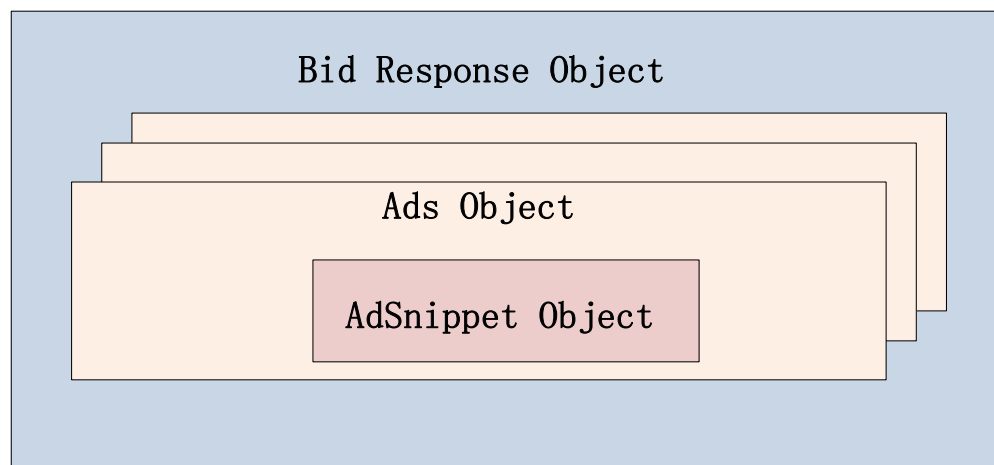
2.1.2 请求格式示例

```
{
  "id": "FSDFKH039484EDLU",
  "version": "1.0.0",
  "is_test": false,
  "adSlot": [
    {
      "id": " PPPVVVVIIIDDD",
      "slotid": "123",
      "min_cpm_price": 500,
      "deal_type": "pa",
      "slot_visibility": 0,
      "excluded_ad_category": [501,502,503],
      "banner": {
        "width":200,
        "height":300,
        "templateId":[110,111],
        "view_type":[1001,1002],
      }
    }
  ]
  "site": {
    "url": "http://www.autohome.com.cn",
    "ref": "http://www.autohome.com.cn/xxx/",
  }
  "user": {
    "id":"MD5COOKIE",
    "ip":"127.0.0.1",
    "user_agent":"Mozilla/5.0 (Windows NT 6.2; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/48.0.2564.109 Safari/537.36",
  }
}
```

2.1.3 应答接口

如果不参与竞价，需要返回一个只有 id、is_cm 和 version 字段的响应,require 字段为必填字段

Bid Response 结构如下图所示：



1. Bid Response Object

Field	Scope	Type	Description
id	required	string	竞价 ID
version	required	string	版本号，需要与请求保持一致
processing_time_ms	optional	uint	dsp 处理时间单位毫秒
is_cm	required	bool	是否 cookie mapping (app 端无效) true 是，false 否
ads	optional	array of object	广告

2. Ads Object

Field	Scope	Type	Description
id	required	string	曝光 id 同请求
slotid	required	string	广告位 id
max_cpm_price	required	uint	最高出价 (PDB PD 请直接返回底价)
creative_id	required	ulong	素材 id 需要使用 素材审核 时返回的素材 id，否则会校验失败
advertiser_id	required	uint	广告主 id
width	required	uint	素材宽(如果是文字链暂不填)
height	required	uint	素材高(如果是文字链暂不填)
category	optional	uint	素材所属行业
creative_type	optional	uint	素材类型(字典定义)
templated	required	uint	从请求中携带的 模板 id 列表 中选择一个模板 id
adsnippet	required	object	广告片段

3. AdSnippet Object

广告片段为 json 格式，不同类型的广告形式 json 字段有所差异，实际字段与 templateId 有关，下表为 templateId = 110 时的字段，其他 templateId 详见 [2.1.5 广告形式说明](#)

Field	Scope	Type	Description
img	required	string	素材 url 地址
pv	required	array of string	曝光监控地址
link	required	string	点击地址， ADX 负责 302 跳转到该地址，如果第二跳不是落地页则第二跳地址需要负责继续跳转到落地页（该字段需要与点击宏结合使用）

2.1.4 应答格式示例

```
{
  "id": "FSDFKH039484EDLU ",
  "processing_time_ms": 99,
  "is_cm": true,
  "ads": [
    {
      "id": "PPPVVVVIIIDDD ",
      "slotid": "123",
      "max_cpm_price": 500,
      "creative_id": 10069,
      "advertiser_id": 1001,
      "width": 200,
      "height": 500,
      "category": 501,
      "creative_type": 1001,
      "templateId": 110,
      "adsnippet": {
        "img": "http://adm.autohome.com.cn/344523.jpg",
        "link": "%CLICK_URL_UNESC%&url=http%3a%5c%2f%5c%2fthird.click.com.cn 点击
```

```

监控地址需要 url_encode",

        "pv":["http://pv.autohome.com.cn","http://pv.dsp.com.cn" ]

    }

]

}

```

2.1.5 广告形式说明

除普通图片模板（110）由广告位尺寸决定素材图片尺寸，其他素材尺寸均由模板决定，详见各模板说明。

● PC、M 端模板（adsnippet）说明

templateId	snippet 示例	说明
110	<pre> adsnippet: { "img": "imgae.url", "link": "click.url", "pv": ["pv.url"] } </pre>	普通图片模板——图片大小受广告位限制
111	<pre> adsnippet: { "link": 'click.url', "img": "imgae.url", "text": "xxxxx", "pv": ["pv.url"] } </pre>	车系综述页图文模板—— img 图片尺寸：height: 104, width: 190 text 文字链长度：min: 26, max: 30
112	<pre> adsnippet: { "link": 'click.url', "img": "imgae.url", "text": "xxxxx", "pv": ["pv.url"] } </pre>	首页论坛图文模板—— img 图片尺寸：height: 92, width: 184 text 文字链长度：min: 26, max: 30
113	<pre> adsnippet: { "link": 'click.url', "img": "imgae.url", "text": "xxxxx", "pv": ["pv.url"] } </pre>	M 端图文—— img 图片尺寸：height: 135, width: 180 text 文字链长度：min: 30, max: 37
114	<pre> adsnippet: { "link": 'click.url', </pre>	M 端图文—— 尺寸组合一：

	<pre> "img": "imgae.url", "text": "xxxxx", "subText": "xxxxx", "pv": ["pv.url"] } </pre>	img 图片尺寸 : height: 120, width: 240 text 文字链长度 : min: 30, max: 37 subText 长度 : min:4, max: 16
		尺寸组合二 : img 图片尺寸 : height: 120, width: 240 text 文字链长度 : min: 20, max: 26 subText 长度 : min:4, max: 16
		尺寸组合三 : img 图片尺寸 : height: 120, width: 120 text 文字链长度 : min: 20, max: 26 subText 长度 : min:50, max: 64

注：文字链说明

text 长度单位为字节

PC 端（除模板 ID111 外）一个汉字对应 2 个字节，一个字母对应 1 个字节

PC 端模板 ID111 一个汉字对应 2 个字节，一个字母对应 1.5 个字节

M 端一个汉字对应 2 个字节，一个字母对应平均 1.5 个字节

● APP 端模板（adsnippet）说明

```

{
  "content": [
    {
      "src": "",
      "type": "", //gif img text bimg simg stext
    }
  ],
  "pv": [], //曝光监控地址
  "link": "", //点击地址（302 跳转）
}

```

模板说明：

- App 端所有广告类型返回素材 adsnippet 模板一致，ADX 将根据 type 的不同组合方式来渲染广告
- 当前汽车之家 ADX 在 app 端点击只支持 302 调转的方式，发生点击 adx 会先跳转到 click 地址，后者负责跳转到落地页

当前支持模板说明：

templateId	snippet 示例	说明
10001	<pre> adsnippet: { "content": [{ "src": "静态图片地址", "type": "img", </pre>	开屏广告模板 100—— img 图片尺寸 : height: 1280, width: 800 gif 图片尺寸 : height: 640, width: 400

	<pre> } { "src": "动态图片地址", "type": "gif", }], "pv": [], //曝光监控地址 "link": "", //点击地址 (302 跳转) } </pre>	注：图片必选，动图可选
10002	<pre> adsnippet: { "content": [{ "src": "小图地址", "type": "simg", } { "src": "大图地址", "type": "bimg", } { "src": "标题内容", "type": "text", }], "pv": [], //曝光监控地址 "link": "", //点击地址 (302 跳转) } </pre>	<p>信息流小图模板——</p> <p>simg 图片尺寸 : height: 112, width: 148 bimg 图片尺寸 : height: 225, width: 400 text 文字链长度 : min: 32, max: 36</p> <p>注：实际展现样式为一图一文，两张图片为了适配不同 app 版本。</p>
10003	<pre> adsnippet: { "content": [{ "src": "图片地址", "type": "img", } { "src": "标题内容", "type": "text", }], "pv": [], //曝光监控地址 "link": "", //点击地址 (302 跳转) } </pre>	<p>行情信息流小图模板——</p> <p>img 图片尺寸 : height: 112, width: 148 text 文字链长度 : min: 32, max: 36</p>

10004	<pre> adsnippet: { "content": [{ "src": "图片地址", "type": "img", } { "src": "标题内容", "type": "text", }], "pv": [], //曝光监控地址 "link": "",//点击地址（302 跳转） } </pre>	行情信息流小图模板—— img 图片尺寸 : height: 225, width: 400 text 文字链长度 : min: 32, max: 36
10005	<pre> adsnippet: { "content": [{ "src": "图片地址", "type": "img", }], "pv": [], //曝光监控地址 "link": "",//点击地址（302 跳转） } </pre>	通栏图片模板—— img 图片尺寸 : height: 120, width: 600
10006 10007	<pre> adsnippet: { "content": [{ "src": "图片地址", "type": "img", }, { "src": "标题内容", "type": "text", } { "src": "副标题内容", "type": "stext", }], "pv": [], //曝光监控地址 "link": "",//点击地址（302 跳转） } </pre>	通栏图文模板—— img 图片尺寸 : height: 100, width: 100 text 文字链长度 : min: 22, max: 26 stext 文字链长度 : min: 46, max: 52

	}	
10008	<pre> adsnippet: { "content": [{ "src": "图片地址", "type": "img", } { "src": "标题内容", "type": "text", }], "pv": [], //曝光监控地址 "link": "", //点击地址 (302 跳转) } </pre>	<p>热门活动图文模板——</p> <p>img 图片尺寸 : height: 120, width: 600 text 文字链长度 : min: 20, max: 30</p>
10009	<pre> adsnippet: { "content": [{ "src": "图片地址", "type": "img", }], "pv": [], //曝光监控地址 "link": "", //点击地址 (302 跳转) } </pre>	<p>APP 焦点图模板——</p> <p>img 图片尺寸 : height: 320, width: 640</p>
10010	<pre> adsnippet: { "content": [{ "src": "图片地址", "type": "img", }], "pv": [], //曝光监控地址 "link": "", //点击地址 (302 跳转) } </pre>	<p>通用+LP 模板——</p> <p>img 图片尺寸 : height: 120, width: 720</p>
10011	<pre> adsnippet: { "content": [{ "src": "图片地址", "type": "img", }] } </pre>	<p>720*240APP 焦点图模板</p> <p>img 图片尺寸 : height: 240, width: 720</p>

	<pre>], "pv": [], //曝光监控地址 "link": "", //点击地址 (302 跳转) } </pre>	
10013	<pre> adsnippet: { "content": [{ "src": "图片地址", "type": "img", }], "pv": [], //曝光监控地址 "link": "", //点击地址 (302 跳转) } </pre>	违章 app_图片模板(1000*200) img 图片尺寸 : height: 200, width: 1000
10014	<pre> adsnippet: { "content": [{ "src": "图片地址", "type": "img", }], "pv": [], //曝光监控地址 "link": "", //点击地址 (302 跳转) } </pre>	违章 app_图片模板 img 图片尺寸 : height: 360, width: 1080
10015	<pre> adsnippet: { "content": [{ "src": "图片地址", "type": "img", }], "pv": [], //曝光监控地址 "link": "", //点击地址 (302 跳转) } </pre>	通栏模板(600*800) img 图片尺寸 : height: 800, width: 600

注：文字链说明

长度单位为字节

一个汉字对应两个字节

一个大写英文字母对应两个字节

一个小写英文字母对应一个字节

一个空格对应一个字节

2.2 宏配置说明

2.2.1 结算价格宏

原型：%%SETTLE_PRICE%%

含义：对于竞价成功的 DSP，可通过该宏获取竞价成功的广告的结算价格。必选。

展开：ADX 会将所有出现的%%SETTLE_PRICE%%替换为加密后的结算价格。如何解密该结算价格加密部分

使用：

该宏可出现在 `adsnippet.pv` 字段中。例如：

```
"pv":["http://pv.autohome.com.cn","http://pv.dsp.com.cn&p=%%SETTLE_PRICE%%"]
```

加密算法及注意事项详见 2.3.1 节

2.2.2 点击宏

原型：%%CLICK_URL_UNESC%%

含义：DSP 必须在竞价的广告中按照一定方式使用点击宏，以便 ADX 跟踪广告的点击情况

展开：该宏会被 ADX 替换为特定的点击字符串。

使用：

该宏必须出现在 `adsnippet.link` 字段中。例如：

```
"link":"%%CLICK_URL_UNESC%%&url=http%3a%2f%2fthird.click.com.cn"
```

点击链接前缀增加点击宏，同时使用 `url` 参数指向 DSP 的点击链接（点击链接需 `url encode`）

2.3 信息处理

2.3.1 价格加密

加密串格式：

initialization_vector(16 bytes)	encrypted_price(8 bytes)	integrity_signature(4 bytes)
---------------------------------	--------------------------	------------------------------

1 加密算法

输入：

iv	16 字节，UUID 的二级制串，转化为字符串后即为 BID
price	8 字节，待加密的价格
e_key	价格加密 key

i_key	完整性校验 key
-------	-----------

伪代码：

```
pad = hmac_sha1(e_key, iv) //take first 8 bytes
enc_price = pad <xor> price
signature = hmac_sha1(i_key, price || iv) //take first 4 bytes
message= encode_base64(iv || enc_price || signature)
```

2 解密算法

输入：

message	加密串
e_key	价格加密 key
i_key	完整性校验 key

伪代码：

```
str = decode_base64(message)
iv, enc_price, signature = split(str)
pad = hmac_sha1(e_key, iv) //take first 8 bytes
price = enc_price<xor> pad
conf_sig = hmac_sha1(i_key, price || iv) //take first 4 bytes
success = (conf_sig == signature)
```

3 加密串说明

价格宏替换之前会对价格串做 url encode：

```
message=加密算法(price);
message=url_encode(message);
宏替换(%%SETTLE_PRICE%% , message) ;
```

拿到价格参数后需要先做 url decode：

```
message=url_deocde(message);
message=replace(' ', '+'); // APP 端的价格解密前需要把空格替换为加号，PC/M 不需要
price=加密算法(message);
```

4 测试数据

```
ekey:testkey
ikey:testikey
price:10086
msg:MTIzNDEyMzQxMjM0MTIzNOX94Llcv7qY3d7+mQ==
```

2.3.2 设备 ID 加密

DSP 获取到的设备 ID 号，是经过编码后的密文 ID。需要配合密钥才能解码成功。

每个 DSP 有一个唯一的解码密钥；

为方便说明，约定如下变量与操作：

ID_real：真实的设备 ID 号，对于 IOS 设备，优先 idfa，idfa 获取不到时是 openudid，对于 android 设备，是 imei

ID_encrypt：编码后的设备 ID 号

K_real：分配给 DSP 的解码密文

K_pad：定长密钥

CRC：校验码

MSG：最终结果串

＋：字符串拼接

MD5()：md5 值计算

Len()：字符串长度

Sub()：子串

加密：

$K_pad = md5(K_real)$	—— 对 key 做 md5，结果是二进制 16 字节
$ID_encrypt = K_pad \wedge ID_real$	—— 循环异或，16 字节密钥循环使用
$CRC = Sub(md5(ID_real), 4)$	—— 对原始 deviceid 做 md5 取 4 字节
$MSG = Base64Enc(ID_encrypt + CRC)$	

解密：

$MSG = Base64Dec(MSG)$	
$CRC = Sub(MSG, -4)$	—— MSG 后 4 字节为 CRC 校验码
$ID_encrypt = Sub(MSG, len - 4)$	—— 除去 CRC 部分为 ID_encrypt
$K_pad = Sub(md5(K_real), 16)$	—— 对 key 做 md5，结果是二进制 16 字节
$ID_real = ID_encrypt \wedge K_pad$	—— 循环异或，16 字节密钥循环使用
$CRC_new = Sub(md5(ID_real), 4)$	—— 对解出的 deviceid 做 md5 取 4 字节
$CRC == CRC_new$	—— CRC 相等则解码正确

测试数据：

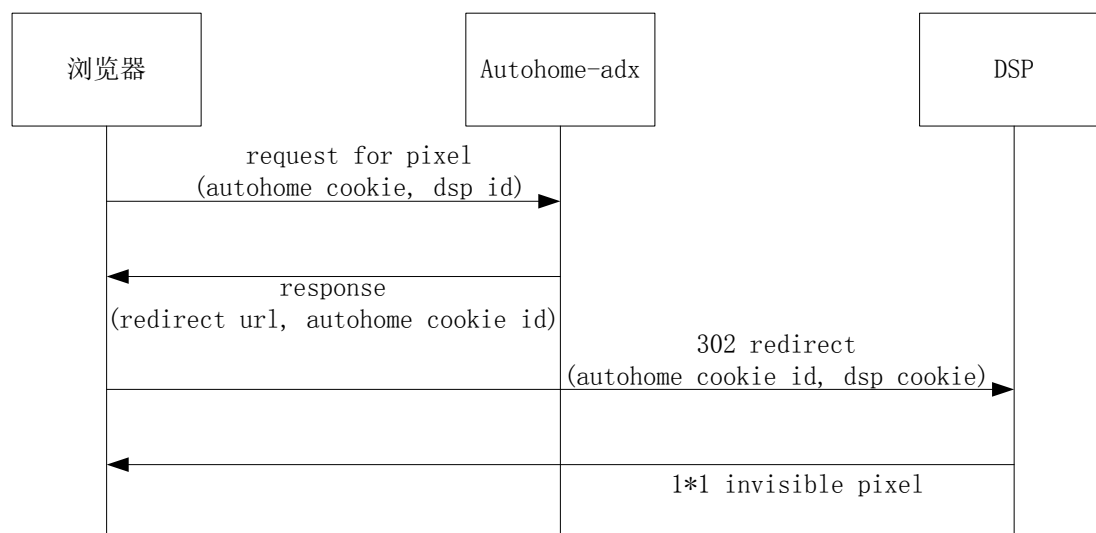
key:test_key

deviceid:test_device_id123

msg:+FeibG01uu5BmkzmXMo1K7+x6wE+

3 cookie mapping

3.1 概述



如果 DSP 需要 cookie mapping (新 cookie 产生或者 cookie 过期), 只需在竞价响应中将 is_cm 字段置为 true , 无论 DSP 是否能竞价成功 , adx 都会为 DSP 生成 1*1 pixel 并发起 cookie mapping。

只有 site 广告支持 cookie mapping。

当前暂不支持 DSP 主动发起的 cookie mapping ;

3.2 服务地址

<http://cm.adx.autohome.com.cn/cm>

3.3 参数说明

Parameter	Description
dspid	DSP ID , 用于 1*1 pixel 请求中
ahuid	加密的汽车之家 cookie ID , 由 adx 添加到重定向 URL 中

3.4 示例

当 dsp 响应中 is_cm 字段置为 true 时 , adx 会生成一个 cm 请求链接返回给页面 :

<http://cm.adx.autohome.com.cn/cm?dspid=1&ahuid=USERID-XXX>

页面发送该 cm 请求给 adx , adx 再次生成一个重定向链接 :

`http://dsp.redirecturl.com? ahuid=USERID-XXX`

页面将重定向请求发送到 dsp 并携带 ahuid 参数 ,dsp 收到该页面请求后对 ahuid 进行关联。

4 信息同步

4.1 报表查询接口

一期暂时由 ADX 团队推送，后续提供查询接口

4.2 素材审核接口

4.2.1 获取素材 API

请求 URL: `https://bizadx.api.autohome.com.cn/adx-api/api/creative/upload`

4.2.1.1 说明

DSP 可通过该 API 向 ADX 推送素材：

素材信息：所属 DSP、所属广告主&广告主 ID、所属行业&行业 ID；

素材元素信息：图片、广告位宽高、文字、点击链接、元素信息等；

素材类型：参考 [5.1 数据字典](#) 中的 creative_type；

模版 ID：为非必填属性，如需要按元素审核，则将此字段留空；如一套素材需按不同的模版审核，则使用 [2.1.5 广告形式说明](#) 中给出的值填充此字段

广告内容：具体定义参考 [2.1.5 广告形式说明](#)

使用 POST 调用。

此接口需要验签。

4.2.1.2 请求参数

```
{
  "dspId": 1,
  "dspName": "xyz",
  "creative": [
    {
      "advertiserId": 101,
      "advertiserName": "advertiser name",
      "industryId": 10,
      "industryName": "industry name",
      "creativeTypeId": 1004,
```

```

        "width": 640,
        "height": 480,
        "templateId": 114,
        "repeatedCode": "abcd",
        "adsnippet": {
            "img": "imgae.url",
            "link": "click.url",
            "text": "xxxxx",
            "subText": "xxxxx"
        }
    },
    "sign": "abcdabcdabcdabcdabcdabcdabcdabcd",
    "timestamp": 1475856123451
}

```

参数说明

名称	类型	必填	备注
dspId	int	是	所属 DSP ID
dspName	string	是	所属 DSP 名称
sign	string	是	验签串
timestamp	long	是	验签时间戳，1970 年 1 月 1 日至今毫秒值

list 元素说明

名称	类型	必填	备注
advertiserId	int	是	所属广告主 ID
advertiserName	string	是	所属广告主名称，需要填写全称
industryId	int	是	所属行业 ID
industryName	string	是	所属行业名称
creativeTypeId	int	是	素材类型 ID 1001-文字链，1002-图片，1003-图文
width	int	是	广告位宽度，如果不知道宽度则填充 0
height	int	是	广告位高度，如果不知道高度则填充 0
templateId	int	否	模板 ID (参考 2.1.5 广告形式说明)
repeatedCode	string	是	去重码，预留字段，填充固定值空字符串""
adsnippet	object	是	广告内容

adsnippet 字段 (具体定义参考 [2.1.5 广告形式说明](#))

名称	类型	必填	备注
img	string		图片地址
link	string		链接地址
text	string		标题地址
subText	string		子标题地址

4.2.1.3 应答参数

```

{
  "data": {
    "creativeIds": [1]
  }
}

```

```
    },
    "statusInfo":{
        "global":"error message"
    },
    "status":1
}
```

字段说明

名称	类型	必填	备注
data	object		具体传输数据
statusInfo	object		错误数据
status	int	是	为 1 表示成功

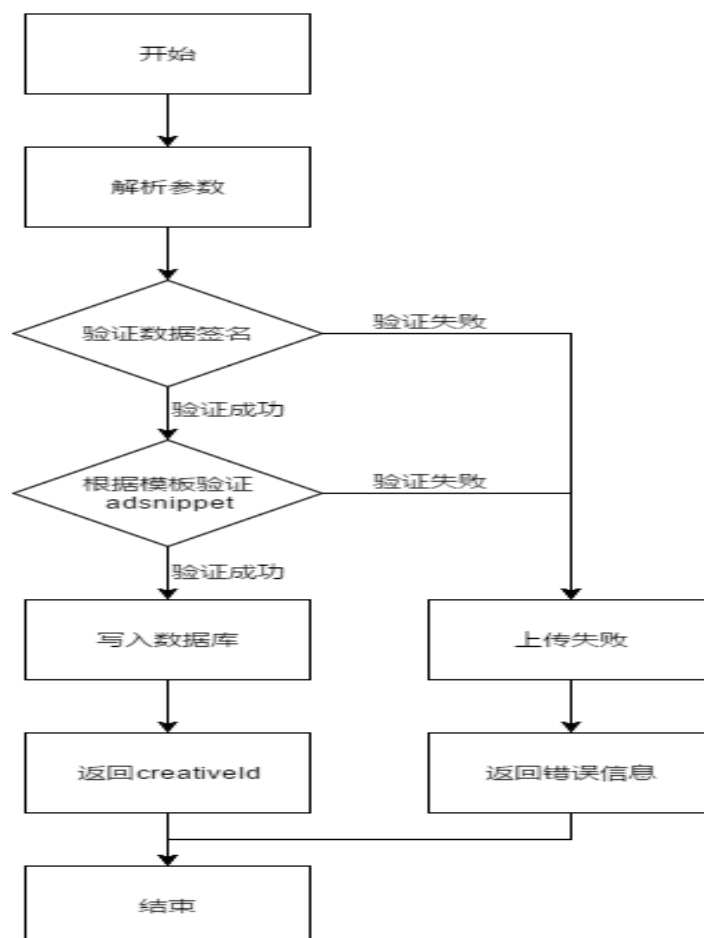
data 说明

名称	类型	必填	备注
creativeIds	int array	是	生成的素材 ID，按传入顺序

statusInfo 说明

名称	类型	必填	备注
globals	string	是	错误信息

4.2.1.4 流程图



4.2.2 获取素材审核信息 API

请求 URL: <https://bizadx.api.autohome.com.cn/adx-api/api/creative/getAuditInfo>

4.2.2.1 说明

DSP 可通过该 API 获得素材 ID&审核结果（通过/拒绝）&拒绝原因

拒绝原因 list：

素材与链接内容不符

请保持素材美观/清晰/完整

素材描述撰写不符

素材内容中不得涉及无意义符号

素材内容中不能使用违禁词

素材及落地页中涉及敏感/低俗内容

素材涉及虚假宣传

落地页无法打开/落地页显示异常

落地页主域与账户信息主域不一致

素材中涉及诱导点击

素材点击不得含有下载内容

创意内容与类目不匹配

素材禁止使用二维码

请提供相关资质或授权文件

有奖销售奖品价值不得超过 5000 元

素材不得涉及竞品

禁止投放的行业

其他：手动输入信息

使用 GET 方式调用。

此接口需要验签。

4.2.2.2 请求参数

dspId=1&creativeId=1,2,3&sign=abcdabcdabcdabcdabcdabcdabcdabcd&time
stamp=1475856123451

参数说明

名称	类型	必填	备注
dspId	int	是	所属 DSP ID
creativeId	int array	是	素材 ID，用逗号分隔的整数
sign	string	是	验签串
timestamp	long	是	验签时间戳，1970 年 1 月 1 日至今毫秒值

4.2.2.3 应答参数

```
{
  "data":{
    "status":[
      {
        "id":1,
        "auditStatus":2,
        "auditComment":"abc",
        "auditTime":1475856000001
      },
      {
        "id":2,
        "auditStatus":1,
        "auditTime":1475856000001
      },
      {
        "id":2,
        "auditStatus":0
      }
    ],
    "statusInfo":{
      "global":"error message"
    },
    "status":1
  }
}
```

字段说明

名称	类型	必填	备注
data	object		具体传输数据
status	int	是	为 1 表示成功
statusInfo	object		错误信息

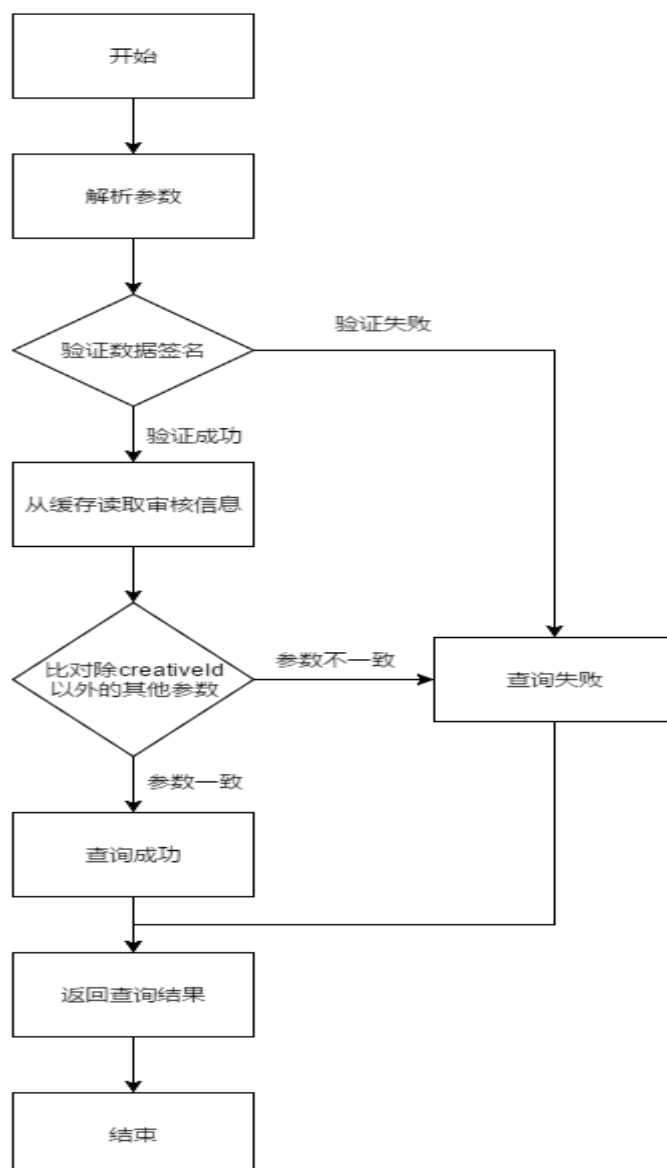
data.status 元素说明

名称	类型	必填	备注
id	int	是	素材 ID
auditStatus	int	是	状态：1-审核通过，2-拒绝，0-未审核
auditComment	string		若未通过，表示审核意见
auditTime	long		审核时间，未审核没有此字段

statusInfo 字段

名称	类型	必填	备注
globals	string	是	错误信息

4.2.2.4 流程图



4.2.3 验签算法

4.2.3.1 GET 和 POST 请求验签算法

1. 获取参数中的时间戳字段；

2. 验证时间戳与本地时间误差不大于 30s；
3. 将除 sign 字段的所有其他参数按照参数名字典序从小到大排序；
4. 将除 sign 字段的排序后的参数和值用=和&连接获得 parameterStr；
5. 根据 DSP ID 查询 DSP signKey；
6. 计算 parameterStr+signKey 的 MD5 值；
7. 与 sign 字段进行比较，相同则通过验签。

4.2.3.2 POST json 请求验签算法

1. 获取参数中的时间戳字段；
2. 验证时间戳与本地时间误差不大于 30s，服务端定时向 NTP 服务器获取授时；
3. 将 sign 字段从请求 json object 中删除；
4. 将 json object 序列化为 jsonString，字段按照字典序从小到大排序，null 值不显示，无缩进，key 带双引号；
5. 根据 DSP ID 查询 DSP signKey；
6. 计算 jsonString+signKey 的 MD5 值；
7. 与 sign 字段进行比较，相同则通过验签。

4.2.3.3 示例代码

```
// 普通 GET 和 POST
private String getSign(Map<String, String> params, String signKey) {
    Set<String> keySet = params.keySet();
    List<String> keyList = new ArrayList<>();
    keyList.addAll(keySet);
    Collections.sort(keyList);
    List<String> paramList = new ArrayList<>();
    for (String key: keyList) {
        paramList.add(key+"="+params.get(key));
    }
    paramList.add("timestamp="+System.currentTimeMillis());
    String str = StringUtils.join(paramList, "&");
    String sign = MD5Util.encode(str+signKey);
    return sign;
}

// POST json
private String getSign(JSONObject obj, String signKey) {
    obj.put("timestamp", System.currentTimeMillis());
    String str = obj.toJSONString();
    String sign = MD5Util.encode(str+signKey);
    return sign;
}
```

5 附录

5.1 数据字典

1. view_type 广告位类型

值	描述
101	PC-按钮
102	PC-通栏
103	PC-图文
104	PC-文字链
201	M-信息流
202	M-图文
203	M-通栏
204	M-文字链
301	APP-开屏
302	APP-信息流
303	APP-通栏
304	APP-图文

2. creative_type 素材类型

值	描述
1001	文字链
1002	图片
1003	图文

3. network 网络环境

值	描述
0	未识别
1	wifi
2	2g
3	3g
4	4g

4. operator 网络运营商

值	描述
0	未识别
7012	移动
70123	联通
70121	电信

5. category

详细见行业类目字典