

# Letv Ad Exchange 对接文档

## 版本记录

版本号	完成时间	备注
1.0.0	2014-12-22	1. 对接文档初始版本。
1.0.1	2015-01-04	1. 修改 Win-notice 宏，大括号改成小括号。
1.0.2	2015-01-06	1. 删除了业务接口中 103,108,109 三个和尺寸相关的错误信息。 2. 广告上传接口中增加了广告尺寸这个必填项。
1.0.3	2015-01-09	1. 业务接口 107 错误注释添加，增加 115,116,404，请求参数 dspid 改为数字。 2. win-notice 接口中增加了对成交价格的加密。
1.0.4	2015-01-14	1. 增加了对 DSP 不参与竞价时返回格式的说明。
1.0.5	2015-01-15	1. 修改广告类型 type 为必填项
1.0.6	2015-01-16	1. 增加 AES 算法的具体细节描述
1.0.7	2015-01-20	1. AES 加密后的密文要进行 url encode 才能放到 url 中 2. 竞价请求中的 Content-Type 为 application/json 3. 增加了对竞价和监测发生流程的说明
1.0.8	2015-01-27	1. 删除 bid-response 中的 mid（素材 ID）字段
1.0.9	2015-02-04	1. 加入和媒体方接口
1.0.10	2015-02-05	1. 增加了对广告素材规格的说明
1.0.11	2015-02-09	1. 增加了对广告素材时长的限制 2. 第一次传的广告若已审为通过状态，后面上传的无效，以第一个为主；其他状态，后面上传的会覆盖之前的
1.0.12	2015-03-10	1. 增加了对广告上传时的 monitor 参数格式注意说明。
1.0.13	2015-03-12	1. 广告时长支持小数类型，如 15.32 2. 广告时长、宽、高兼容字符串数值类型，分别举例：“15.32”，“480”，“70”。
1.0.14	2015-03-18	1. 修改了对广告素材规格的说明
1.0.15	2015-03-20	1. 增加售卖方式字段(bid-request.imp.bidtype)，标明本次曝光的售卖方式是 PDB 还是 RTB 2. 标出移动端支持的字段
1.0.16	2015-03-25	1. 增加了 bid-request.imp.pmp 对象，删掉了 bid-request.imp.bidtype 字段。
1.0.17	2015-04-02	1. 在业务数据接口中增加了测试接口 2. 在 device 字段中增加了 did 和 dpid 两个字段 3. 增加了加解密的示例 4. 增加了对联调过程的说明 5. 增加了移动端视频素材上传转码的描述。
1.0.18	2015-04-28	1. 在曝光监测中增加了宏替换功能

1.0.19	2015-04-30	1. 在批量获取指定广告的审核状态接口，修改错误码为 302 的返回示例
1.0.20	2015-05-21	1. 两个获取广告状态接口增加了测试线上地址的说明
1.0.21	2015-06-04	1. 增加了 bid-response 中对点击跳转地址的限制描述。 2. 修改了前贴片素材时长和尺寸限制。
1.0.22	2015-07-16	1. 增加获取 deal 及 deal 对应的广告主接口。 2. 增加获取所有广告主 id 及 name 的接口。 3. 广告上传接口的“广告主”字段必须为广告主真实名称，如广告主的公司详称，否则广告不能通过审核。 4. 广告上传接口增加平台 platform、媒体 media 字段，一段时间内选填，DSP 方作好升级后这两字段将改为必填。 5. 广告上传接口新增 117、118、119、120 响应 code。 6. 更新了乐视 ADX 平台广告素材的标准。
1.0.23	2015-08-11	1. Bid-response 添加 admpara 字段，该字段用于存储动态物料的拼接参数
1.0.24	2015-08-21	1. 广告上传接口的 platform、media 这两参数同时支持数组和数字两种形式。 2. media 参数的取值范围由 0、1 改为 1、2。
1.0.25	2015-12-02	1. Bid-request 的 app, site 对象添加 cr 版权信息节点 2. Bid-request 的 sited 对象的 page 节点为媒体域名 3. 上传广告素材时不再需要 width, height, platform, monitor 四个字段 4. 上传广告素材时可以上传多个 landingpage 5. 查询素材状态接口中增加了“转码中”状态码。 6. 视频素材规范变更
1.0.26	2016-01-22	1. PC 端暂停广告位尺寸变更
1.0.27	2016-02-17	1. 上传广告信息增加广告位类型（display）、行业（industry） 广告上传接口新增 121、122、123 响应 code 2. 获取广告的审核状态接口及获取未通过的广告信息接口增加 adid 3. 竞价请求（bid-request）的 display 字段从 video 里移除，挪到上一级
1.0.28	2016-03-14	1. 上传广告信息中 display 字段改为数组类型 2. 竞价请求（bid-request）的 imp 对象增加 excluded_ad_industry 字段表示当前广告曝光机会不能展示对应行业的广告（注意：动漫、亲子、教育频道会屏蔽相关行业（4、21、22、23、24、25）以及没有行业字段的素材）

## 概述

本文档描述了接入 Letv Ad Exchange 的各接口的详细含义，包括离线数据同步接口和实时竞价接口。

其中离线数据同步接口中的广告位同步接口传输的数据和实时竞价接口中的数据是一致的（广告位底价只通过实时竞价接口给出），DSP 可以自行选择使用离线接口或者实时竞价接口获取广告位信息。

## 对 DSP 接口

## 业务数据接口

### 概述

Letv exchange 和 DSP 间接口使用 HTTP POST 方式，数据以 json 格式组织。下表中所列出的 dspid 和 token 字段在每次请求中是必填字段，其他字段在下述具体接口中分别说明。所有数据都使用 UTF-8 编码。

### 请求参数

参数名称	是否必须	含义
dspid	是	DSP 在 ADX 系统中的 ID，用来标识是哪个 DSP。
token	是	DSP 对应的 token 值，由 ADX 生成，用来进行权限验证。

### 返回参数

参数名称	是否必须	含义
result	是	0: 执行成功 1: 系统认证失败 2: 请求参数校验错误
message	是	如果 API 执行成功，返回结果数据或成功提示信息；如果执行失败，返回具体的错误提示信息
code	否	如果 API 执行失败，来标识具体的错误代码。请参考下表中的错误代码说明

### 错误代码

- 101~120 同步广告时的错误类型，即接口后缀：/ad/sync
- 201 获取审核未通过的广告信息时的错误类型，即接口后缀：/ad/denylist
- 301~302 获取指定的广告审核结果时的错误类型，即接口后缀：/ad/getstatus
- 401~404 DSP 报表查询时的错误类型，即接口后缀：/report/detail

ERROR_CODE	错误注释	处理说明
------------	------	------

101	广告加载失败	广告加载经常是因为根据广告 url 地址获取信息时超时, 确保 url 正常的情况下重试
102	必填项, 不支持的文件格式, 目前支持的文件格式: jpg,png,swf,flv,mp4,vpaid	
104	执行插入过程异常	该错误经常是因为将广告记录到 Adx 超时, 确保 url 正常的情况下重试
105	广告所属的广告主不能为空	添加广告的 advertiser 属性
106	广告生效时间为空或者不能解析	注: 时间格式为 YYYY-mm-dd
107	广告失效时间为空或者不是有效格式或者广告失效时间小于当前时间	注: 时间格式为 YYYY-mm-dd
110	参数中存在广告地址为空	
111	第三方监测地址错误	核对监测地址后重试
112	同步广告物料的 ad 参数必须要有	添加 ad 参数列表
113	广告物料的跳转地址 landingpage 参数不能为空	添加可用的 landingpage 参数
114	广告时长不能解析	将时长值修改为整数值
115	广告的宽度必填且为整数类型值	将宽度修改为整数值
116	广告的高度必填且为整数类型值	将高度修改为整数值
117	此广告 url 与 adx 库的已审核通过广告重复	更换广告 url 后重新上传
118	广告对应的平台为整数类型值	将平台修改为整数值
119	adx 平台库没有此广告对应的平台值	根据所给平台表更改正确平台值
120	广告的媒体属性不正确	更改正确媒体值
121	广告主非白名单	填写规范完整的广告主名称
122	广告的行业属性不正确	更改正确行业值
123	广告的广告位类型 display 不正确	更改正确广告位类型 display 值
201	上传时间不能解析	注: 时间格式为 YYYY-mm-dd
301	adurl 不能为空	添加 adurl 属性
302	该广告 url 并未同步到 Adx 中	确保 url 正常的情况下调用同步操作同步广告到 Adx
401	报表查询的开始时间为空或者不能解析	注: 时间格式为 YYYY-mm-dd
402	报表查询的结束时间为空或者不能解析	注: 时间格式为 YYYY-mm-dd
403	报表查询时间跨度超过 7 天	日期段跨度最大允许 7 天
404	报表查询时的结束时间早于开始时间	核对时间后重试

注: Letv exchange 的接口分测试接口和线上接口, 两套接口是独立的, 使用不同的 dspid 和 token, 测试接口 url 以 “http://ark.letv.com/apitest/” 为前缀, 线上接口 url 以 “http://ark.letv.com/api/” 为前缀, 请注意两者的区别。

## 广告位信息获取 API

DSP 通过该接口获得 Exchange 拥有的库存信息，包括广告位总量，每个广告位的 ID、名称、尺寸以及支持的物料格式，目前系统支持的物料格式请查看 Ad Exchange 广告素材规格说明章节。

线上接口	<a href="http://ark.letv.com/api/adzone/list">http://ark.letv.com/api/adzone/list</a>
测试接口	<a href="http://ark.letv.com/apitest/adzone/list">http://ark.letv.com/apitest/adzone/list</a>

请求示例：

POST: <a href="http://ark.letv.com/api/adzone/list">http://ark.letv.com/api/adzone/list</a>
JSON: {"dspid":11268,"token":"92205dff8f9d48e1b7a26b0b88af7dc1"}

正常返回示例：

<pre>{   "result":0,   "message":   {     "total":2,     "records": [       {         "adzoneid": "101",         "adzonename": "汽车-首页-banner",         "size": "300*250",         "duration": "15-60",         //广告位最小-最大时长，单位：秒(s)         "allowAdTypes": ["image/jpeg", "video/x-flv", "image/png"]         //允许播放的广告素材类型       },       {         "adzoneid": "102",         "adzonename": "新闻-视频",         "size": "400*300",         "duration": "15-75",         "allowAdTypes": ["video/x-flv"]       }     ]   } }</pre>
---

## 广告信息同步 API

广告信息同步 API 包括上传广告、获取审核结果、批量获取结果三个部分。

DSP 向 exchange 上传物料，由 exchange 审核，通过审核的物料才能在 exchange 平台上投放

出来。审核的结果通过获取审核结果的接口提供给 DSP。

## 上传广告信息 API

线上接口	<a href="http://ark.letv.com/api/ad/sync">http://ark.letv.com/api/ad/sync</a>
测试接口	<a href="http://ark.letv.com/apitest/ad/sync">http://ark.letv.com/apitest/ad/sync</a>

业务参数：本次同步需要新增的广告信息，是 JSON 格式的数据，具体参考请求示例数据。

注：广告时长必填，且图片类型的时长值为 0。

第一次传的广告若已审为通过状态，重复上传后会覆盖之前的。

若广告无对应的监测 monitor 信息，则参数格式为"monitor":[]。

动漫、亲子、教育频道会屏蔽相关行业（4、21、22、23、24、25）以及没有行业字段的素材

请求示例：

POST: <a href="http://ark.letv.com/api/ad/sync">http://ark.letv.com/api/ad/sync</a>
<p>JSON:</p> <pre>{   "dspid":11268,   "token":"92205dff8f9d48e1b7a26b0b88af7dc1",   "ad":   [{     "url": "http://somecdn.com/media/v1/00.jpg",     //广告的地址，必填。     "landingpage": ["http://www.some.com", "http://www.some2.com"],     //点击跳转落地页的地址，必填，实际使用的点击跳转地址以竞价响应中返回的为准（若竞价响应中没有给出点击跳转地址，则广告展现后不能跳转）。此处填写点击跳转地址主要用于审核。可以有多个跳转地址。     "advertiser": "广告主的中文名称",     //必填，广告主名称必须规范完整，否则不能通过人工审核。     "startdate":"2014-11-16",     //物料生效时间，格式要求：YYYY-mm-dd，必填。     "enddate":"2014-12-30",     //物料失效时间，格式要求：YYYY-mm-dd，必填。     "type":"jpg",     //物料类型：必填，取值范围参看素材类型表中第一列，必填。     "duration": 0,     //时长，图片素材时长值必须为 0     //支持浮点数，可以用数字类型，也可以用字符串类型，必填。     "media": [1,2,3],     //media 字段的取值范围见下表，数字和数组类型都支持，选填。     //也可以是： "media": 1,     "industry": 11,     //行业字段的取值范围见下方说明，必填。     "display": [2,4,5]</pre>

```
//广告位类型，display字段的取值范围见下方说明，数组类型，选填。  
}  
]  
}
```

返回示例：

```
{  
  "result": 2,  
  "message": {  
    "102": ["http://somecdn.com/media/v1/00.png"],  
    "111": ["http://somecdn.com/media/v1/01.jpg"],  
    "114": [  
      "http://somecdn.com/media/v1/03.jpg",  
      "http://somecdn.com/media/v1/04.png"]  
    }  
  }  
}
```

如果返回信息是：{"result":0,"message":{}}，说明物料上传成功 即物料成功上传到我们的 exchange 系统,物料上传之后的状态是待审核。

其他情况当 result 为 2，message 是一个 object，前面的数值是错误的 code 类型，后面的数组是该错误类型下的物料 URL 列表

### Media 字段取值范围

媒体类型 ID	媒体名称
1	乐视网
2	华数 TV 网
3	乐视云视频

### 行业 industry 字段取值范围

行业 ID	行业名称
1	其他
2	食品饮料类
3	日化类
4	美容及化妆品类
5	服饰类
6	汽车及其附属品类
7	母婴类
8	网络服务类
9	家居装饰类
10	家用电器类
11	手机数码与消费电子
12	通讯服务类
13	金融保险类

14	教育类
15	旅游类
16	珠宝奢侈品类
17	休闲娱乐类
18	房地产类
19	生活服务类
20	运动健身类
21	保健食品类
22	医疗药品类
23	游戏类
24	烟酒类
25	成人用品类

注意：动漫、亲子、教育频道会屏蔽相关行业（4、21、22、23、24、25）以及没有行业字段的素材

### 广告位类型 display 字段取值范围

广告位类型 ID	广告位类型名称
0	页面
1	拉幕
2	前贴
3	标版
4	中贴
5	后贴
6	暂停
7	浮层

### 获取审核未通过的广告信息 API

线上接口	<a href="http://ark.letv.com/api/ad/denylist">http://ark.letv.com/api/ad/denylist</a>
测试接口	<a href="http://ark.letv.com/apitest/ad/denylist">http://ark.letv.com/apitest/ad/denylist</a>

业务参数：

字段名称	类型	描述
uploaddate	string	上传的时间：将返回该时间之后的所有未审核通过的广告列表 可以为空，为空时返回所有的审核未通过的广告列表 格式要求：YYYY-MM-DD

请求示例：

POST: <a href="http://ark.letv.com/api/ad/denylist">http://ark.letv.com/api/ad/denylist</a>
JSON: {



```
"dspid":11268,
"token":"92205dff8f9d48e1b7a26b0b88af7dc1",
"uploaddate":"2014-12-04"
}
```

返回示例:

正常返回

```
{
  "result":0,
  "message":
  {
    "total":2,
    "records": [
      {
        "adid": 11223,  //adx ad id
        "url": "http://material.client.com/123.swf", //广告 url
        "reason": "显示异常"  //审核不通过原因
      },
      {
        "adid": 11226,  //adx ad id
        "url": "http://material.client.com/134.swf",
        "reason": "跳转地址异常"
      }
    ]
  }
}
```

校验出请求参数存在错误时的返回示例

```
{
  "result":2,
  "message": "上传时间不能解析"
}
```

## 批量获取指定广告的审核状态

线上接口	<a href="http://ark.letv.com/api/ad/getstatus">http://ark.letv.com/api/ad/getstatus</a>
测试接口	<a href="http://ark.letv.com/apitest/ad/getstatus">http://ark.letv.com/apitest/ad/getstatus</a>

业务参数:

字段名称	类型	描述
adurl	array of string	需要查询的一个或多个广告的 URL

请求示例:

POST: <http://ark.letv.com/api/ad/getstatus>

JSON:

```
{
```

```
"dspid":11268,
"token":"92205dff8f9d48e1b7a26b0b88af7dc1",
"adurl":["http://material.client.com/134.swf","http://material.client.com/135.s
wf","http://material.client.com/136.swf"]
}
```

返回示例

正常返回

```
{
  "result":0,
  "message":
  {
    "total" :3,
    "records" : [
      {
        "adid": "134",
        "url": "http://material.client.com/134.swf",
        "result":"未通过",
        "reason": "默认素材不需要上传了"
      },
      {
        "adid": "135",
        "url": "http://material.client.com/135.swf",
        "result":"通过",
        "reason": ""
      },
      {
        "adid": "136",
        "url": "http://material.client.com/136.swf",
        "result":"待审核",
        "reason": ""
      }
    ]
  }
}
```

业务参数校验出错误的示例

```
{
  "result":2,
  "message":{"301":"adurl 不能为空"}
}
```

或者

多条广告 url，既有未同步到 adx 中，又有状态结果返回。若全部指定的 adurl 均未同步到 adx 中，则下方示例为“total”:0, “records”:[]

```
{
  "result":2,
  "message":
```

```
{
  "302": ["http://material.client.com/135.swf", "http://material.client.com/136.swf"],
  "total": 1,
  "records": [
    {
      "adid": "134",
      "url": "http://material.client.com/134.swf",
      "result": "未通过",
      "reason": "默认素材不需要上传了"
    }
  ]
}
```

## Deal 信息获取 API

DSP 通过该接口获得 Exchange 创建使用的 deal 及其对应的 advertiser 广告主信息。

线上接口	<a href="http://ark.letv.com/api/deal/list">http://ark.letv.com/api/deal/list</a>
测试接口	<a href="http://ark.letv.com/apitest/deal/list">http://ark.letv.com/apitest/deal/list</a>

请求示例：

POST: <a href="http://ark.letv.com/api/deal/list">http://ark.letv.com/api/deal/list</a>
JSON: {"dspid":11268,"token":"92205dff8f9d48e1b7a26b0b88af7dc1"}

正常返回示例：

```
{
  "result":0,
  "message":
  {
    "total":2,
    "records": [
      {
        "dealId": 5,
        "advertiserIds": [2,7,9]
      },
      {
        "dealId": 8,
        "advertiserIds": [5,11,32]
      }
    ]
  }
}
```

## 广告主信息获取 API

DSP 通过该接口获得 Exchange 所有 advertiser 广告主信息。

线上接口	http://ark.letv.com/api/advertiser/list
测试接口	http://ark.letv.com/apitest/advertiser/list

请求示例：

POST: http://ark.letv.com/api/advertiser/list
JSON: {"dspid":11268,"token":"92205dff8f9d48e1b7a26b0b88af7dc1"}

正常返回示例：

<pre>{   "result":0,   "message":   {     "total":2,     "records": [       {         "id": 5,         "name": "联合利华（中国）有限公司"       },       {         "id": 11,         "name": "上海天天鲜果电子商务有限公司"       }     ]   } }</pre>
---

## 报表获取 API

DSP 接入 ADX 平台后，可以通过 API 获取相关的报表数据，传入参数是一个时间段：开始时间和结束时间，返回的信息是[每天、每个广告位] 上的参与竞价数、成功竞价数、曝光数、点击数、计费的费用。

特别说明：为了保证数据返回的速度，一次获取的时间间隔建议不要超过一周。对于时间跨度超过一周的请求，我们将直接返回异常。

线上接口	http://ark.letv.com/api/report/detail
测试接口	http://ark.letv.com/apitest/report/detail

业务参数：

字段名称	类型	描述
startdate	string	报表查询的开始日期，要求是 YYYY-MM-dd 的格式，如

		2012-02-01，不能为空
enddate	string	报表查询的结束日期，要求是 YYYY-MM-dd 的格式，如 2012-02-07，不能为空

请求示例：

POST: <http://ark.letv.com/api/report/detail>

JSON:

```
{
  "dspid":11268,
  "token":"92205dff8f9d48e1b7a26b0b88af7dc1",
  "startdate":"2014-12-05",
  "enddate":"2014-12-11 13:14:15"
}
```

返回示例：

```
{
  "result":2,
  "message":
  {
    "402": "报表查询的结束时间为空或者不能解析"
  }
}
```

或者

```
{
  "result":2,
  "message":
  {
    "403": "报表查询时间跨度超过 7 天"
  }
}
```

或者

```
{
  "result":0,
  "message":
  {
    "total":2,
    "records":
    {
      "2014-12-06": [
        {
          "adzoneid": 1,
          "bid": 12,
          "winbid": 8,
          "pv": 8,
          "click": 2,
          "expend": 1000
        }
      ]
    }
  }
}
```

```
    }
  ]
  "2014-12-07": [
    {
      "adzoneid": 1,
      "bid": 162,
      "winbid": 61,
      "pv": 61,
      "click": 10,
      "expend": 3000
    },
    {
      "adzoneid": 2,
      "bid": 5445,
      "winbid": 554,
      "pv": 365,
      "click": 14,
      "expend": 9000
    }
  ]
}
}
```

## 实时竞价接口

### 概述

实时竞价发生时, exchange server 和 DSP server 间使用 HTTP POST 方式通信, 数据格式为 json, Http 请求头中 Content-Type 为 application/json, 启用 keep-alive, 单次请求超时时间设置为 120ms。

### 竞价请求 (bid-request)

字段名称	类型	描述
id	string	用来表示本次竞价请求的唯一 ID, 对应 DSP 响应中的 id 字段, 由 exchange 生成。
imp	array of objects	曝光对象数组
site		网站对象。
device	object	设备对象

user	object	用户对象
app	object	应用对象，包含移动应用的信息。

## imp 对象

字段名称	类型	子字段	子字段类型	描述
id	string			针对每一个曝光的唯一 ID, 对应 DSP 响应中 bid.impid 字段，由 exchange 生成。
adzoneid	int			乐视 ADX 中的广告位 id
pmp	object			PMP 对象，当本节点存在时，使用 PDB 售卖方式，否则使用 RTB 方式。
bidfloor	double			广告位底价，单位是（分/CPM）
display	int			广告位类型，参看广告位类型表。
banner	object	banner 类型广告位，该字段和 video 字段有且只有一个在乐视 ADX 中，视频播放器中的暂停位是 banner 类型。		
		w	int	广告位宽度
		h	int	广告位高度 注：banner 广告素材的宽和高必须和广告位的宽和高完全匹配，否则无法竞价成功。
video	object	video 类型广告位		
		mime	array of string	支持播放的视频格式，目前有 video/x-flv video/mp4
		minduration	int	视频广告最小时长
		maxduration	int	视频广告最大时长 目前最小时长会和最大时长的值一样，可以忽略最小时长。 广告素材的时长只要不超过最大时长+0.5s 就能正常展示。
		w	int	广告位宽度
		h	int	广告位高度 此处的宽高只是一个建议值。对视频广告位来说，因为可以随客户端动态缩放，所以并没有明确的宽高属性，广告素材只要符合下文中 <a href="#">广告素材规格说明</a> 就可以正常展示。
ext	object			
		excluded_ad_industry	array of int	表示该广告位不能展示列表中出现的行业的广告。行业 ID 参考 <a href="#">这里</a>

## site 对象

字段名称	类型	子字段	子字段类型	描述
name	string			媒体网站名称
page	string			当前页面 url
ref	string			Referer url
cr	Int			版权信息,默认为 1
content	object	视频的内容相关信息，只有视频贴片类广告才有这个字段 目前整个 content 字段为空		
		title	string	视频标题
		keywords	string	视频标签关键字，多个关键字用英文逗号分隔
		ext	object	扩展字段，具体描述见下方

## site.content.ext 对象

字段名称	类型	描述
channel	string	视频频道 ID
cs	string	二级频道 ID

## user 对象

user 对象目前只有用户的 id 信息，其他字段为空

字段名称	类型	子字段	子字段类型	描述
id	string			exchange 用户 ID
gender	string			性别，M 为男性，F 为女性，空表示未知。
yob	int			出生年份，4 位数字。
ext	object			
		models		用数字形式标记的 DMP 信息标签

## app 对象

app 对象中，标明“支持”的字段会有有效的值，标明“不支持”的字段一律为空字符串（string 类型）或者 0 值（int 类型）。

字段名称	类型	子字段	子字段类型	描述	支持情况
------	----	-----	-------	----	------



name	string			App 的名称	支持， 提供 APP 名称和版本
cr	Int			版权信息,默认为 1	
ext	object	扩展信息			不支持
		sdk	string	投放 SDK 的版本，例如“91_v1”	不支持
		market	int	应用商店编号	不支持
		appid	string	应用在应用商店内的 ID。	不支持
		cat	string	应用在应用商店内的分类 ID。	不支持
		tag	string	app 在上述应用商店内的标签 (英文或中文 UTF8-urlencode 编码) 多个标签使用半角逗号分隔。	不支持
content	object			视频的内容相关信息。只有视频贴片类型的广告位才会有这个字段，同 site.content 对象	不支持

## device 对象

device 对象中，标明“支持”的字段会有有效的值，标明“不支持”的字段一律为空字符串（string 类型）或者 0 值（int 类型）。

字段名称	类型	子字段	子字段类型	描述	支持情况
ua	string			user agent	PC 端、移动端支持
ip	string			ip	PC 端、移动端支持
geo	object	设备的当前地理位置信息			
		lat	float	纬度（-90~90）	不支持
		lon	float	经度（-180~180）	不支持
		ext	object	扩展属性，参见 Geo.ext 详细字段解释。	
didmd5	string				不支持
dpidmd5	string			不提供经过 md5 哈希的 IMEI 和 UUID 等值，DSP 方收到原始数据后可以自己哈希。	不支持
did	string			原始的 IMEI，没有经过 MD5 哈希	移动端支持
dpid	string			原始的平台相关 ID，Android ID 或 iOS 的 UDID，或者 windows phone 的 id，没有经过 MD5 哈希	移动端支持
make	string			设备生产商，如“Apple”	移动端支持

model	string			设备型号，如“iPhone”	移动端支持
os	string			操作系统，如“ANDROID”， 目前包括以下四种情况： ANDROID IOS WP OTHERS	移动端支持
osv	string			操作系统版本号，如“4.1”	移动端支持
carrier	string			运营商的 ID，参见 OpenRTB 协议及 <a href="http://en.wikipedia.org/wiki/Mobile_Network_Code">http://en.wikipedia.org/wiki/Mobile_Network_Code</a> 。	不支持
language	string			目前使用的国家—语言，如“zh_CN”	不支持
js	int			是否启用 Javascript， 1:启用（默认值） 0:未启用	不支持
connectiontype	int			网络连接类型，和 OpenRTB 一致： 0—未知，1—Ethernet，2—wifi，3— 蜂窝网络，未知代，4—蜂窝网络， 2G，5—蜂窝网络，3G，6—蜂窝网 络，4G。	PC 端、移动 端支持
devicetype	int			设备类型，和 0—手机，1—平板， 2—PC，3—互联网电视。	PC 端、移动 端支持
ext	object	扩展信息			
		idfa	string	iOS 的 IDFA 字段， 如：“1E2DFA89-496A-47FD-9941-D F1FC4E6484A”	移动端支持
		mac	string	MAC 地址，如“F0B4799A8CC9”	移动端支持
		macmd5	string	MD5 哈希的 MAC 地址	不支持
		ssid	string	WIFI 的名称	不支持
		w	int	设备的屏幕宽度，以像素为单位	移动端支持
		h	int	设备的屏幕高度，以像素为单位	移动端支持
		brk	int	设备是否越狱，1—已启用（默认）， 0—未启用。	不支持
		ts	int	发送请求时的本地 UNIX 时间戳(秒 数，10 进制)	PC 端、移动 端支持
		interstitial	int	是否使用全屏/互动方式来展现广 告。1—是，0—否（默认值）。	不支持

## geo.ext 对象

该对象目前未使用，填充为空

字段名称	类型	描述	支持情况
------	----	----	------

accuracy	int	GPS 的精确度，单位为米。如：100 表示精确度为 100 米。	不支持
----------	-----	-----------------------------------	-----

## pmp 对象

字段名称	类型	描述
private_auction	int	<p>1 表示只接受 PDB 竞价（保价保量，PDB）。这个过程中，不会竞价，也不会失去展示机会。</p> <p>0 表示这部分流量会和其他 DSP 竞价（邀请竞价，PMP）。这部分竞价规则是价高者得，以出价结算，不用第二竞价法。和 RTB 部分的区别在于这部分是在小范围的 DSP 内部竞争，会优先于 RTB 成交，且流量较为优质。</p> <p>以上两种流量在竞价时都必须带上商务上约定的价格，也就是下面表格 deal 对象中的底价。当且仅当 DSP 出价且出价大于等于低价时，ADX 会按照出价进行结算（出价比约定价格高也会按照出价结算）。</p>
deals	array of objects	Deal 数组，具体看 deal 对象描述

## deal 对象

字段名称	类型	描述
id	int	Deal id
bidfloor	int	Deal 的价格，单位是（分/CPM）。当 bid-request 中存在 pmp 对象时，以此处的底价为准，请忽略 imp.bidfloor。
wseat	array of string	可以参与的广告主 ID 列表，如果列表为空或者本字段不存在，则表示允许任意广告主参与。

## 示例

<b>PC端请求示例：</b>
<pre>{   "id": "a3a22934e2cd4afab2140ed8ec4637cf",   "imp": [     {       "id": "75991c55f1c34b06be6b487b0129b792",</pre>

```
        "bidfloor": 500,
        "adzoneid": 22,
        "display": 2,
        "video": {
            "w": 640,
            "h": 480,
            "minduration": 15,
            "maxduration": 15,
            "mime": [
                "application/x-shockwave-flash",
                "video/x-flv",
                "video/mp4"
            ]
        }
        "ext": {
            "excluded_ad_industry": [1,2,3]
        }
    },
    ],
    "site": {
        "name": "",
        "page": "",
        "ref": "",
        "cr": 1,
        "content": {
            "title": "",
            "keywords": "",
            "ext": {
                "channel": "",
                "cs": ""
            }
        }
    },
    },
    "device": {
        "ua": "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)",
        "ip": "113.56.215.251",
        "did": "",
        "dpid": "",
        "make": "",
        "model": "",
        "os": "",
        "osv": "",
        "carrier": "",
```

```
"language": "",
"js": 0,
"connectiontype": 1,
"devicetype": 2,
"geo": {
  "lat": 0,
  "lon": 0,
  "ext": {
    "accuracy": 0
  }
},
"ext": {
  "idfa": "",
  "mac": "",
  "ssid": "",
  "w": 0,
  "h": 0,
  "brk": 0,
  "ts": 1432515600,
  "interstitial": 0
},
"user": {
  "id": "b2612d5d268f4db69601c6bfb8890630",
  "gender": "",
  "yob": 0,
  "ext": {
    "models": []
  }
},
}
```

移动端请求示例:

```
{

  "id": "073724f1eb6d4c16bd12c8398b7ebdd2",
  "imp": [
    {
      "id": "02d527bd500a41ab909625e44c4feb7a",
      "bidfloor": 500,
      "adzoneid": 13,
      "display": 2,
      "video": {
        "w": 640,
```

```
        "h": 480,
        "minduration": 15,
        "maxduration": 15,
        "mime": [
            "video/mp4"
        ]
    }
    "ext": {
        "excluded_ad_industry": [1,2,3]
    }
}
],
"device": {
    "ua": "android/4.1.1 (samsung;GT-I9300) letvVideo/5810/aps_cm_3.0.5.6",
    "ip": "111.37.17.173",
    "did": "354316054366184",
    "dpid": "d5bae2f169a62ada",
    "make": "samsung",
    "model": "GT-I9300",
    "os": "ANDROID",
    "osv": "4.1.1",
    "carrier": "",
    "language": "",
    "js": 0,
    "connectiontype": 2,
    "devicetype": 0,
    "geo": {
        "lat": 0,
        "lon": 0,
        "ext": {
            "accuracy": 0
        }
    },
    "ext": {
        "idfa": "",
        "mac": "38aa3c37247",
        "ssid": "",
        "w": 720,
        "h": 1280,
        "brk": 0,
        "ts": 1432515603,
        "interstitial": 0
    }
},
```

```
{
  "user": {
    "id": "903dced32546b7126aed1b59026058dd",
    "gender": "",
    "yob": 0,
    "ext": {
      "models": []
    }
  },
  "app": {
    "name": "letvVideo",
    "cr": 1,
    "ext": {
      "sdk": "5810",
      "market": 0,
      "appid": "",
      "cat": "",
      "tag": ""
    }
  }
}
```

## 竞价响应（bid-response）

DSP 端返回出价的时候，按照下面 json 格式返回相应的出价和广告素材，不出价的时候，可以直接返回 HTTP 204 状态码或者示例中的空 json。

字段名称	类型	描述
id	string	exchange 生成，用来标识这一次 http 请求，会被记录到竞价日志中
bidid	string	DSP 生成，需要填写到 DSP 的 win-notice 地址中
seatbid	array of objects	bid 对象数组的数组，每个 bid 对象代表针对单次曝光的出价信息。由于 seatbid 中包含的对象本身是个(bid 对象的)数组，所以 seatbid 只有一个元素。

## bid 对象

seatbid 数组中的元素也是一个数组，其中包含多个 bid 对象。

bid 对象数组中的每一个元素表示一次 HTTP 请求中的一个广告曝光请求，在 exchange 的内部实现中，是用 impid 字段来作为唯一标示的。对每一个曝光请求，如果 DSP 选择不参与竞价，就不需要返回对应的 bid 对象，如果一次 HTTP 请求中的所有广告曝光请求都不参与，就直接返回一个空的 bid 数组。具体看下面示例。

下面是 bid 对象的格式。

字段名称	类型	描述
id	string	DSP 生成，针对每一次曝光，同样需要填写到 DSP 的检测地址中
impid	string	exchange 生成，针对每一次曝光，会被记录到竞价日志中。同时 exchange 用这个 ID 来区别多个曝光请求的响应。 本字段是必填的，填的数据必须和 bid request 中的 imp.id 一致。 Exchange 用这个字段将 request 和 response 关联到一起。
price	double	单位为分
nurl	string	win notice url。 win notice 中包含三个宏： \$(DSP_BID_ID)\$ \$(DSP_BID_IMP_ID)\$ \$(DSP_BID_PRICE)\$ 其中： \$(DSP_BID_ID)\$ 会被替换成 bid-response 中的 bidid ， \$(DSP_BID_IMP_ID)\$会被替换成 bid-response 中 bid 对象的 id， \$(DSP_BID_PRICE)\$会被替换成实际成交价。 其中，价格部分会通过 aes 加密，然后通过 base64 编码转换成可打印字符，最后进行 url encode 后放到 url 中，加密时使用 DSP token 作为密钥。解密的时候，需要首先进行 url decode，然后做 base64 解码，最后再用 aes 解密才能得到真实价格。 可以参看下文中 <a href="#">示例</a>  <b>Aes 的算法模式为 ECB，密钥长度为 256 位，补码方式采用 PKCS5Padding。密钥按照字符串来处理（每个字符代表 8bit），超过 32 字节的密钥会被截断。</b>
adm	string	素材 url，这个字段既是素材的实际地址，同时也用来识别素材。 url 必须以"http://"开头。
admpara	string	动态素材拼接参数，如果该参数不为空，则将该字段拼接到 adm 后作为实际展示的素材地址。
ext	object	扩展字段

## ext 对象

字段名称	类型	描述
ldp	string	点击跳转地址。 跳转地址支持重定向，但是不支持跳转 app store 内。 <b>支持 302 跳转</b>
pm	array of string	曝光监测 url <b>曝光监测中可以使用和 nurl 中一样的宏替换</b> 具体查上表中 nurl 字段解释。 <b>不支持 302 跳转。</b>
cm	array of string	点击监测 url， <b>不支持 302 跳转。</b>



		需要注意的是：所有终端上，不管是 PC 还是移动端，曝光和点击监测都不支持 HTTPS，只支持 HTTP。
type	string	素材类型，形式为 MIME Type 必须为具体的 MIME Type，不支持形如 “video/*” 这种形式。

## 示例

### 正常返回：

```
{
  "id": "exchange-id",
  "bidid": "dsp-id-1111",
  "seatbid": [
    {
      "bid": [
        {
          "id": "dsp-imp-id1",
          "impid": "exchange-imp-id1",
          "price": 11.11,
          "nurl": "http://www.dsp1.com/winnotice1",
          "adm": "http://www.cdn.com/a1.jpg",
          "ext": {
            "ldp": "http://www.aol.com",
            "pm": [
              "http://www.letv.com/pv1",
              "http://www.dsp.com/pv1"
            ],
            "cm": [
              "http://www.letv.com/click1",
              "http://www.dsp.com/click1"
            ],
            "type": "image/jpeg"
          }
        }
      ]
    }
  ]
}
```

### 不参与竞价：

不参与竞价可以直接返回 **HTTP 204**状态码或者下面格式的空json

```
{
  "id": "exchange-id",
  "bidid": "dsp-id-1111",
  "seatbid": [
```

```
{
  "bid": []
}
```

## Win-notice

DSP 在实时竞价过程中通过 bid-response 中的 nurl 字段向 exchange 通知 win-notice 发送地址。nurl 中可以包含三种宏，由 exchange 来替换。

例如：nurl 为：

```
http://www.dsp.com/win-notice?bidid=${DSP_BID_ID}$&bidimpid=${DSP_BID_IMP_ID}$&price=${DSP_BID_PRICE}$
```

其中的\${DSP\_BID\_ID}\$会被替换成 bid-response 中的 bidid，\${DSP\_BID\_IMP\_ID}\$会被替换成 bid-response 中 bid 对象的 id，\${DSP\_BID\_PRICE}\$会被替换成实际成交价。

注：宏的名字区分大小写，且均为大写。

其中，价格部分会通过 aes 加密，然后通过 base64 编码转换成可打印字符，最后进行 url encode 后放到 url 中，加密时使用 DSP token 作为密钥。

解密的时候，需要首先进行 url decode，然后做 base64 解码，最后再用 aes 解密才能得到真实价格。

Aes 的算法模式为 ECB，密钥长度为 256 位，补码方式采用 PKCS5Padding。密钥按照字符串来处理（每个字符代表 8bit，就是说需要 32 字节长的密钥），超过 32 字节的密钥会被截断。Exchange 向媒体前端返回时，会将 win-notice 当成 DSP 方的曝光监测地址之一，由前端广告播放器向 DSP 发送 win-notice。

下面是示例，供测试使用：

```
encrypt    123.45    d3fcb17709b94522a709802f00df7e62
6CwjFfLD2%2Fa3OCKjVRjQZw%3D%3D
decrypt    6CwjFfLD2%2Fa3OCKjVRjQZw%3D%3D    d3fcb17709b94522a709802f00df7e62
123.45
```

## Cookie-mapping

Letv exchange 使用 letv.com 的流量资源和 DSP 进行 cookie-mapping，由 DSP 方存储 cookie 数据。具体过程如下：

1. DSP 提供 cookie-mapping 接口地址，例如：  
[http://www.dsp.com/cookie-mapping/letv?id={letv\\_user\\_id}](http://www.dsp.com/cookie-mapping/letv?id={letv_user_id})
2. Letv 通过浏览器，APP 等客户端向 cookie-mapping url 发送请求，请求中 letv\_user\_id 为 letv exchange 中的用户 ID
3. DSP 收到请求后，保存 DSP cookie 中的用户 ID 和 letv\_user\_id 的对应关系。

## 竞价和监测发生流程

在正常无用户干预的情况下，竞价可能比播放提前 0 到 60s:

在不考虑用户端网络延迟的情况下，

对于前贴广告，如果第一个前贴广告由 exchange 系统投出，那么竞价和曝光几乎是同时发生（可能有一些网络延迟）；如果是第五支前贴广告由 exchange 系统投出，那么曝光会在竞价 60s 之后发生。

对于中插广告，竞价发生在广告播放前 30s。

但是在有用户干预的时候，竞价时间和广告播放时间之间的时间差是不确定的，竞价和曝光监测的时间差会拉长，任何时间长度都有可能。

无论任何情况下，曝光监测一定是在广告播放的同时并行发送的，点击监测一定是在点击事件发生时同时并行发送的。

## 联调对接流程

DSP 和 exchange 首先应该对接业务数据接口。因为后续的实时竞价部分需要使用的素材需要先审核通过之后才能真正投放出来。所以测试步骤如下：

### 1. 业务数据接口

业务数据接口有三个，其中[广告信息同步接口](#)是必须的，优先联调这个接口。

[广告位信息](#)和[报表接口](#)看 DSP 具体情况对接。

### 2. 实时竞价接口。

### 3. 对比曝光和点击等数据。

双方的曝光点击等数据吻合之后，可以在线上试投，然后进入正式投放过程。

业务数据接口方面，我们这边有测试地址，供联调使用。

实时竞价方面，DSP 通过广告信息同步接口上传完素材之后，由 exchange 向 DSP 发送竞价请求，广告展示和点击的同时，会发送监测和 winnotice，然后双方对数。

## 对媒体接口

对媒体接口适用于媒体方将资源导入到乐视 Ad Exchange 平台进行交易时参考。

## VAST 3.0 接口

### 接口 URL

乐视 Ad Exchange 的接口 URL 为

<http://ark.letv.com/adx?adzone=123&res=xml>

其中各个参数以及含义如下表：

参数	说明
----	----

adzone	乐视 exchange 广告位 id，可以有多个，用英文逗号分隔。 同一个广告位 id 允许出现多次，只要有多个广告位 ID，就会返回多个广告。
res	响应的类型，可以是 json，jsonp，xml 三者之一
j	当 res=jsonp 时，这个参数指明前端调用的函数名，在请求发生时由前端填入。 例如： 请求是 res=jsonp&j=fun_name 那么返回的 jsonp 数据就是 <i>fun_name (完整 json)</i> ;

此处 adzone 是乐视 exchange 中定义的广告位 ID，媒体方接入时，需要在发送请求时填入对应的广告位 ID。

## VAST 数据格式

VAST 格式分为线性和非线性两种。其中，前贴，中贴，后贴会返回线性格式，暂停广告会返回非线性格式。

示例如下：

线性广告
<pre>&lt;!-- 线性类型素材 --&gt; &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;VAST version="3.0"&gt;   &lt;Ad id="44781"&gt; &lt;!-- 此处的ID是exchange中的分配的广告位ID --&gt;     &lt;InLine&gt;       &lt;AdSystem version="1"&gt;&lt;![CDATA[letv exchange]]&gt;&lt;/AdSystem&gt;       &lt;!-- 曝光地址有1到多个，id属性可以不用 --&gt;  &lt;Impression&gt;&lt;![CDATA[http://ark.letv.com/et?mid=182942]]&gt;&lt;/Impression&gt;  &lt;Impression&gt;&lt;![CDATA[http://vletv.admaster.com.cn/i/a23351,b200550473,c1760,i200002107,m202,h]]&gt;&lt;/Impression&gt;   &lt;Creatives&gt;     &lt;Creative&gt;       &lt;Linear&gt;         &lt;Duration&gt;&lt;![CDATA[00:00:15]]&gt;&lt;/Duration&gt; &lt;!-- 视频时长 --&gt;          &lt;!-- 整个trackingevents字段都暂不支持，只有曝光监测 --&gt;         &lt;TrackingEvents&gt;&lt;/TrackingEvents&gt;          &lt;MediaFiles&gt;           &lt;!--           id是乐视广告平台中的广告素材ID，           delivery 属性暂时不支持，"progressive"参数没有用处，           宽高等属性已经支持，是素材的实际尺寸，           apiFramework 现在不支持，"VPAID"参数没有用处           --&gt;</pre>

```

-->
    <MediaFile id="123"
type="application/x-shockwave-flash" delivery="progressive"
width="640" height="360" apiFramework="VPAID">

<![CDATA[http://i1.letvimg.com/gugwl/201311/04/miaoZhenApp.swf]]>
    </MediaFile>
</MediaFiles>

    <!-- 这个字段可以完全忽略，在后续版本中会删除 -->
    <AdParameters
xmlEncoded="false"><![CDATA[{"tMethod":
"", "reqType":0, "pid":0, "autoPlay":0, "ast":0}]]></AdParameters>
    <VideoClicks>
        <!-- 点击跳转，只能一个 -->

<ClickThrough><![CDATA[http://ark.letv.com/et?mid=182941&u=http%3A%2F
%2Fc.admaster.com.cn%2Cc1760%2Ci200002107%2Cm101%2Ch]]></ClickThrough
>

        <!-- 点击监测，在点击跳转同时请求，可有0到多个 -->

<ClickTracking><![CDATA[http://ark.letv.com/et?mid=182943]]></ClickTr
acking>

<ClickTracking><![CDATA[http://letv.m.cn.miaozhen.com/x.gif?k=1017163
&p=40LJH0&rt=2&ns=[M_ADIP]&ni=[M_IESID]&v=[M_LOC]&o=]]></ClickTrackin
g>

    </VideoClicks>
    </Linear>
    </Creative>
</Creatives>
</InLine>
</Ad>
</VAST>

```

#### 非线性广告

```

<!-- 非线性类型素材 -->
<?xml version="1.0" encoding="UTF-8"?>
<VAST version="3.0" code="" cuid="00000000000000000000000000000000">
    <Ad id="44781"> <!-- 此处的ID是exchange中的分配的广告位ID -->
        <InLine>
            <AdSystem version="1"><![CDATA[letv exchange]]></AdSystem>
            <!-- 有1到多个曝光监测地址 -->

<Impression><![CDATA[http://ark.letv.com/et?mid=191454]]></Impression

```

```
>

<Impression><![CDATA[http://s.cr-nielsen.com/cgi-bin/hat?_t=i&_htsinf
o=dT0yJmM9ODAwMDAwNDEmZz0xMDAwMTQ3NSZhPTMwMDE2NDgx&_htspl=]]></Impres
sion>

<Impression><![CDATA[http://8b85ya.v.admaster.com.cn/i/a17555,b200337
509,c1760,i0,m202,h,uhttp://p-cn.acxiom-online.com/pixel/trc]]></Impr
ession>

<Impression><![CDATA[http://g.cn.miaozhen.com/x.gif?k=1012334&p=4+QS0
0&rt=2&ns=[M_ADIP]&ni=[M_IESID]&na=[M_MAC]&v=[M_LOC]&o=]]></Impressio
n>

    <Creatives>
        <Creative>
            <!-- 只返回一个素材 -->
            <NonLinearAds>
                <!-- minSuggestedDuration保留, 值为空, apiFramework
保留, 值为空 -->
                <!-- width 和 height两个属性需要填上 -->
                <NonLinear minSuggestedDuration="" apiFramework=""
width="640" height="960">
                    <!-- 多个点击监测地址 -->

<NonLinearClickTracking><![CDATA[http://ark.letv.com/et?mid=191788]]>
</NonLinearClickTracking>

<NonLinearClickTracking><![CDATA[http://vletv.admaster.com.cn/i/a2367
6,b200565840,c1762,i0,m202,h]]></NonLinearClickTracking>
    <!-- 一个点击跳转地址 -->

<NonLinearClickThrough><![CDATA[http://ark.letv.com/et?mid=191786&u=h
ttp%3A%2F%2Fc.admaster.com.cn%2F%2Fa22976%2Cb200535951]]></NonLinear
ClickThrough>

                <!-- creativeType 中填写素材类型, 值是素材地址 -->
                <StaticResource creativeType="image/jpeg">

<![CDATA[http://i3.letvimg.com/gugwl/201410/22/23day-640-960.jpg]]>
                </StaticResource>

                <!-- 这个字段可以完全忽略, 在后续版本中会删除 -->
                <AdParameters>
```

```
xmlEncoded="false"><![CDATA[{"url":  
"", "type": "", "duration": ""}]]></AdParameters>  
    </NonLinear>  
    </NonLinearAds>  
    </Creative>  
    </Creatives>  
    </InLine>  
    </Ad>  
</VAST>
```

## 数据字典

## Ad Exchange 广告素材规格说明

### 广告位类型

乐视 ADX 中目前支持视频贴片（包括前贴，中贴，后贴）和暂停两种广告位形式。页面 banner 位正在测试中，近期会上线。

贴片广告位支持视频类型素材。

暂停广告位支持图片类型素材。

### 视频素材

视频素材尺寸：高清 640\*360 以上，且宽、高比例为 4:3 或者 16:9，大小 1M~50M。

视频文件格式：支持 MP4 和 flv 两种格式，mp4 必须为 h.264 编码。

视频时长：15s 素材时长应该小于等于 15s，大于等于 14s。30s 素材时长应该小于等于 30s，大于等于 29s。

视频素材创意：不管素材的比例是 16:9 还是 4:3，都不要人为的添加黑边，广告播放器能正确处理 16:9 和 4:3 比例的素材。视频创意必须有动态效果，不能是静止的图片。

### 暂停素材

pc	410*305 ≤ 60K	支持 jpg 和 png
----	---------------	--------------

Android	410*232 ≤ 30K	支持 jpg 和 png
iPhone	620*300 ≤ 70K	支持 jpg 和 png
iPad	1420*800 ≤ 140K	支持 jpg 和 png

## Banner 素材（测试中）

位置	尺寸	制作要求	格式	备注
PC 网页端 播放页	400*90<20K		swf、gif、png、 jpg	swf 素材中不 可以加任何代 码
	400*180<25K			
	284*40<15K			
	940*90<40K	内容有效区域：水 平居中 562 像素		

## 监测代码

监测代码 url 只支持 http 协议的，不支持 https 等其他协议的。

## 特殊注意点

目前乐视 ADX 上所有贴片广告位的尺寸仍然都是 640\*480，上传素材和投放过程中，可以忽略掉广告位尺寸，上传素材时请使用素材真实尺寸，投放时实际能正常播放的素材尺寸以上表为准。

## 广告位类型表

值	描述
0	页面
1	拉幕
2	前贴片
3	标版
4	中贴片
5	后贴片
6	暂停
7	浮层



素材类型表

素材	对应文件类型	MIME	备注
png	*.png	image/png	PNG 图片
Jpg	*.jpg,* .jpeg	image/jpeg	JPG 或 JPEG 图片
Flv	*.flv	video/x-flv	FLV 视频
mp4	*.mp4	video/mp4	MP4 视频