

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。

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，用来标识使 API 的是哪个 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

101~116 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,gif,png,swf,flv,mp4	
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
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

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	http://ark.letv.com/api/adzone/list
测试接口	http://ark.letv.com/apitest/adzone/list

请求示例：

POST: http://ark.letv.com/api/adzone/list
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	http://ark.letv.com/api/ad/sync
测试接口	http://ark.letv.com/apitest/ad/sync

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，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。

若广告无对应的监测 **monitor** 信息，则参数格式为“**monitor**”:[]

请求示例：

POST: http://ark.letv.com/api/ad/sync
<div>JSON: { "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", //点击跳转落地页的地址，必填 "advertiser": "广告主的中文名称", //必须填写 "startdate":"2014-11-16", //物料生效时间，格式要求：YYYY-mm-dd，必须填写 "enddate":"2014-12-30", //物料失效时间，格式要求：YYYY-mm-dd，必须填写 "monitor": [{"url":"http://impression.com/1", "type":"impression"}, {"url":"http://click.com/2", "type":"click"}], //曝光监测地址：非必填，若"type":""，则默认为 impression，可以有多个 "type":"jpg", //物料类型：必填。 "duration": 0, //时长，必填，且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 "width": 40, "height": 60 //宽和高两个字段可以用数字类型，也可以用字符串类型 }, { "url": "http://somecdn.com/media/v1/01.flv", "landingpage": "http://www.some.com", "advertiser": "广告主的中文名称", "startdate":"2014-11-16", "enddate":"2015-2-26", "monitor": [], "type":"gg", "duration": 15, "width": 20,</div>

```
"height": 30
}]
}
```

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

获取审核未通过的广告信息 API

API 接口: <http://ark.letv.com/api/ad/denylist>

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: <http://ark.letv.com/api/ad/denylist>

JSON:

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

返回示例:

正常返回
<pre>{ "result":0, "message": { "total":2, "records":[{ "url": "http://material.client.com/123.swf", //广告 url "reason": "显示异常" //审核不通过原因 }, { "url": "http://material.client.com/134.swf", "reason": "跳转地址异常" }] } }</pre>
校验出请求参数存在错误时的返回示例
<pre>{ "result":2, "message": "上传时间不能解析" }</pre>

批量获取指定广告的审核状态

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

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

返回示例

正常返回

```

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

```

{
  "result":2,
  "message":
  {
    "302": ["http://material.client.com/135.swf","http://material.client.com/136.swf"]
  }
}

```

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，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, 目前仅使用了 ip 字段
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			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
banner	object	banner 类型广告位, 该字段和 video 字段有且只有一个		
		w	int	广告位宽度
		h	int	广告位高度
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
		display	int	广告位类型, 参看广告位类型表。
		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 广告素材规格说明 就可以正常展示。

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

称			型		
name	string			App 的名称	支持， 提供 APP 名称和版本
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）	移动端支持
		lon	float	经度（-180~180）	移动端支持
		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”	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 http://en.wikipedia.org/wiki/Mobile_Network_Code 。	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	1 表示只接受 PDB 竞价； 0 表示在 PDB 竞价失败的时候，可以接受公开竞价
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。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

```
{
  "id": "a9eedb41f98941e88289f829e238ded0",
  "imp": [
    {
      "id": "exchange-imp-id1",
      "adzoneid": 123,
      "bidfloor": 1.01,
      "banner": { "w": 980, "h": 90 }
    }
  ],
  "site": {
    "name": "",
    "page": "",
    "ref": "",
    "content": {
      "title": "",
      "ext": { "channel": "", "cs": "" }
    },
    "keywords": ""
  },
  "device": {
    "ua": "",
```

```
"ip": "10.58.88.39",
"didmd5": "",
"dpidmd5": "",
"make": "",
"model": "",
"os": "",
"osv": "",
"carrier": "",
"language": "",
"js": 0,
"connectiontype": 0,
"devicetype": 0,
"geo": { "lat": 0.000000, "lon": 0.000000, "ext": { "accuracy": 0 } },
"ext": {
  "idfa": "",
  "mac": "",
  "macmd5": "",
  "ssid": "",
  "w": 0,
  "h": 0,
  "brk": 0,
  "ts": 0,
  "interstitial": 0
}
},
"user": {
  "id": "881996789ad2d2acc3507e435a0497c",
  "gender": "",
  "yob": 0,
  "ext": { "models": [ ] } }
},
"app": {
  "name": "",
  "ext": {
    "sdk": "",
    "market": 0,
    "appid": "",
    "cat": "",
    "tag": ""
  }
}
}
```


竞价响应（bid-response）

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
price	double	单位为分
nurl	string	<p>win notice url。</p> <p>win notice 中包含三个宏：</p> <p>\$(DSP_BID_ID)\$</p> <p>\$(DSP_BID_IMP_ID)\$</p> <p>\$(DSP_BID_PRICE)\$</p> <p>其中：</p> <p>\$(DSP_BID_ID)\$ 会被替换成 bid-response 中的 bidid ，</p> <p>\$(DSP_BID_IMP_ID)\$会被替换成 bid-response 中 bid 对象的 id，</p> <p>\$(DSP_BID_PRICE)\$会被替换成实际成交价。</p> <p>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。</p> <p>可以参看下文中示例</p> <p>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。</p>
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。

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
pm	array of string	曝光检测 url
cm	array of string	点击检测 url
type	string	素材类型，形式为 MIME Type

示例

正常返回
<pre>{ "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" } }] }] }</pre>

```
}
不参与竞价
{
  "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。

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/a3OCkjVRjQZw==
decrypt 6CwjFfLD2/a3OCkjVRjQZw==    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}
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。

移动端素材上传说明

投放移动端的所有视频类型广告素材（mp4 格式）都必须通过乐视转码系统转码并由乐视 CDN 承压。

转码系统有独立的账号密码，在对接阶段，由乐视网向 DSP 方提供，DSP 自行上传，并获取 CDN 上的素材地址。

通过[审核接口](#)上传素材时，需要使用乐视 CDN 的素材地址，否则广告素材不能正常投放。
上传素材后，可以通过[审核状态接口](#)获取所有素材的审核结果以及未通过原因。

对媒体接口

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><!-- 线性类型素材 --> <?xml version="1.0" encoding="UTF-8"?> <VAST version="3.0"> <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> <!-- 曝光地址有1到多个，id属性可以不用 --> <Impression><![CDATA[http://ark.letv.com/et?mid=182942]]></Impression> <Impression><![CDATA[http://vletv.admaster.com.cn/i/a23351,b200550473,c1760,i200002107,m202,h]]></Impression> <Creatives></pre>

```
<Creative>
  <Linear>
    <Duration><![CDATA[00:00:15]]></Duration> <!-- 视
频时长 -->

    <!-- 整个trackingevents字段都暂不支持，只有曝光监测 -->
    <TrackingEvents></TrackingEvents>

    <MediaFiles>
      <!--
      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
      -->
      <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&_htsinfo=dT0yJmM90DAwMDAwNDEmZz0xMDAwMTQ3NSZhPTMwMDE2NDgx&_htspl=]]></Impression>

            <Impression><![CDATA[http://8b85ya.v.admaster.com.cn/i/a17555,b200337509,c1760,i0,m202,h,uhttp://p-cn.acxiom-online.com/pixel/trc]]></Impression>

            <Impression><![CDATA[http://g.cn.miaozhen.com/x.gif?k=1012334&p=4+QS00&rt=2&ns=[M_ADIP]&ni=[M_IESID]&na=[M_MAC]&v=[M_LOC]&o=]]></Impression>

            <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/a23676,b200565840,c1762,i0,m202,h]]></NonLinearClickTracking>
                            <!-- 一个点击跳转地址 -->
```

```

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

        <!-- 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

乐视 Ad Exchange 支持为视频，图片两种类型的广告素材：

视频文件格式为 flv 和 mp4，mp4 必须为 h.264 编码，

图片文件格式为 jpeg，png，gif，其中 gif 只能用单帧，

贴片广告：

素材的时长：不限制最短时间，最长不能超过广告位时长+0.5s

素材的尺寸：最小 400*300，最大 1920*1080，4:3 和 16:9 都可以，播放时会根据播放器缩放。

素材的大小：15s 不要超过 2M，30s 不要超过 4M

素材的类型：PC 端支持 MP4 和 flv，**移动端支持 MP4**

暂停广告：

素材的尺寸：PC：640*480，Android：410*232，iPhone：620*300，iPad：1420*800，**尺寸要求严格匹配**

素材的大小： 不要超过 150k

素材的类型： jpg,png,gif

物料类型	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
gif	*.gif	image/gif	GIF 图片
flv	*.flv	video/x-flv	FLV 视频
mp4	*.mp4	video/mp4	MP4 视频

广告位类型表

值	描述
0	页面广告
1	拉幕广告
2	前贴片广告
3	标版广告
4	中贴片广告
5	后贴片广告
6	暂停广告
7	浮层广告