

VMS API 开发指南

VMS视频 | 视频云时代
内容全在线

版本号：v2.4.2

2016/12/08

成都华栖云科技有限公司

目录

VMS API 开发指南.....	1
1 文档说明.....	5
1.1 文档概述.....	5
2 通信约定.....	5
2.1 通信协议.....	5
2.2 数据交互格式.....	6
2.2.1 数据类型.....	6
2.2.2 编码格式.....	6
2.2.3 错误码定义.....	6
2.3 访问频度（暂时无限制功能）.....	6
2.4 其他约定.....	7
3 数据交互.....	7
3.1 接口调用示意图.....	7
3.2 接口地址.....	8
3.3 请求参数.....	8
3.4 应用举例.....	8
4 接口明细.....	10
4.1 合作伙伴认证信息.....	10
4.1.1 接口说明.....	10
4.1.2 应用示例.....	10
4.2 获取栏目列表信息.....	11
4.2.1 接口说明.....	11
4.2.2 应用示例.....	11
4.3 获取点播视频列表信息.....	16
4.3.1 接口说明.....	16
4.3.2 应用示例.....	16
4.4 获取某个点播视频信息.....	24
4.4.1 接口说明.....	24
4.4.2 应用示例.....	24
4.5 获取直播频道列表信息.....	30
4.5.1 接口说明.....	30
4.5.2 应用示例.....	30
4.6 获取某个直播频道信息.....	35
4.6.1 接口说明.....	35
4.6.2 应用示例.....	35
4.7 获取直播节目单信息.....	39
4.7.1 接口说明.....	39
4.7.2 应用示例接口说明.....	39
4.8 获取电视剧列表信息.....	42
4.8.1 接口说明.....	42
4.8.2 应用示例.....	42

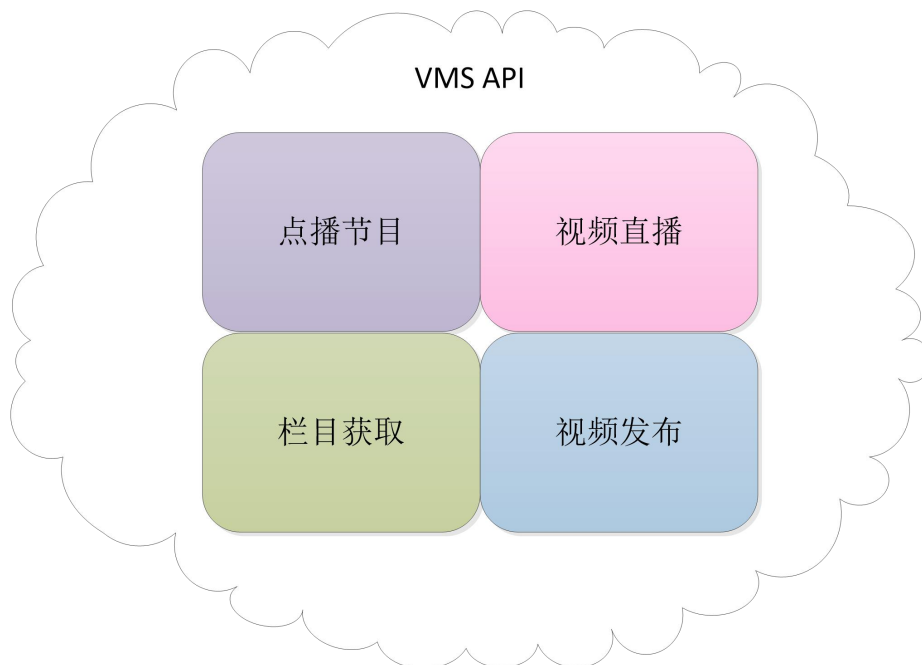
4.9 获取某个电视剧信息.....	46
4.9.1 接口说明.....	46
4.9.2 用示例.....	47
4.10 获取某个电视剧的集数列表.....	49
4.10.1 接口说明.....	49
4.10.2 应用示例.....	49
4.11 获取音频列表信息.....	57
4.11.1 接口说明.....	57
4.11.2 应用示例.....	57
4.12 获取某个音频信息.....	63
4.12.1 接口说明.....	63
4.12.2 应用示例.....	63
4.13 删除视频、音频或电视剧信息.....	65
4.13.1 接口说明.....	65
4.13.2 应用示例.....	65
4.14 获取站点的信息.....	66
4.14.1 接口说明.....	66
4.14.2 应用示例.....	66
4.15 获取转码组的信息.....	67
4.15.1 接口说明.....	67
4.15.2 应用示例.....	68
4.16 获取播放器列表信息.....	69
4.16.1 接口说明.....	69
4.16.2 应用示例.....	69
4.17 获取某个播放器信息.....	71
4.17.1 接口说明.....	71
4.17.2 应用示例.....	71
4.18 更新某个视频信息.....	72
4.18.1 接口说明.....	72
4.18.2 应用示例.....	72
4.19 获取虚拟直播频道列表信息.....	73
4.19.1 接口说明.....	73
4.19.2 应用示例.....	73
4.20 获取某个虚拟直播频道信息.....	75
4.20.1 接口说明.....	75
4.20.2 应用示例.....	75
4.21 获取虚拟直播节目单信息.....	76
4.21.1 接口说明.....	76
4.21.2 应用示例.....	76
4.22 获取所有租户信息.....	76
4.22.1 接口说明.....	76
4.22.2 应用示例.....	76
5 通知机制.....	82

5.1 视频更新通知机制.....	82
5.1.1 接口说明.....	82
5.1.2 应用示例.....	82
5.2 音频更新通知机制.....	82
5.2.1 接口说明.....	82
5.2.2 应用示例.....	82

1 文档说明

1.1 文档概述

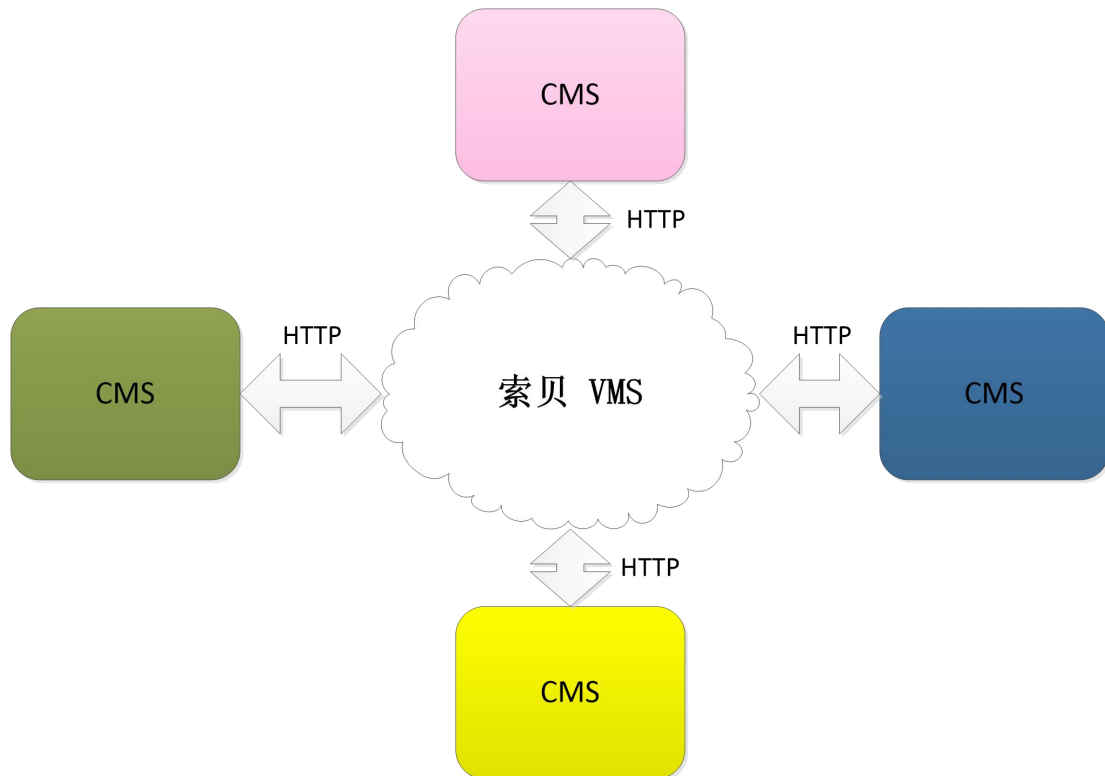
利用 VMS API 可以与索贝 VMS 视频管理系统进行对接，使用索贝 VMS 视频管理系统的主要视频功能。当前 VMS API 的版本号为 v2.4.1，具有获取视频分类信息、获取视频信息、获取直播频道信息、获取某个视频的播放器代码、获取某个直播频道的播放器代码等接口。目前 VMS API 仅对合作方开启，如果您有使用的需求，请通过 VMS 视频管理系统的后台申请 API 合作伙伴的账号。



2 通信约定

2.1 通信协议

索贝 VMS API 的远程通信接口均采用 HTTP 协议，并且都采用 HTTP POST 或者是 HTTP GET 的方式来调用。



2.2 数据交互格式

2.2.1 数据类型

VMS API v2.4.1 的数据交互支持 JSON 数据格式，并且返回的数据编码为 UTF-8 格式。

2.2.2 编码格式

VMS API 只接受 UTF-8 编码格式的编码信息。传入的参数和返回的结果都是 UTF-8 格式的。需要注意的是如果采用 HTTP GET 请求传递参数时，请求的参数 URL 需要进行 URL Encode 之后，再进行传递。否则会造成乱码。

2.2.3 错误码定义

VMS API 针对于不同的接口，正确的返回结果会在每个接口的返回结果中单独定义，如果在请求和相应的过程中发生错误，那么错误的返回错误信息的字符串，如下所示：

“您输入的 partnerToken 不能为空，请重新输入!”

2.3 访问频度（暂时无限制功能）

当用户申请访问 VMS API 并且通过审核之后，默认的请求限制是每分钟 60 次，当用户的请求超过这个频度的时候会返回错误信息，并且该用户的 API 功能将会被禁止，之后所有的请求都将失效，如果上述该值无法满足您的正常需求的时候，可以联系索贝 VMS 相关人员申请提升 API 的请求频。

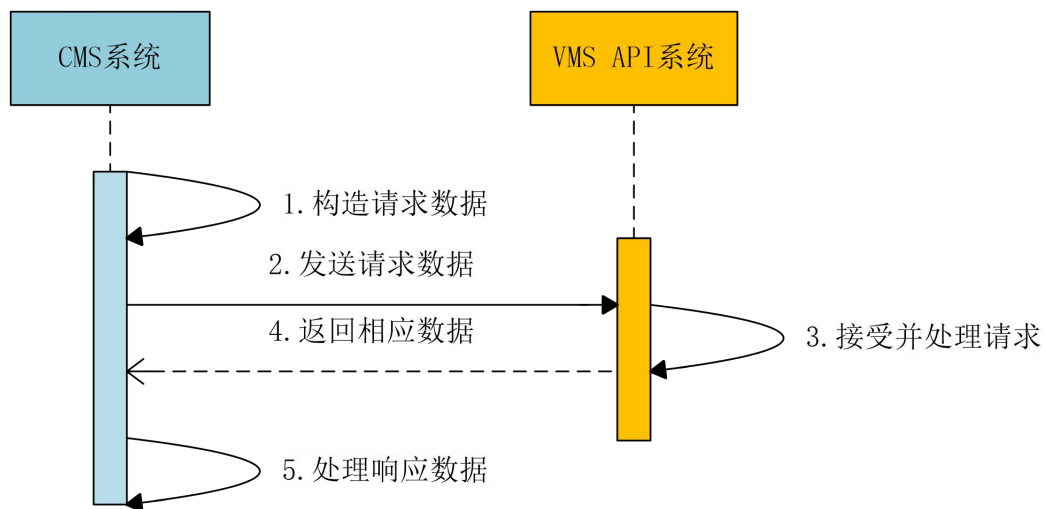
2.4 其他约定

通过 HTTP 传给 VMS API 的参数区分大小写

严禁使用测试接口地址，做线上业务应用。

3 数据交互

3.1 接口调用示意图



1).构造请求数据

用户根据 API 接口的规则（可以参考示例），按照一定的方式组织并且需要传给 VMS API 系统的数据集合。

2).发送数据请求

把构造完成的数据集合，通过 HTTP POST 的请求方式提交给 VMS API 系统

3).接受并处理请求

VMS 接收到传过来的数据集之后，会进行 token 校验，业务处理等。然后会根据请求的数据集返回相应的结果集。

4).返回响应数据

把处理完成的数据集通过 Http Response 返回给合作用户系统

5).处理相应数据

合用户把从 VMS 接口中获取到的数据结合自身的业务逻辑进行数据处理，组织成自己所需要的数据结构。

3.2 接口地址

通过访问 [http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)(测试地址)，并且传入不同的参数，来访问到 VMS API 中不同的接口，此地址为索贝公司测试平台地址，项目对接请以实际的项目地址为准

partnerToken 也由 VMS 提供，不同的接口地址会有不同的 partnerToken，项目对接时请以实际项目给出的为准。

3.3 请求参数

VMS API_v2.4.1 请求的参数全部为简单类型，不需要构造复杂的 XML 格式。在请求接口的时候对于不同的接口需要几个参数，就在 HTTP POST 或者是 HTTP GET 方法中传几个参数。当然如果用 GET 方式请求的时候需要使用一下。其中 method 参数表示请求的接口方法，针对不同的接口 method 的值会有所不同，partnerToken 为获得权限认证之后获取到的 token 令牌也可以叫认证码，在请求不同的接口方法的时候这两个参数每次请求的时候都是必须要传值的，对于其他的参数会有所不同，在之后的文档说明中会给出详细的解释

3.4 应用举例

以获调用 VMS API_v_2.3.2 中调用合作伙伴认证信息的接口为例。

例：http://113.142.30.108:8080/vms/APIServiceReceiver?method=authenticate&partnerToken=***

示例：（method 传入 authenticate 表示该示例调用的是 authenticate 这个方法，也就是调用合作伙伴认证信息的接口）

需要传入以下参数：

参数	说明
partnerToken	用户申请访问接口邮箱发送的 token 值

返回的参数格式：

JSON 格式：

```
{
```



```
{
  "expireDate": "永久有效",
  "partnerName": "test",
  "token": "0c3417f89a759199c5e51a8128355c5a",
  "version": "VMS2.3.2",
  "partnerCode": "test",
  "siteId": "822",
  "siteNames": "ms3.test",
  "accessCount": 0
}
```

返回的参数说明：

参数	说明
partnerName	用户申请接口访问权限时所填写的中文名称
partnerCode	用户申请接口访问权限时所填写的厂商用户的编码。
token	用户的 token 令牌，也是用户访问各个接口的认证码，用户访问每个接口的时候都需要带上这个 token
version	当前的 VMS API 的版本号
siteNames	用户能够访问的站点权限。返回的是站点名称的中文名，如果有多个站点是用 "," 号隔开。比如： siteNames : 山东卫视, 重庆卫视表示用户可以拥有这两个站点的访问权限，在之后访问接口的过程中需要传递站点名称的参数可以根据获取的这个参数来传入，所需要的站点信息。
expireDate	表示当前用户所能访问 VMS API 的过期时间，如果超过这个时间则不能够再访问。
accessCount	访问接口的频度。当用户申请访问 VMS API 并且通过审核之后，默认的请求限制是每分钟 60 次，当用户的请求超过这个频度的时候会返回错误信息，并且该用户的 API 功能将会被禁止，之后所有的请求都将失效，如果上述该值无法满足您的正常需求的时候，可以联系索贝 VMS 相关人员申请提升 API 的请求频。

4 接口明细

4.1 合作伙伴认证信息

4.1.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 **key**，否则没有访问接口的权限。通过 **authenticate** 可以获取认证的基本信息和调用的站点名称。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：[method=authenticate](#)

4.1.2 应用示例

➤ 调用合作伙伴认证信息接口应用举例：

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 **authenticate** 表示该示例调用的是 **authenticate** 这个方法，也就是调用合作伙伴认证信息的接口）

需要传入以下参数：

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌。

返回的参数格式：

JSON 格式：

```
{
  "expireDate": "永久有效",
  "partnerName": "test",
  "token": "0c3417f89a759199c5e51a8128355c5a",
  "version": "VMS2.3.2",
  "partnerCode": "test",
  "siteId": "822",
  "siteNames": "ms3.test",
  "accessCount": 0
}
```

返回的参数说明：

参数	说明
partnerName	用户申请接口访问权限时所填写的中文名称

partnerCode	用户申请接口访问权限时所填写的厂商用户的编码。
token	用户的 token 令牌，也是用户访问各个接口的认证码，用户访问每个接口的时候都需要带上这个 token
version	当前的 VMS API 的版本号
siteNames	siteNames 表示用户可以拥有该站点的访问权限，在之后访问接口的过程中需要传递站点名称的参数可以根据获取的这个参数来传入，所需要的站点信息。
expireDate	表示当前用户所能访问 VMS API 的过期时间，如果超过这个时间则不能够再访问。
accessCount	访问接口的频度。当用户申请访问 VMS API 并且通过审核之后，默认的请求限制是每分钟 60 次，当用户的请求超过这个频度的时候会返回错误信息，并且该用户的 API 功能将会被禁止，之后所有的请求都将失效，如果上述该值无法满足您的正常需求的时候，可以联系索贝 VMS 相关人员申请提升 API 的请求频。（暂时未做限制）
siteId	站点 Id

4.2 获取栏目列表信息

4.2.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 key，否则没有访问接口的权限。用户通过传入站点名称以获取到该站点下的所有栏目分类。在之后获取视频列表的接口中，可以通过传入某个栏目名称以获取到该栏目下的视频。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：`method= getCatalogList`

4.2.2 应用示例

➤ 调用获取栏目列表信息接口应用举例：

本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 getCatalogList 表示该示例调用的是 getCatalogList 这个方法，也就是调获取栏目列表信息的接口）

需要传入以下参数：

注：蓝色为选填

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌。
catalogType	获取栏目的枚举类型 默认值为 5 枚举值：5、6 和 7 获取普通视频的栏目 catalogType 值为 5 获取音频的栏目 catalogType 值为 6 获取电视剧的栏目 catalogType 值为 7
catalogStyle	栏目路径选择传参数的方式 默认值为 0 枚举值：1 和 0 使用 ID 查询时 catalogStyle 值为 1 使用中文全路径查询时 catalogStyle 值为 0
catalogPath	站点名称或者是栏目名称，通过传入这个参数来获取栏目信息，如果只传入的站点名称，则返回该站点下的栏目，如果传入的是站点下的某个栏目，则返回该栏目下的子栏目， (1) 当 catalogStyle 值为 0，则格式为： 测试站点@重庆卫视 注意：如果传入的某个栏目，必须要按照栏目全路径的方式传入，比如要想获取重庆卫视下的新闻栏目下的子栏目那么传入的就应该为：测试站点@重庆卫视@新闻 (2) 当 catalogStyle 值为 1，则格式为： catalogPath 则为全路径叶子节点的 id 值，例如：查询 测试站点@重庆卫视@新闻下面的栏目的时候，则 catalogPath 值为 新闻 栏目的 ID 值；如果查询全站的栏目，则 catalogPath 值为 0
getAllData	枚举值 1 和 0 默认值 0 1：表示返回该栏目下的所有栏目，包括子栏目和子栏目的子栏目

	<p>0 : 表示只返回该栏目下的一级子栏目</p> <p>比如: 假如有这样一个栏目层级</p> <p>测试站点</p> <p>---- 重庆卫视</p> <p>---- 新闻</p> <p>---- 娱乐新闻</p> <p>---- 图片</p> <p>----山东卫视</p> <p>如果传入的 CatalogPath 为: 测试站点@重庆卫视</p> <p>当 getAllData 为 1 时返回 : 新闻, 图片, 娱乐新闻这三个栏目的栏目信息</p> <p>当 getAllData 为 0 时返回 : 新闻, 图片这两个栏目的信息</p>
--	---

返回的参数格式: JSON 格式

```
[
  {
    "catalogId": "20",
    "name": "四级栏目 2",
    "parentId": "14",
    "innerCode": "000008000001000001000002",
    "treeLevel": "4"
  },
  {
    "catalogId": "17",
    "name": "六级栏目字符编码字符很多的情况, 最多字符",
    "parentId": "16",
    "innerCode": "000008000001000001000001000001",
    "treeLevel": "6"
  },
  {
    "catalogId": "16",
    "name": "五级栏目",
    "parentId": "15",
    "innerCode": "000008000001000001000001000001",
    "treeLevel": "5"
  },
  {
    "catalogId": "15",
    "name": "四级栏目",
```

```
    "parentId": "14",
    "innerCode": "000008000001000001000001",
    "treeLevel": "4"
  },
  {
    "catalogId": "14",
    "name": "三级栏目",
    "parentId": "13",
    "innerCode": "000008000001000001",
    "treeLevel": "3"
  },
  {
    "catalogId": "10",
    "name": "整片转码",
    "parentId": "8",
    "innerCode": "000006000002",
    "treeLevel": "2"
  },
  {
    "catalogId": "9",
    "name": "分片转码",
    "parentId": "8",
    "innerCode": "000006000001",
    "treeLevel": "2"
  },
  {
    "catalogId": "8",
    "name": "视频",
    "parentId": "0",
    "innerCode": "000006",
    "treeLevel": "1"
  },
  {
    "catalogId": "7",
    "name": "整片转码",
    "parentId": "5",
    "innerCode": "000005000002",
    "treeLevel": "2"
  },
  {
    "catalogId": "6",
    "name": "分片转码",
    "parentId": "5",
    "innerCode": "000005000001",
```

```

        "treeLevel": "2"
    },
    {
        "catalogId": "5",
        "name": "素材",
        "parentId": "0",
        "innerCode": "000005",
        "treeLevel": "1"
    },
    {
        "catalogId": "4",
        "name": "收录",
        "parentId": "0",
        "innerCode": "000004",
        "treeLevel": "1"
    },
    {
        "catalogId": "1",
        "name": "默认视频栏目",
        "parentId": "0",
        "innerCode": "000001",
        "treeLevel": "1"
    }
}
]

```

返回的参数说明：

参数	说明
catalogId	栏目的 ID 号
innerCode	栏目的 code 编码，为方便管理栏目而设置的字段。如：父栏目 innercode 为 0001 则其子栏目 1 的 innercode 为 00010001，子栏目 2 的 innercode 为 00010002，以此类推。
name	单纯的栏目名称 不包含栏目的全路径
parentId	栏目的父栏目 ID
treeLevel	栏目的级数

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

4.3 获取点播视频列表信息

4.3.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 key，否则没有访问接口的权限。用户可以根据自己需要通过传入站点名称或者是栏目全路径，分页信息和关键字等参数，来获取到自己所需要的视频节目。返回的结果包含视频的播放地址，播放代码，视频名称，所属栏目，视频关键帧，发布状态，视频 ID 等信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：`method= getVideoList`

4.3.2 应用示例

➤ 调用获取点播视频列表信息接口应用举例：

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 getVideoList 表示该示例调用的是 getVideoList 这个方法，也就是调获取点播视频列表信息的接口）

需要传入以下参数：

注：蓝色为选填

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌。
catalogStyle	栏目路径选择传参数的方式 默认值为 0 枚举值：1 和 0 使用 ID 查询时 catalogStyle 值为 1 使用中文全路径查询时 catalogStyle 值为 0
catalogPath	站点名称或者是栏目名称，通过传入这个参数来获取某个站点或者是某个栏目下的视频信息，如果只传入的站点名称，则返回该站点下的视频，如果传入的是站点下的某个栏目，则返回该栏目下的视频信息 （1）当 catalogStyle 值为 0，则格式为： 测试站点@重庆卫视 注意：如果传入的某个栏目，必须要按照栏目全路径的方

	<p>式传入，比如要想获取重庆卫视下的新闻栏目下的视频列表那么传入的就应该为：测试站点@重庆卫视@新闻</p> <p>（2）当 <code>catalogStyle</code> 值为 1，则格式为：</p> <p><code>catalogPath</code> 则为全路径叶子节点的 <code>id</code> 值，例如：查询 测试站点@重庆卫视@新闻下面的视频列表的时候，则 <code>catalogPath</code> 值为 新闻 栏目的 ID 值；如果查询全站的视频，则 <code>catalogPath</code> 值为 0</p>
getAllData	<p>枚举值 1 和 0 默认值为 0</p> <p>1：表示返回该栏目下的所有视频信息，包括子栏目和子栏目的子栏目下的所有视频。</p> <p>0：表示只返回该栏目下的一级子栏目下的视频信息</p> <p>比如：假如有这样一个栏目层级</p> <p>测试站点</p> <ul style="list-style-type: none"> ---- 重庆卫视 ---- 新闻 ---- 娱乐新闻 ---- 图片 ---- 山东卫视 <p>如果传入的 <code>CatalogPath</code> 为：测试站点@重庆卫视</p> <p>当 <code>getAllData</code> 为 1 时返回：新闻，图片，娱乐新闻这三个栏目下的点播视频信息。</p> <p>当 <code>getAllData</code> 为 0 时返回：新闻，图片这两个栏目下的点播视频信息</p>
pageSize	<p>当获取数据需要以分页的方式来显示的话。<code>pageSize</code> 作为分页中的每页显示多少条，如果当此字段为空时，<code>pageSize</code> 默认为 10</p>
pageNum	<p>当获取数据需要以分页的方式来显示的话。<code>pageNum</code> 作为分页中的第几页，如果当此字段为空时，<code>pageNum</code> 默认为 1</p>

startTime	用来查询发布时间段用的开始时间 格式为 yyyy-MM-dd HH:mm:ss
endTime	用来查询发布时间段用的结束时间 格式为 yyyy-MM-dd HH:mm:ss
keywords	用作查询用的关键字，可以为空
sortField	用于返回视频数据排序的排序字段
sort	用于返回视频数据排序的方式
status	-2:全部状态-1:初稿状态 0:待发布 1:发布 2:转码中 3:转码失败 4:回收站 5:流转中 6:流转异常 7:正在上传 8:取消转码 默认值为 1

返回的参数格式：JSON 格式

```
{
  "video": [
    {
      "vodAddress": {
        "duration": 10,
        "title": "ceshi",
        "host": "http://172.16.146.19/cswd/vod/",
        "sourceType": "mp4",
        "formats": [
          "标清"
        ],
        "clips": [
          {
            "index": "001",
            "duration": "10",
            "urls": [
              "2016/08/12/9d10ee2024584df1ae27f100981182bb/9d10ee2024584df1ae27f100981182bb_h264_800k_mp4"
            ]
          }
        ],
        "subPath": "cswd/vod/",
        "editHost": "http://172.16.146.19/cswd/vod/",
        "isSuccess": "1",
        "message": "播出 URL 信息获取成功!"
      },
      "tsAddress": {
```

```
"duration": 10,
"title": "10s",
"host": "http://172.16.146.19/cswd/vod/",
"formats": [
  "流畅"
],
"clips": [
  {
    "index": "001",
    "duration": "10",
    "urls": [
      "2016/08/12/9d10ee2024584df1ae27f100981182bb/9d10ee2024584df1ae27f100981182bb_h264_300k_mp4.mp4"
    ]
  }
],
"subPath": "cswd/vod/",
"isSuccess": "1",
"message": "播出 URL 信息获取成功!"
},
"mp4Address": {
  "duration": 10,
  "title": "10s",
  "host": "http://172.16.146.19/cswd/vod/",
  "formats": [
    "流畅"
  ],
  "clips": [
    {
      "index": "001",
      "duration": "10",
      "urls": [
        "2016/08/12/9d10ee2024584df1ae27f100981182bb/9d10ee2024584df1ae27f100981182bb_h264_300k_mp4.mp4"
      ]
    }
  ],
  "subPath": "cswd/vod/",
  "isSuccess": "1",
  "message": "播出 URL 信息获取成功!"
},
"sourceMediaInfo": {
  "fileSize": "1377",
  "duration": 10000,
```

```

        "videoFormat": "12544",
        "audioFormat": "12800",
        "imageWidth": "720",
        "imageHeight": "576",
        "keyFrameRate": "25"
    },
    "shareSwfUrl": "http://172.16.146.55/vms//servlet/shareservlet/siteid/886/mediaType/5/mediaTime/2016-08-12/cid/9d10ee2024584df1ae27f100981182bb/v.swf",
    "shareEmbedUrl": "<embed width=\\\"640\\\" height=\\\"512\\\" align=\\\"middle\\\" type=\\\"application/x-shockwave-flash\\\" allowscriptaccess=\\\"always\\\" quality=\\\"high\\\" allowFullScreenInteractive = \\\"true\\\" allowfullscreen=\\\"true\\\" src=\\\"http://172.16.146.55/vms//servlet/shareservlet/siteid/886/mediaType/5/mediaTime/2016-08-12/cid/9d10ee2024584df1ae27f100981182bb/playerId/0c49b2b63bf64f2eba710918d59a8489/v.swf\\\" id=\\\"player1\\\">",
    "id": "9d10ee2024584df1ae27f100981182bb",
    "interfacesTypes": "",
    "programLength": "10",
    "duration": 10000,
    "classifyType": 0,
    "title": "ceshi",
    "subTitle": "",
    "tag": "",
    "description": "",
    "createTime": "2016-08-12 14:34:32",
    "status": "1",
    "playTime": "",
    "sourceFrom": "2",
    "catalogId": "1",
    "publishedTime": "2016-08-12 14:36:40",
    "imagePath": "http://172.16.146.53:81/cswd/upload/Image/default/2016/08/12/9d10ee2024584df1ae27f100981182bb/3_640_245.jpg",
    "playerCodeList": [
        {
            "name": "默认播放器",
            "defaultFlag": "1",
            "playerCode": "<![CDATA[<div id=\\\"0c49b2b63bf64f2eba710918d59a8489\\\"></div><script type=\\\"text/javascript\\\">var vmsPlayer_callback = function () {createPlayer(\\\"video://vid:9d10ee2024584df1ae27f100981182bb\\\",\\\"%3Fguid%3D9d10ee2024584df1ae27f100981182bb%26siteid%3D886\\\",\\\"2016-08-12\\\",\\\"@WIDTH@\\\",\\\"@HEIGHT@\\\")};</script><script type=\\\"text/javascript\\\" src=\\\"http://172.16.146.53//cswd/media/playerJson/vod/0c49b2b63bf64f2eba710918d59a8489.js\\\"></script>]]>",
            "staticFilePaths": {
                "playJS": "http://172.16.146.53/cswd//media/playerJson/newvod/0c49b2b63bf64f2eba710918d59a8489.js",
                "H5Soplayer": "http://172.16.146.53/cswd//media/html5/player/dist/sojs-module/H5Soplayer.js"
            }
        }
    ]
}

```

```

    }
  },
  {
    "name": "test",
    "defaultFlag": "0",
    "playerCode": "<![CDATA[<div id=\"ccf346e73a4f465e9b0308b4db0533a6\"></div><script type=\"text/javascript\">var vmsPlayer_callback = function () {createPlayer(\"video://vid:9d10ee2024584df1ae27f100981182bb\", \"%3Fguid%3D9d10ee2024584df1ae27f100981182bb%26siteId%3D886\", \"2016-08-12\", \"@WIDTH@\", \"@HEIGHT@\")};</script><script type=\"text/javascript\" src=\"http://172.16.146.53//cswd/media/playerJson/vod/ccf346e73a4f465e9b0308b4db0533a6.js\"></script>]]>",
    "staticFilePaths": {
      "playJS": "http://172.16.146.53/cswd//media/playerJson/newvod/ccf346e73a4f465e9b0308b4db0533a6.js",
      "H5Soplayer": "http://172.16.146.53/cswd//media/html5/player/dist/sojs-module/H5Soplayer.js"
    }
  },
  {
    "name": "打点",
    "defaultFlag": "0",
    "playerCode": "<![CDATA[<div id=\"d275cae5f92040f8831f4061ccf177f0\"></div><script type=\"text/javascript\">var vmsPlayer_callback = function () {createPlayer(\"video://vid:9d10ee2024584df1ae27f100981182bb\", \"%3Fguid%3D9d10ee2024584df1ae27f100981182bb%26siteId%3D886\", \"2016-08-12\", \"@WIDTH@\", \"@HEIGHT@\")};</script><script type=\"text/javascript\" src=\"http://172.16.146.53//cswd/media/playerJson/vod/d275cae5f92040f8831f4061ccf177f0.js\"></script>]]>",
    "staticFilePaths": {
      "playJS": "http://172.16.146.53/cswd//media/playerJson/newvod/d275cae5f92040f8831f4061ccf177f0.js",
      "H5Soplayer": "http://172.16.146.53/cswd//media/html5/player/dist/sojs-module/H5Soplayer.js"
    }
  }
]
},
"total": "1",
"pageNum": "1",
"pageSize": "10",
"pageTotal": "1",
"sort": "ASC",
"sortField": "PublishDate"}

```

返回的参数说明：

参数	说明
total	获取视频的总条数
pageTotal	视频通过传入的分页参数统计出来的分页总数
pageSize	每页显示的条数
pageNum	当前处于第几页
sortField	排序的字段名称
sort	排序的方式
title	视频节目的节目名称
tag	视频的标签
catalogId	该视频所属的栏目的 ID
id	视频的 GUID （是一个 32 位的唯一标识）
createTime	视频节目的创建时间。
publishedTime	视频节目的发布时间
imagePath	该视频节目的关键帧图片
status	枚举值 -1:初稿状态 0:待发布 1:发布 2:转码中 3:转码失败 4:回收站 5:流转中 6:流转异常 7:正在上传 8:取消转码
classifyType	频道类型 枚举值（根据实际项目的定义的枚举值确定）
custom	自定义字段的 JSONObject (key 值需要根据项目实际自定义的 key，并且每个 key 的 value 都是字符串类型)
vodAddress mp4Address tsAddress	该节目的多码率分片播放地址 vodAddress ->isSuccess: 1: 获取的播出地址及可用 0: 解析播放URL出现问题，或者因为转码没有完成未产生可用播出URL vodAddress ->message: 获取播出URL的返回信息 vodAddress ->duration : 该节目的总时长 vodAddress ->title : 视频的名称

	<p>vodAddress ->host: 视频的播放前缀</p> <p>vodAddress ->subPath: 介于视频播放域名和url间的相对路径目录</p> <p>vodAddress ->clips: 包含了分片视频的信息</p> <p>vodAddress ->formats: 多中码率对应的码率标题 比如是高清, 标清, 还是流畅等</p> <p>vodAddress ->clips->duration: 该分片视频的时长</p> <p>vodAddress ->clips->title: 该分片视频的名称</p> <p>vodAddress ->clips->urls: 该分片的不同码率的播放地址 其中完整的播放地址为: host 的播放前缀 + clips->urls[i] 并且 urls 中的码率顺序对应着 formats 中的码率标题的顺序</p> <p>比如说: url 中的第一个播放地址为 450K, formats 中的第一个值为标清, 那么 450K 就对应到标清。</p>
playerCodeList	<p>该节目的多个播放器代码</p> <p>playerCodeList包含了多个player, 其中player节点中包含了播放器的名称,播放器代码, 是否是默认播放器等信息</p> <p>playerName:播放器名称</p> <p>playerCode:播放器代码</p> <p>defaultFlag:是否是默认播放器</p> <p>playJS: 播放模板 js 资源</p> <p>H5Soplayer: H5播放器资源</p>
interfacesTypes	<p>下发的类型平台</p> <p>ChinaCache :代表蓝汛CDN</p> <p>FastWeb:代表快网CDN</p> <p>Letv:代表乐视CDN</p>
subTitle	视频的副标题
description	视频描述
programLength	节目时长

playTime	节目首播时间
sourceFrom	来源系统 id 枚举值 1: 拍客上传 2: Web 上传 3: Ftp 上传 4: 综合入库 5: 在线快编 6: 视频收录 7: 迁移入库 8: 主干入库 9: 素材转码

4.4 获取某个点播视频信息

4.4.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 key，否则没有访问接口的权限。用户通过传入视频 id 来获取到对应的该视频的信息，返回的结果包含视频的播放地址，播放代码，视频名称，所属栏目，视频关键帧，发布状态，剧集信息，视频 ID 等信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：`method= getVideoById`

4.4.2 应用示例

➤ 调用获取某个直播频道信息接口应用举例：

本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 getVideoById 表示该示例调用的是 getVideoById 这个方法，也就是调获取某个点播视频信息的接口）

需要传入以下参数：

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌。
videoId	查询的视频 id

返回的参数格式：

JSON 格式:

```
{
  "video": [
    {
      "vodAddress": {
        "duration": 10,
        "title": "ceshi",
        "host": "http://172.16.146.19/cswd/vod/",
        "sourceType": "mp4",
        "formats": [
          "标清"
        ],
        "clips": [
          {
            "index": "001",
            "duration": "10",
            "urls": [
              "2016/08/12/9d10ee2024584df1ae27f100981182bb/9d10ee2024584df1ae27f100981182bb_h264_800k_mp4"
            ]
          }
        ],
        "subPath": "cswd/vod/",
        "editHost": "http://172.16.146.19/cswd/vod/",
        "isSuccess": "1",
        "message": "播出 URL 信息获取成功!"
      },
      "tsAddress": {
        "duration": 10,
        "title": "10s",
        "host": "http://172.16.146.19/cswd/vod/",
        "formats": [
          "流畅"
        ],
        "clips": [
          {
            "index": "001",
            "duration": "10",
            "urls": [
              "2016/08/12/9d10ee2024584df1ae27f100981182bb/9d10ee2024584df1ae27f100981182bb_h264_300k_mp4.mp4"
            ]
          }
        ]
      }
    }
  ]
}
```

```

    ],
    "subPath": "cswd/vod/",
    "isSuccess": "1",
    "message": "播出 URL 信息获取成功!"
  },
  "mp4Address": {
    "duration": 10,
    "title": "10s",
    "host": "http://172.16.146.19/cswd/vod/",
    "formats": [
      "流畅"
    ],
    "clips": [
      {
        "index": "001",
        "duration": "10",
        "urls": [
          "2016/08/12/9d10ee2024584df1ae27f100981182bb/9d10ee2024584df1ae27f100981182bb_h264_300k_mp4.mp4"
        ]
      }
    ],
    "subPath": "cswd/vod/",
    "isSuccess": "1",
    "message": "播出 URL 信息获取成功!"
  },
  "sourceMediaInfo": {
    "fileSize": "1377",
    "duration": 10000,
    "videoFormat": "12544",
    "audioFormat": "12800",
    "imageWidth": "720",
    "imageHeight": "576",
    "keyFrameRate": "25"
  },
  "shareSwfUrl": "http://172.16.146.55/vms//servlet/shareservlet/siteid/886/mediaType/5/mediaTime/2016-08-12/cid/9d10ee2024584df1ae27f100981182bb/v.swf",
  "shareEmbedUrl": "<embed width=\"640\" height=\"512\" align=\"middle\" type=\"application/x-shockwave-flash\" allowscriptaccess=\"always\" quality=\"high\" allowFullScreenInteractive = \"true\" allowfullscreen=\"true\" src=\"http://172.16.146.55/vms//servlet/shareservlet/siteid/886/mediaType/5/mediaTime/2016-08-12/cid/9d10ee2024584df1ae27f100981182bb/playerId/0c49b2b63bf64f2eba710918d59a8489/v.swf\" id=\"player1\">",
  "id": "9d10ee2024584df1ae27f100981182bb",
  "interfacesTypes": "",

```

```

    "programLength": "10",
    "duration": 10000,
    "classifyType": 0,
    "title": "ceshi",
    "subTitle": "",
    "tag": "",
    "description": "",
    "createTime": "2016-08-12 14:34:32",
    "status": "1",
    "playTime": "",
    "sourceFrom": "2",
    "catalogId": "1",
    "publishedTime": "2016-08-12 14:36:40",
    "imagePath": "http://172.16.146.53:81/cswd/upload/Image/default/2016/08/12/9d10ee2024584df1ae27f100981182bb/3_640_245.jpg",
    "playerCodeList": [
        {
            "name": "默认播放器",
            "defaultFlag": "1",
            "playerCode": "<![CDATA[<div id=\"0c49b2b63bf64f2eba710918d59a8489\"></div><script type=\"text/javascript\">var vmsPlayer_callback = function () {createPlayer(\"video://vid:9d10ee2024584df1ae27f100981182bb\", \"%3Fguid%3D9d10ee2024584df1ae27f100981182bb%26siteId%3D886\", \"2016-08-12\", \"@WIDTH@\", \"@HEIGHT@\")};</script><script type=\"text/javascript\" src=\"http://172.16.146.53//cswd/media/playerJson/vod/0c49b2b63bf64f2eba710918d59a8489.js\"></script>]]>",
            "staticFilePaths": {
                "playJS": "http://172.16.146.53/cswd//media/playerJson/newvod/0c49b2b63bf64f2eba710918d59a8489.js",
                "H5Soplayer": "http://172.16.146.53/cswd//media/html5/player/dist/sojs-module/H5Soplayer.js"
            }
        },
        {
            "name": "test",
            "defaultFlag": "0",
            "playerCode": "<![CDATA[<div id=\"ccf346e73a4f465e9b0308b4db0533a6\"></div><script type=\"text/javascript\">var vmsPlayer_callback = function () {createPlayer(\"video://vid:9d10ee2024584df1ae27f100981182bb\", \"%3Fguid%3D9d10ee2024584df1ae27f100981182bb%26siteId%3D886\", \"2016-08-12\", \"@WIDTH@\", \"@HEIGHT@\")};</script><script type=\"text/javascript\" src=\"http://172.16.146.53//cswd/media/playerJson/vod/ccf346e73a4f465e9b0308b4db0533a6.js\"></script>]]>",
            "staticFilePaths": {
                "playJS": "http://172.16.146.53/cswd//media/playerJson/newvod/ccf346e73a4f465e9b0308b4db0533a6.js",
                "H5Soplayer": "http://172.16.146.53/cswd//media/html5/player/dist/sojs-module/H5Soplayer.js"
            }
        }
    ]

```

```

    }
  },
  {
    "name": "打点",
    "defaultFlag": "0",
    "playerCode": "<![CDATA[<div id=\"d275cae5f92040f8831f4061ccf177f0\"></div><script type=\"text/javascript\">var vmsPlayer_callback = function () {createPlayer(\"video://vid:9d10ee2024584df1ae27f100981182bb\", \"%3Fguid%3D9d10ee2024584df1ae27f100981182bb%26siteId%3D886\", \"2016-08-12\", \"@WIDTH@\", \"@HEIGHT@\")};</script><script type=\"text/javascript\" src=\"http://172.16.146.53/cswd/media/playerJson/vod/d275cae5f92040f8831f4061ccf177f0.js\"></script>]]>",
    "staticFilePaths": {
      "playJS": "http://172.16.146.53/cswd//media/playerJson/newvod/d275cae5f92040f8831f4061ccf177f0.js",
      "H5Soplayer": "http://172.16.146.53/cswd//media/html5/player/dist/sojs-module/H5Soplayer.js"
    }
  }
]
},
{
  "total": "1",
  "pageNum": "1",
  "pageSize": "10",
  "pageTotal": "1",
  "sort": "ASC",
  "sortField": "PublishDate"
}

```

返回的参数说明：

参数	说明
total	获取视频的总条数
title	视频节目的节目名称
tag	视频的标签
catalogId	该视频所属的栏目的 ID
id	视频的 GUID （是一个 32 位的唯一标识）
subTitle	视频的副标题
description	视频描述
programLength	节目时长
playTime	节目首播时间

sourceFrom	<p>来源系统 id 枚举值</p> <p>1: 拍客上传</p> <p>2: Web 上传</p> <p>3: Ftp 上传</p> <p>4: 综合入库</p> <p>5: 在线快编</p> <p>6: 视频收录</p> <p>7: 迁移入库</p> <p>8: 主干入库</p> <p>9: 素材转码</p>
createTime	视频节目的创建时间。
publishedTime	视频节目的发布时间
imagePath	该视频节目的关键帧图片
status	<p>枚举值</p> <p>-1:初稿状态 0:待发布 1:发布 2:转码中 3:转码失败 4:回收站 5:流转中 6:流转异常 7:正在上传 8:取消转码</p>
classifyType	频道类型 枚举值（根据实际项目的定义的枚举值确定）
custom	<p>自定义字段的 JSONObject</p> <p>(key 值需要根据项目实际自定义的 key，并且每个 key 的 value 都是字符串类型)</p>
playerCodeList	<p>该节目的多个播放器代码</p> <p>playerCodeList包含了多个player，其中player节点中包含了播放器的名称,播放器代码，是否是默认播放器等信息</p> <p>playerName:播放器名称</p> <p>playerCode:播放器代码</p> <p>defaultFlag:是否是默认播放器</p> <p>playJS: 播放模板 js 资源</p> <p>H5Soplayer: H5播放器资源</p>
interfacesTypes	下发的类型平台

	<p>ChinaCache :代表蓝汛CDN</p> <p>FastWeb:代表快网CDN</p> <p>Letsv:代表乐视CDN</p>
--	--

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

4.5 获取直播频道列表信息

4.5.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 **key**，否则没有访问接口的权限。用户通过调用 **getChannelList** 获取到该站点下的所有直播频道的信息，包括直播频道的名称，简介等信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：**method= getChannelList**

4.5.2 应用示例

➤ **调用获取直播频道列表信息接口应用举例：**

本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 **getChannelList** 表示该示例调用的是 **getChannelList** 这个方法，也就是调获取直播频道列表信息的接口）

需要传入以下参数：

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌。
channelType	频道类型， 默认值为 8 枚举值： 8 （电视直播） 15 （广播直播） 16 （虚拟直播） 17 （设备直播） 18 （纳加直播）
type	获取直播频道直播流的类型 默认为全部类型 枚举值： PC 、 IOS 、 ANDROID 和 TV 不填写表示获取全部类型

返回的参数格式：

JSON 格式：

```
[
  {
```

```

    "id": "97b5c638353f4d818b0dbda668610bf1",
    "name": "test",
    "level": "",
    "channelType": 8,
    "summary": "",
    "channelImgUrl": "",
    "channelActHost": "http://172.16.146.53//cswd/liveChannel",
    "playerCode": "<![CDATA[<link href='\"http://172.16.146.53//cswd/media/playerJson//styles/vms_style.css\"' rel='\"stylesheet\"' type='\"text/css\"'><div class='\"vms-player playerWrapper\"'><div class='\"vms-player-live\"'><div class='\"vms-player-panel\"'><div id='\"livePanel2\"'></div></div></div><div class='\"vms-player-main\"' id='\"flashdiv\"'><div id='\"5e57a83f6216400991ac1c80203bfa7d\"'></div></div></div><script type='\"text/javascript\"'>var vmsPlayer_callback = function () {createLivePlayer(\"97b5c638353f4d818b0dbda668610bf1\", \"@WIDTH@\", \"@HEIGHT@\");};</script><script type='\"text/javascript\"' src='\"http://172.16.146.53//cswd/media/playerJson/liveChannel/5e57a83f6216400991ac1c80203bfa7d.js\"'></script>]]>",
    "liveAddress": [
        {
            "type": "PC",
            "formats": [
                {
                    "title": "高清",
                    "fileTypeId": "h264_800k_flv",
                    "liveStream": "http://appstreamings.hrbtv.net/live/392a5d168cef46ceae6af7462e678789.m3u8?fmt=h264_400k_ts&m3u8",
                    "streamSource": 2,
                    "url": "http://appstreamings.hrbtv.net/live/392a5d168cef46ceae6af7462e678789.m3u8?fmt=h264_400k_ts&m3u8"
                }
            ]
        }
    ],
    "playerCodeList": [
        {
            "name": "默认播放器",
            "defaultFlag": "1",
            "playerCode": "<![CDATA[<link href='\"http://172.16.146.53//cswd/media/playerJson//styles/vms_style.css\"' rel='\"stylesheet\"' type='\"text/css\"'><div class='\"vms-player playerWrapper\"'><div class='\"vms-player-live\"'><div class='\"vms-player-panel\"'><div id='\"livePanel2\"'></div></div></div><div class='\"vms-player-main\"' id='\"flashdiv\"'><div id='\"5e57a83f6216400991ac1c80203bfa7d\"'></div></div></div><script type='\"text/javascript\"'>var vmsPlayer_callback = function () {createLivePlayer(\"97b5c638353f4d818b0dbda668610bf1\", \"@WIDTH@\", \"@HEIGHT@\");};</script><script type='\"text/javascript\"' src='\"http://172.16.146.53//cswd/media/playerJson/liveChannel/5e57a83f6216400991ac1c80203bfa7d.js\"'></script>]]>",
            "staticFilePaths": {
                "playlistJS": "http://172.16.146.53//cswd//media/playerJson/dist/playerjs-module/vms-playlist_right_v1.0.js",
            }
        }
    ]
}

```

```

        "playlistCSS": "http://172.16.146.53/cswd//media/playerJson/styles/playlist_style_rig
ht_v1.0.css",
        "playJS": "http://172.16.146.53/cswd//media/playerJson/newliveChannel/5e57a83f6
216400991ac1c80203bfa7d.js",
        "H5Soplayer": "http://172.16.146.53/cswd//media/html5/player/dist/sojs-module/H
5Soplayer.js"
    },
    {
        "name": "test",
        "defaultFlag": "0",
        "playerCode": "<![CDATA[<link href=\"http://172.16.146.53//cswd/media/playerJson//sty
les/vms_style.css\" rel=\"stylesheet\" type=\"text/css\"><div class=\"vms-player playerWrapper\"><div clas
s=\"vms-player-live\"><div class=\"vms-player-panel\"><div id=\"livePanel2\"></div></div></div><div class=\"
vms-player-main\" id=\"flashdiv\"><div id=\"a1ad4d3865c6491bb5ee4423b83ee392\"></div></div></div><s
cript type=\"text/javascript\">var vmsPlayer_callback = function () {createLivePlayer(\"97b5c638353f4d818
b0dbda668610bf1\", \"@WIDTH@\", \"@HEIGHT@\");};</script><script type=\"text/javascript\" src=\"http://1
72.16.146.53//cswd/media/playerJson/liveChannel/a1ad4d3865c6491bb5ee4423b83ee392.js\"></script>]]>",
        "staticFilePaths": {
            "playlistJS": "http://172.16.146.53/cswd//media/playerJson/dist/playerjs-module/vm
s-playlist_right_v1.0.js",
            "playlistCSS": "http://172.16.146.53/cswd//media/playerJson/styles/playlist_style_rig
ht_v1.0.css",
            "playJS": "http://172.16.146.53/cswd//media/playerJson/newliveChannel/a1ad4d386
5c6491bb5ee4423b83ee392.js",
            "H5Soplayer": "http://172.16.146.53/cswd//media/html5/player/dist/sojs-module/H
5Soplayer.js"
        }
    }
],
{
    "id": "356d4576424c4d4c8b2cd37921213044",
    "name": "test2",
    "level": "",
    "channelType": 8,
    "summary": "",
    "channellmgUrl": "",
    "channelActHost": "http://172.16.146.53//cswd/liveChannel",
    "playerCode": "<![CDATA[<link href=\"http://172.16.146.53//cswd/media/playerJson//styles/vms_s
tyle.css\" rel=\"stylesheet\" type=\"text/css\"><div class=\"vms-player playerWrapper\"><div class=\"vms-p
layer-live\"><div class=\"vms-player-panel\"><div id=\"livePanel2\"></div></div></div><div class=\"vms-play
er-main\" id=\"flashdiv\"><div id=\"5e57a83f6216400991ac1c80203bfa7d\"></div></div></div><script type=
\"text/javascript\">var vmsPlayer_callback = function () {createLivePlayer(\"356d4576424c4d4c8b2cd37921

```



```

213044\\",\\"@WIDTH@\\",\\"@HEIGHT@\\");</script><script type=\\\"text/javascript\\\" src=\\\"http://172.16.146.
53//cswd/media/playerJson/liveChannel/5e57a83f6216400991ac1c80203bfa7d.js\\\"></script>]]>\",
    \"liveAddress\": [
        {
            \"type\": \"PC\",
            \"formats\": [
                {
                    \"title\": \"高清\",
                    \"fileTypeId\": \"h264_800k_flv\",
                    \"liveStream\": \"\",
                    \"streamSource\": 1,
                    \"url\": \"http://172.16.146.19/live/356d4576424c4d4c8b2cd37921213044?fmt=h2
64_800k_flv\"
                }
            ]
        }
    ],
    \"playerCodeList\": [
        {
            \"name\": \"默认播放器\",
            \"defaultFlag\": \"1\",
            \"playerCode\": \"<![CDATA[<link href=\\\"http://172.16.146.53//cswd/media/playerJson//sty
les/vms_style.css\\\" rel=\\\"stylesheet\\\" type=\\\"text/css\\\"><div class=\\\"vms-player playerWrapper\\\"><div clas
s=\\\"vms-player-live\\\"><div class=\\\"vms-player-panel\\\"><div id=\\\"livePanel2\\\"></div></div></div><div class=\\
vms-player-main\\\" id=\\\"flashdiv\\\"><div id=\\\"5e57a83f6216400991ac1c80203bfa7d\\\"></div></div></div><scr
ipt type=\\\"text/javascript\\\">var vmsPlayer_callback = function () {createLivePlayer(\\\"356d4576424c4d4c8b
2cd37921213044\\\",\\\"@WIDTH@\\\",\\\"@HEIGHT@\\\");</script><script type=\\\"text/javascript\\\" src=\\\"http://17
2.16.146.53//cswd/media/playerJson/liveChannel/5e57a83f6216400991ac1c80203bfa7d.js\\\"></script>]]>\",
            \"staticFilePaths\": {
                \"playlistJS\": \"http://172.16.146.53/cswd//media/playerJson/dist/playerjs-module/vm
s-playlist_right_v1.0.js\",
                \"playlistCSS\": \"http://172.16.146.53/cswd//media/playerJson/styles/playlist_style_rig
ht_v1.0.css\",
                \"playJS\": \"http://172.16.146.53/cswd//media/playerJson/newliveChannel/5e57a83f6
216400991ac1c80203bfa7d.js\",
                \"H5Soplayer\": \"http://172.16.146.53/cswd//media/html5/player/dist/sojs-module/H
5Soplayer.js\"
            }
        },
        {
            \"name\": \"test\",
            \"defaultFlag\": \"0\",
            \"playerCode\": \"<![CDATA[<link href=\\\"http://172.16.146.53//cswd/media/playerJson//sty
les/vms_style.css\\\" rel=\\\"stylesheet\\\" type=\\\"text/css\\\"><div class=\\\"vms-player playerWrapper\\\"><div clas

```

```
s="\vms-player-live\""><div class="\vms-player-panel\""><div id="\livePanel2\""></div></div></div><div class="\vms-player-main\" id="\flashdiv\""><div id="\a1ad4d3865c6491bb5ee4423b83ee392\""></div></div></div><script type="\text/javascript\"">var vmsPlayer_callback = function () {createLivePlayer("\356d4576424c4d4c8b2cd37921213044","\">@WIDTH@\","\">@HEIGHT@\");};</script><script type="\text/javascript\" src="\http://172.16.146.53//cswd/media/playerJson/liveChannel/a1ad4d3865c6491bb5ee4423b83ee392.js\""></script>]]>";

    "staticFilePaths": {
        "playlistJS": "http://172.16.146.53/cswd//media/playerJson/dist/playerjs-module/vms-playlist_right_v1.0.js",
        "playlistCSS": "http://172.16.146.53/cswd//media/playerJson/styles/playlist_style_right_v1.0.css",
        "playJS": "http://172.16.146.53/cswd//media/playerJson/newliveChannel/a1ad4d3865c6491bb5ee4423b83ee392.js",
        "H5Soplayer": "http://172.16.146.53/cswd//media/html5/player/dist/sojs-module/H5Soplayer.js"
    }
}
]
}
]
```

返回的参数说明:

参数	说明
id	直播频道 ID
name	直播频道的名称
channelType	枚举值: 8 (电视直播) 15 (广播直播) 16 (虚拟直播) 17 (设备直播) 18 (纳加直播)
channelImgUrl	直播平台的封面图
liveAddress	<p>直播的播放地址信息, 包括了多个终端和多码率的信息</p> <p>liveTypes: 表示终端类型其中 p2p 用于加速的特殊配置;</p> <p>type 表示终端类型, 比如是 pc, android 还是 ios。</p> <p>liveTypes->formats 表示该终端类型下的播放地址, 多码率的可以有多组</p> <p>liveTypes->formats->title : formats 下的 title 属性表示这个码率对应的是高清还是标清</p> <p>liveTypes->formats->url: 表示该码率下的播放地址</p> <p>liveTypes->formats->streamSource:标示码流来源</p>

	枚举值：1 标示来自索贝流媒体 其 URL 为相对 URL 地址 枚举值：2 标示来自第三方流媒体，其 URL 为完整 HTTP 地址
currentName	当前正在播放的直播节目名称。
playerCode	直播频道的播放器代码。把这段代码可以嵌入到自己的网页中，或者是嵌入到文稿编辑的界面中发布出去。
summary	直播频道的简介
playerCodeList	name: 播放器名称 defaultFlag: 是否是默认播放器 1 标示是,0 标示否 playerCode: 播放器代码 staticFilePaths: 播放器静态资源文件 playlistJS: 节目单 js 资源 playlistCSS: 节目单 CSS 资源 playJS: 播放模板 js 资源 H5Soplayer: H5 播放器资源

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

4.6 获取某个直播频道信息

4.6.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 key，否则没有访问接口的权限。用户通过传入直播频道来获取到对应的该直播频道的信息，包括直播频道的名称，简介等信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：[method= getChannelById](#)

4.6.2 应用示例

➤ **调用获取直播频道列表信息接口应用举例：**

本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 getChannelById 表示该示例调用的是 getChannelById 这个方法，也就是调获取某个直播频道信息的接口）

需要传入以下参数：

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌。
channelId	查询的直播频道 id
type	获取直播频道直播流的类型 默认为获取全部类型 枚举值：PC、IOS、ANDROID 和 TV 不填写表示获取全部类型

返回的参数格式：

JSON 格式：

```
[
  {
    "id": "356d4576424c4d4c8b2cd37921213044",
    "name": "test2",
    "level": "",
    "channelType": 8,
    "summary": "",
    "channelImgUrl": "",
    "channelActHost": "http://172.16.146.53//cswd/liveChannel",
    "playerCode": "<![CDATA[<link href=\"http://172.16.146.53//cswd/media/playerJson//styles/vms_style.css\" rel=\"stylesheet\" type=\"text/css\"><div class=\"vms-player playerWrapper\"><div class=\"vms-player-live\"><div class=\"vms-player-panel\"><div id=\"livePanel2\"></div></div></div><div class=\"vms-player-main\" id=\"flashdiv\"><div id=\"5e57a83f6216400991ac1c80203bfa7d\"></div></div></div><script type=\"text/javascript\">var vmsPlayer_callback = function () {createLivePlayer(\"356d4576424c4d4c8b2cd37921213044\", \"@WIDTH@\", \"@HEIGHT@\");};</script><script type=\"text/javascript\" src=\"http://172.16.146.53//cswd/media/playerJson/liveChannel/5e57a83f6216400991ac1c80203bfa7d.js\"></script>]]>",
    "liveAddress": [
      {
        "type": "PC",
        "formats": [
          {
            "title": "高清",
            "fileTypeId": "h264_800k_flv",
            "liveStream": "",
            "streamSource": 1,
            "url": "http://172.16.146.19/live/356d4576424c4d4c8b2cd37921213044?fmt=h264_800k_flv"
          }
        ]
      }
    ]
  },
  "playerCodeList": [
```

```

{
  "name": "默认播放器",
  "defaultFlag": "1",
  "playerCode": "<![CDATA[<link href=\"http://172.16.146.53//cswd/media/playerJson//styles/vms_style.css\" rel=\"stylesheet\" type=\"text/css\"><div class=\"vms-player playerWrapper\"><div class=\"vms-player-live\"><div class=\"vms-player-panel\"><div id=\"livePanel2\"></div></div></div><div class=\"vms-player-main\" id=\"flashdiv\"><div id=\"5e57a83f6216400991ac1c80203bfa7d\"></div></div></div><script type=\"text/javascript\">var vmsPlayer_callback = function () {createLivePlayer(\"356d4576424c4d4c8b2cd37921213044\", \"@WIDTH@\", \"@HEIGHT@\");}</script><script type=\"text/javascript\" src=\"http://172.16.146.53//cswd/media/playerJson/liveChannel/5e57a83f6216400991ac1c80203bfa7d.js\"></script>]]>",
  "staticFilePaths": {
    "playlistJS": "http://172.16.146.53/cswd//media/playerJson/dist/playerjs-module/vms-playlist_right_v1.0.js",
    "playlistCSS": "http://172.16.146.53/cswd//media/playerJson/styles/playlist_style_right_v1.0.css",
    "playJS": "http://172.16.146.53/cswd//media/playerJson/newliveChannel/5e57a83f6216400991ac1c80203bfa7d.js",
    "H5Soplayer": "http://172.16.146.53/cswd//media/html5/player/dist/sojs-module/H5Soplayer.js"
  }
},
{
  "name": "test",
  "defaultFlag": "0",
  "playerCode": "<![CDATA[<link href=\"http://172.16.146.53//cswd/media/playerJson//styles/vms_style.css\" rel=\"stylesheet\" type=\"text/css\"><div class=\"vms-player playerWrapper\"><div class=\"vms-player-live\"><div class=\"vms-player-panel\"><div id=\"livePanel2\"></div></div></div><div class=\"vms-player-main\" id=\"flashdiv\"><div id=\"a1ad4d3865c6491bb5ee4423b83ee392\"></div></div></div><script type=\"text/javascript\">var vmsPlayer_callback = function () {createLivePlayer(\"356d4576424c4d4c8b2cd37921213044\", \"@WIDTH@\", \"@HEIGHT@\");}</script><script type=\"text/javascript\" src=\"http://172.16.146.53//cswd/media/playerJson/liveChannel/a1ad4d3865c6491bb5ee4423b83ee392.js\"></script>]]>",
  "staticFilePaths": {
    "playlistJS": "http://172.16.146.53/cswd//media/playerJson/dist/playerjs-module/vms-playlist_right_v1.0.js",
    "playlistCSS": "http://172.16.146.53/cswd//media/playerJson/styles/playlist_style_right_v1.0.css",
    "playJS": "http://172.16.146.53/cswd//media/playerJson/newliveChannel/a1ad4d3865c6491bb5ee4423b83ee392.js",
    "H5Soplayer": "http://172.16.146.53/cswd//media/html5/player/dist/sojs-module/H5Soplayer.js"
  }
}
]

```

```
}
1
```

返回的参数说明:

为了方便于第三方解析，此处的格式和直播列表格式一致，只是内部的数组长度为 1

参数	说明
id	直播频道 ID
name	直播频道的名称
channelType	枚举值：8（电视直播）15（广播直播）16（虚拟直播） 17（设备直播） 18（纳加直播）
liveAddress	直播的播放地址信息，包括了多个终端和多码率的信息 liveTypes: 表示终端类型其中的 type 表示终端类型，比如是 pc, android 还是 ios。 liveTypes->formats 表示该终端类型下的播放地址，多码率的可以有多组 liveTypes->formats->title : formats 下的 title 属性表示这个码率对应的是高清还是标清 liveTypes->formats->url: 表示该码率下的播放地址 liveTypes->formats->streamSource:标示码流来源 枚举值：1 标示来自索贝流媒体 其 URL 为相对 URL 地址 枚举值：2 标示来自第三方流媒体，其 URL 为完整 HTTP 地址
currentName	当前正在播放的直播节目名称。
playerCode	直播频道的播放器代码。把这段代码可以嵌入到自己的网页中，或者是嵌入到文稿编辑的界面中发布出去。
summary	直播频道的简介
playerCodeList	name: 播放器名称 defaultFlag: 是否是默认播放器 1 标示是,0 标示否 playerCode: 播放器代码 staticFilePaths: 播放器静态资源文件 playlistJS: 节目单 js 资源

	<p>playlistCSS: 节目单 CSS 资源</p> <p>playJS: 播放模板 js 资源</p> <p>H5Soplayer: H5 播放器资源</p>
--	--

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

4.7 获取直播节目单信息

4.7.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 **key**，否则没有访问接口的权限。用户通过传入直播频道 **id**、查询时间来获取到对应的该节目单的信息，返回的结果包含节目的开始与结束时间、标题、描述等信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：**method= getChannelActList**

4.7.2 应用示例接口说明

➤ **调用获取某个直播频道节目单信息接口应用举例：**

本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（**method** 传入 **getChannelActList** 表示该示例调用的是 **getChannelActList** 这个方法，也就是调获取某个直播频道节目单信息的接口）

需要传入以下参数：

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌。
channelId	查询的直播频道 id
type	获取直播频道直播流的类型 默认为获取 PC 枚举值：PC、IOS、ANDROID
startDate	选择查询节目单的开始时间 格式 yyyy-MM-dd
endDate	选择查询节目单的结束时间 格式 yyyy-MM-dd

返回的参数格式：

JSON 格式：

```
[
  {
```

```

    "date": "2013-01-29",
    "playBill": [
      {
        "name": "今日新闻坊",
        "description": "今日新闻坊",
        "length": "00:21:00",
        "id": 785,
        "startTime": "2013-01-29 06:00:00",
        "endTime": "2013-01-29 06:21:00",
        "p2p": "&svr=glbdemo.sobeycache.com:15476&svt=1",
        "url": "mr://j:[{'title':'标清
', 'url': 'http://vod.sobeycache.com/live/7dbelca7a8ee4d029fd6c95cb69879a3?fmt=h264
_800k_flv'}, {'title':'高清
', 'url': 'http://vod.sobeycache.com/live/7dbelca7a8ee4d029fd6c95cb69879a3?fmt=h264
_1200k_flv'}]]",
        "channelId": "7dbelca7a8ee4d029fd6c95cb69879a3"
      },
      {
        "name": "天气预报",
        "description": "天气预报",
        "length": "00:05:00",
        "id": 810,
        "startTime": "2013-01-29 19:51:00",
        "endTime": "2013-01-29 19:56:00",
        "url": "mr://j:[{'title':'标清
', 'url': 'http://vod.sobeycache.com/live/7dbelca7a8ee4d029fd6c95cb69879a3?fmt=h264
_800k_flv'}, {'title':'高清
', 'url': 'http://vod.sobeycache.com/live/7dbelca7a8ee4d029fd6c95cb69879a3?fmt=h264
_1200k_flv'}]]",
        "channelId": "7dbelca7a8ee4d029fd6c95cb69879a3"
      }
    ]
  },
  {
    "date": "2013-01-30",
    "playBill": [
      {
        "name": "d",
        "description": "d",
        "length": "00:00:11",
        "id": 863,
        "startTime": "2013-01-30 11:38:49",
        "endTime": "2013-01-30 11:39:00",
        "p2p": "&svr=glbdemo.sobeycache.com:15476&svt=1",

```



```

        "url": "mr://j:[{'title':'标清
', 'url': 'http://vod.sobeycache.com/live/7dbelca7a8ee4d029fd6c95cb69879a3?fmt=h264
_800k_flv'}, {'title':'高清
', 'url': 'http://vod.sobeycache.com/live/7dbelca7a8ee4d029fd6c95cb69879a3?fmt=h264
_1200k_flv'}]]",
        "channelId": "7dbelca7a8ee4d029fd6c95cb69879a3"
    },
    {
        "name": "风动旖动",
        "description": "风动旖动",
        "length": "00:00:60",
        "id": 864,
        "startTime": "2013-01-30 11:39:00",
        "endTime": "2013-01-30 11:40:00",
        "url": "mr://j:[{'title':'标清
', 'url': 'http://vod.sobeycache.com/live/7dbelca7a8ee4d029fd6c95cb69879a3?fmt=h264
_800k_flv'}, {'title':'高清
', 'url': 'http://vod.sobeycache.com/live/7dbelca7a8ee4d029fd6c95cb69879a3?fmt=h264
_1200k_flv'}]]",
        "channelId": "7dbelca7a8ee4d029fd6c95cb69879a3"
    }
]
}
]

```

返回的参数说明：

参数	说明
name	节目名称
date	节目单时间
playBill	date 当天的节目单
description	节目描述
length	节目时长
id	节目 id
startTime	节目开始时间
endTime	节目结束时间
url	节目播放地址
channelId	直播频道 id

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

4.8 获取电视剧列表信息

4.8.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 key，否则没有访问接口的权限。用户可以根据自己需要通过传入站点名称或者是栏目全路径，分页信息和关键字等参数，来获取到自己所需要的电视剧剧集节目。返回的结果电视剧名称，所属栏目，电视剧关键帧，发布状态，剧集信息，电视剧 ID 等信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：`method= getSeriesList`

4.8.2 应用示例

➤ 调用获取电视剧列表信息的接口应用举例：

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 getSeriesList 表示该示例调用的是 getSeriesList 这个方法，也就是调用获取电视剧列表信息的接口）

需要传入以下参数：

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌。
catalogStyle	<p>栏目路径选择传参数的方式 默认值为 0</p> <p>枚举值：1 和 0</p> <p>使用 ID 查询时 catalogStyle 值为 1</p> <p>使用中文全路径查询时 catalogStyle 值为 0</p>
catalogPath	<p>站点名称或者是栏目名称，通过传入这个参数来获取某个站点或者是某个栏目下的电视剧信息，如果只传入的站点名称，则返回该站点下的电视剧信息，如果传入的是站点下的某个栏目，则返回该栏目下的电视剧信息</p> <p>（1）当 catalogStyle 值为 0，则格式为：</p> <p>测试站点@重庆卫视</p> <p>注意：如果传入的某个栏目，必须要按照栏目全路径的方式传入，比如要想获取重庆卫视下的新闻栏目下的电视剧</p>

	<p>列表那么传入的就应该为：测试站点@重庆卫视@新闻</p> <p>（2）当 catalogStyle 值为 1，则格式为：</p> <p>catalogPath 则为全路径叶子节点的 id 值，例如：查询 测试站点@重庆卫视@新闻下面的视频列表的时候，则 catalogPath 值为 新闻 栏目的 ID 值；如果查询全站的电视剧，则 catalogPath 值为 0</p>
getAllData	<p>枚举值 1 和 0 默认值 0</p> <p>1：表示返回该栏目下的所有视频信息，包括子栏目和子栏目的子栏目下的所有视频。</p> <p>0：表示只返回该栏目下的一级子栏目下的视频信息</p> <p>比如：假如有这样一个栏目层级</p> <p>测试站点</p> <p>---- 重庆卫视</p> <p>---- 新闻</p> <p>---- 娱乐新闻</p> <p>---- 图片</p> <p>----山东卫视</p> <p>如果传入的 CatalogPath 为：测试站点@重庆卫视</p> <p>当 getAllData 为 1 时返回：新闻，图片，娱乐新闻这三个栏目下的点播视频信息。</p> <p>当 getAllData 为 0 时返回：新闻，图片这两个栏目下的点播视频信息</p>
pageSize	<p>当获取数据需要以分页的方式来显示的话。pageSize 作为分页中的每页显示多少条，如果当此字段为空时，pageSize 默认为 10</p>
pageNum	<p>当获取数据需要以分页的方式来显示的话。pageNum 作为分页中的第几页，如果当此字段为空时，pageNum 默认为 1</p>
startTime	<p>用作查询时间段用的起始时间</p>

	格式为 yyyy-MM-dd HH:mm:ss
endTime	用作查询时间段用的结束时间 格式为 yyyy-MM-dd HH:mm:ss
keywords	用作查询用的关键字，可以为空
sortField	用于返回视频数据排序的排序字段
sort	用于返回视频数据排序的方式

返回的参数格式：

JSON 格式：

```
{
  "total": "2",
  "sort": "DESC",
  "sortField": "PublishDate",
  "pageTotal": "1",
  "pageSize": "10",
  "pageNum": "1",
  "video": [
    {
      "catalogId": "3",
      "createTime": "2015-07-17 11:00:34",
      "description": "",
      "director": "",
      "id": "1ed88c21ce9b4da0bf1b7c4338a8978e",
      "imagePath": "http://113.142.30.107:81/ms3.test/",
      "mainPlayer": "",
      "producingArea": "",
      "publishedTime": "2015-07-17 11:00:37",
      "seriesSummary": "",
      "seriesTotal": "5",
      "shootTime": "",
      "status": "1",
      "title": "电视剧之夺宝奇兵",
      "playerCodeList": [
        {
          "name": "默认播放器",
          "defaultFlag": "1",
          "playerCode": "
          "<![CDATA[<div
id=\"44bbe9f2abe24d2bbfef453082ce7205\"></div><script
type=\"text/javascript\">var
vmsPlayer_callback
=
function
()
{createPlayer(\"multi://sid:1ed88c21ce9b4da0bf1b7c4338a8978e\", \"%3Fguid%3D1ed88c21ce9b4d
          "
        }
      ]
    }
  ]
}
```

```

a0bf1b7c4338a8978e%26siteId%3D822\", \"2015-07-17\", \"@WIDTH@\", \"@HEIGHT@\");</script
><script
                                type=\"text/javascript\"
src=\"http://113.142.30.107/ms3.test/media/playerJson/vod/44bbe9f2abe24d2bbfef453082ce7205.
js\"></script>]]>\"
        },
        {
            \"name\": \"test\",
            \"defaultFlag\": \"0\",
            \"playerCode\": \"<![CDATA[<div
id=\"a9a1cba6652b4d619362d5dfc11c86fe\"></div><script
                                type=\"text/javascript\">var
vmsPlayer_callback
                                =
                                function
                                ()
{createPlayer(\"multi://sid:1ed88c21ce9b4da0bf1b7c4338a8978e\", \"%3Fguid%3D1ed88c21ce9b4d
a0bf1b7c4338a8978e%26siteId%3D822\", \"2015-07-17\", \"@WIDTH@\", \"@HEIGHT@\");</script
><script
                                type=\"text/javascript\"
src=\"http://113.142.30.107/ms3.test/media/playerJson/vod/a9a1cba6652b4d619362d5dfc11c86fe.
js\"></script>]]>\"
        }
    ]
},
{
    \"catalogId\": \"3\",
    \"createTime\": \"2015-06-30 14:40:43\",
    \"description\": \"\",
    \"director\": \"\",
    \"id\": \"02a4526df76e47ce8a9db06a0a2fbd14\",
    \"imagePath\": \"http://113.142.30.107:81/ms3.test/\",
    \"mainPlayer\": \"\",
    \"producingArea\": \"\",
    \"publishedTime\": \"2015-06-30 15:21:21\",
    \"seriesSummary\": \"\",
    \"seriesTotal\": \"45\",
    \"shootTime\": \"\",
    \"status\": \"1\",
    \"title\": \"电视剧熊出没\"
}
]
}

```

返回的参数说明:

参数	说明
total	获取视频的总条数
pageTotal	视频通过传入的分页参数统计出来的分页总数
pageSize	每页显示的条数

pageNum	当前处于第几页
sortField	排序的字段名称
sort	排序的方式
title	电视剧的标题
seriesTotal	电视剧总集数
shootTime	电视剧拍摄时间
producingArea	电视剧产地
mainPlayer	电视剧主演
director	电视剧导演
catalogId	该电视剧所属的栏目的 ID
id	视频的 GUID （是一个 32 位的唯一标识）
createTime	电视剧的创建时间。
publishedTime	电视剧的发布时间
imagePath	该电视剧的关键帧图片
playerCodeList	该节目的多个播放器代码 playerCodeList包含了多个player，其中player节点中包含了播放器的名称,播放器代码，是否是默认播放器等信息 playerName:播放器名称 playerCode:播放器代码 defaultFlag:是否是默认播放器
status	发布的状态。枚举值 1 和 0 1：表示已发布 0：表示未发布

4.9 获取某个电视剧信息

4.9.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 key，否则没有访问接口的权限。用户传入电视剧 id 来查询电视剧的信息。返回的结果电视剧名称，所属栏目，电视剧关键帧，发布状态，剧集信息，电视剧 ID 等信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：`method= getSeriesById`

4.9.2 应用示例

➤ 调用获取某个直播频道信息接口应用举例：

本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 `getSeriesById` 表示该示例调用的是 `getSeriesById` 这个方法，也就是调获取某个直播频道信息的接口）

需要传入以下参数：

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌。
seriesId	查询的电视剧 id

返回的参数格式：

JSON 格式：

```
{
  "total": "1",
  "video": [
    {
      "catalogId": "3",
      "createTime": "2015-07-17 11:00:34",
      "description": "",
      "director": "",
      "id": "1ed88c21ce9b4da0bf1b7c4338a8978e",
      "imagePath": "http://113.142.30.107:81/ms3.test/",
      "mainPlayer": "",
      "producingArea": "",
      "publishedTime": "2015-07-17 11:00:37",
      "seriesSummary": "",
      "seriesTotal": "5",
      "shootTime": "",
      "status": "1",
      "title": "电视剧熊出没",
      "playerCodeList": [
        {
          "name": "默认播放器",
          "defaultFlag": "1",
          "playerCode":
            "<![CDATA[<div
id=\"44bbe9f2abe24d2bbfef453082ce7205\"></div><script
type=\"text/javascript\">var
```

```

vmsPlayer_callback = function ()
{createPlayer(\"multi://sid:1ed88c21ce9b4da0bf1b7c4338a8978e\", \"%3Fguid%3D1ed88c21ce9b4d
a0bf1b7c4338a8978e%26siteId%3D822\", \"2015-07-17\", \"@WIDTH@\", \"@HEIGHT@\");</script
><script type=\"text/javascript\"
src=\"http://113.142.30.107/ms3.test/media/playerJson/vod/44bbe9f2abe24d2bbfef453082ce7205.
js\"></script>]]>\"
    },
    {
        \"name\": \"test\",
        \"defaultFlag\": \"0\",
        \"playerCode\": \"<![CDATA[<div
id=\"a9a1cba6652b4d619362d5dfc11c86fe\"></div><script type=\"text/javascript\">var
vmsPlayer_callback = function ()
{createPlayer(\"multi://sid:1ed88c21ce9b4da0bf1b7c4338a8978e\", \"%3Fguid%3D1ed88c21ce9b4d
a0bf1b7c4338a8978e%26siteId%3D822\", \"2015-07-17\", \"@WIDTH@\", \"@HEIGHT@\");</script
><script type=\"text/javascript\"
src=\"http://113.142.30.107/ms3.test/media/playerJson/vod/a9a1cba6652b4d619362d5dfc11c86fe.
js\"></script>]]>\"
    }
    ]
    }
    ]
}

```

返回的参数说明：

参数	说明
total	获取视频的总条数
title	电视剧的标题
seriesTotal	电视剧总集数
shootTime	电视剧拍摄时间
producingArea	电视剧产地
mainPlayer	电视剧主演
director	电视剧导演
catalogId	该电视剧所属的栏目的 ID
id	电视剧的 GUID （是一个 32 位的唯一标识）
createTime	电视剧的创建时间。
publishedTime	电视剧的发布时间

imagePath	该电视剧的关键帧图片
playerCodeList	<p>该节目的多个播放器代码</p> <p>playerCodeList包含了多个player，其中player节点中包含了播放器的名称,播放器代码，是否是默认播放器等信息</p> <p>playerName:播放器名称</p> <p>playerCode:播放器代码</p> <p>defaultFlag:是否是默认播放器</p>
status	<p>发布的状态。枚举值 1 和 0</p> <p>1：表示已发布</p> <p>0：表示未发布</p>

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

4.10 获取某个电视剧的集数列表

4.10.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的验证码作为参数中的一部分，作为调用其他接口的 key，否则没有访问接口的权限。用户可以根据自己需要通过传入电视剧 ID，分页信息和排序等参数，来获取到自己所需的电视剧下的视频。返回的结果包含视频的播放地址，播放代码，视频名称，所属栏目，视频关键帧，发布状态，视频 ID 等信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：`method= getSeriesInfo`

4.10.2 应用示例

➤ 调用获取具体某个电视剧下的视频接口应用举例：

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 getSeriesInfo 表示该示例调用的是 getSeriesInfo 这个方法，也就是调用获取具体某个电视剧下的视频接口）

需要传入以下参数：

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌。
seriesId	电视剧外壳的唯一标示符 Id

pageSize	当获取数据需要以分页的方式来显示的话。pageSize 作为分页中的每页显示多少条, 如果当此字段为空时, pageSize 默认为 10
pageNum	当获取数据需要以分页的方式来显示的话。pageNum 作为分页中的第几页, 如果当此字段为空时, pageNum 默认为 1
sortField	用于返回视频数据排序的排序字段
sort	用于返回视频数据排序的方式

返回的参数格式:

JSON 格式:

```
{
  "total": "2",
  "pageNum": "1",
  "pageSize": "10",
  "pageTotal": "1",
  "sort": "ASC",
  "sortField": "part",
  "video": [
    {
      "id": "e0b80418ec624651b49fe1f2434a64c8",
      "title": "1 熊出没之夺宝熊兵_超清",
      "tag": "",
      "programLength": "625",
      "createTime": "2015-07-17 09:35:11.0",
      "part": "1",
      "playerCodeList": [
        {
          "name": "默认播放器",
          "defaultFlag": "1",
          "playerCode":
            "<![CDATA[<div
id=\"44bbe9f2abe24d2bbfef453082ce7205\"></div><script
type=\"text/javascript\">var
vmsPlayer_callback
=
function
()
{createPlayer(\"multi://sid:1ed88c21ce9b4da0bf1b7c4338a8978e\", \"%3Fguid%3D1ed88c21ce9b4d
a0bf1b7c4338a8978e%26siteId%3D822\", \"2015-07-17\", \"@WIDTH@\", \"@HEIGHT@\");}</script
><script
type=\"text/javascript\"
src=\"http://113.142.30.107/ms3.test/media/playerJson/vod/44bbe9f2abe24d2bbfef453082ce7205.
js\"></script>]]>"
        }
      ]
    }
  ],
}
```

```

        {
            "name": "test",
            "defaultFlag": "0",
            "playerCode": "![CDATA[<div
id=\"a9a1cba6652b4d619362d5dfc11c86fe\"></div><script
type=\"text/javascript\">var
vmsPlayer_callback = function ()
{createPlayer(\"multi://sid:1ed88c21ce9b4da0bf1b7c4338a8978e\", \"%3Fguid%3D1ed88c21ce9b4d
a0bf1b7c4338a8978e%26siteId%3D822\", \"2015-07-17\", \"@WIDTH@\", \"@HEIGHT@\");}</script
><script
type=\"text/javascript\"
src=\"http://113.142.30.107/ms3.test/media/playerJson/vod/a9a1cba6652b4d619362d5dfc11c86fe.
js\"></script>]]>"
        }
    ],
    "vodAddress": {
        "duration": 625,
        "title": "1 熊出没之夺宝熊兵_超清",
        "host": "http://mdntest.sobeycache.com/ms3.test/vod/",
        "sourceType": "mp4",
        "formats": [
            "流畅"
        ],
        "clips": [
            {
                "index": "001",
                "duration": "625",
                "urls": [
                    "2015/07/17/e0b80418ec624651b49fe1f2434a64c8/h264_450k_flv.flv"
                ]
            }
        ]
    },
    "subPath": "ms3.test/vod/",
    "isSuccess": "1",
    "message": "播出 URL 信息获取成功!"
},
    "tsAddress": {
        "duration": 625,
        "title": "1 熊出没之夺宝熊兵_超清",
        "host": "http://mdntest.sobeycache.com/ms3.test/vod/",
        "formats": [
            "流畅"
        ],
        "clips": [
            {

```

```

        "index": "001",
        "duration": "625",
        "urls": [
            "/2015/07/17/e0b80418ec624651b49fe1f2434a64c8/h264_300k_ts.m3u8"
        ]
    },
    ],
    "subPath": "ms3.test/vod/",
    "isSuccess": "1",
    "message": "播出 URL 信息获取成功!"
},
"mp4Address": {
    "duration": 625,
    "title": "1 熊出没之夺宝熊兵_超清",
    "host": "http://mdntest.sobeycache.com/ms3.test/vod/",
    "formats": [
        "流畅"
    ],
    "clips": [
        {
            "index": "001",
            "duration": "625",
            "urls": [
                "2015/07/17/e0b80418ec624651b49fe1f2434a64c8/h264_300k_mp4.mp4"
            ]
        }
    ],
    ],
    "subPath": "ms3.test/vod/",
    "isSuccess": "1",
    "message": "播出 URL 信息获取成功!"
},
"status": "1",
"publishedTime": "2015-07-17 10:59:44.0",
"catalogId": "9",
"imagePath":
"http://113.142.30.107:81/ms3.test/vms/2015/07/17/e0b80418ec624651b49fe1f2434a64c8/suffix_
width_height_15638.jpg"
},
{
    "id": "49377fa4b0834a1ca8fa64a3995679f6",
    "title": "2 熊出没之夺宝熊兵_超清",
    "tag": "",

```

```

        "programLength": "619",
        "createTime": "2015-07-17 09:33:26.0",
        "part": "2",
        "playerCodeList": [
            {
                "name": "默认播放器",
                "defaultFlag": "1",
                "playerCode":
                    "<![CDATA[<div
id=\"44bbe9f2abe24d2bbfef453082ce7205\"></div><script
type=\"text/javascript\">var
vmsPlayer_callback
=
function
()
{createPlayer(\"multi://sid:1ed88c21ce9b4da0bf1b7c4338a8978e\", \"%3Fguid%3D1ed88c21ce9b4d
a0bf1b7c4338a8978e%26siteId%3D822\", \"2015-07-17\", \"@WIDTH@\", \"@HEIGHT@\");}</script
><script
type=\"text/javascript\"
src=\"http://113.142.30.107/ms3.test/media/playerJson/vod/44bbe9f2abe24d2bbfef453082ce7205.
js\"></script>]]>"
            },
            {
                "name": "test",
                "defaultFlag": "0",
                "playerCode":
                    "<![CDATA[<div
id=\"a9a1cba6652b4d619362d5dfc11c86fe\"></div><script
type=\"text/javascript\">var
vmsPlayer_callback
=
function
()
{createPlayer(\"multi://sid:1ed88c21ce9b4da0bf1b7c4338a8978e\", \"%3Fguid%3D1ed88c21ce9b4d
a0bf1b7c4338a8978e%26siteId%3D822\", \"2015-07-17\", \"@WIDTH@\", \"@HEIGHT@\");}</script
><script
type=\"text/javascript\"
src=\"http://113.142.30.107/ms3.test/media/playerJson/vod/a9a1cba6652b4d619362d5dfc11c86fe.
js\"></script>]]>"
            }
        ],
        "vodAddress": {
            "duration": 619,
            "title": "2 熊出没之夺宝熊兵_超清",
            "host": "http://mdntest.sobeycache.com/ms3.test/vod/",
            "sourceType": "mp4",
            "formats": [
                "流畅"
            ],
            "clips": [
                {
                    "index": "001",
                    "duration": "619",
                    "urls": [
                        "2015/07/17/49377fa4b0834a1ca8fa64a3995679f6/h264_450k_mp4.mp4"
                    ]
                }
            ]
        }
    }
}

```

```

        ]
    }
},
"subPath": "ms3.test/vod/",
"isSuccess": "1",
"message": "播出 URL 信息获取成功!"
},
"tsAddress": {
    "duration": 619,
    "title": "2 熊出没之夺宝熊兵_超清",
    "host": "http://mdntest.sobeycache.com/ms3.test/vod/",
    "formats": [
        "流畅"
    ],
    "clips": [
        {
            "index": "001",
            "duration": "619",
            "urls": [
                "2015/07/17/49377fa4b0834a1ca8fa64a3995679f6/h264_300k_mp4.mp4"
            ]
        }
    ],
    "subPath": "ms3.test/vod/",
    "isSuccess": "1",
    "message": "播出 URL 信息获取成功!"
},
"mp4Address": {
    "duration": 619,
    "title": "2 熊出没之夺宝熊兵_超清",
    "host": "http://mdntest.sobeycache.com/ms3.test/vod/",
    "formats": [
        "流畅"
    ],
    "clips": [
        {
            "index": "001",
            "duration": "619",
            "urls": [
                "2015/07/17/49377fa4b0834a1ca8fa64a3995679f6/h264_300k_mp4.mp4"
            ]
        }
    ]
}

```

```

    ],
    "subPath": "ms3.test/vod/",
    "isSuccess": "1",
    "message": "播出 URL 信息获取成功!"
  },
  "status": "1",
  "publishedTime": "2015-07-17 10:59:41.0",
  "catalogId": "10",
  "imagePath":
"http://113.142.30.107:81/ms3.test/vms/2015/07/17/49377fa4b0834a1ca8fa64a3995679f6/suffix_
width_height_15484.jpg"
  }
]
}

```

返回的参数说明:

参数	说明
total	获取视频的总条数
sortField	排序的字段名称
sort	排序的方式
title	视频节目的节目名称
catalogId	该视频所属的栏目的 ID
catalogPath	该视频所属的栏目的全路径 比如 : catalogPath :测试站点@重庆卫视表示该节目位于测试站点下的重庆卫视的栏目下
id	视频的 GUID （是一个 32 位的唯一标识）
createTime	视频节目的创建时间。
publishedTime	视频节目的发布时间
imagePath	该视频节目的关键帧图片
status	发布的状态。枚举值 1 和 0 1: 表示已发布 0: 表示未发布
vodAddress mp4Address tsAddress	该节目的多码率分片播放地址 vodAddress -> isSuccess: 1: 获取的播出地址及可用

	<p>0: 解析播放URL出现问题, 或者因为转码没有完成未产生可用播出URL</p> <p>vodAddress ->message: 获取播出URL的返回信息</p> <p>vodAddress ->duration : 该节目的总时长</p> <p>vodAddress ->title : 视频的名称</p> <p>vodAddress ->host: 视频的播放前缀</p> <p>vodAddress ->clips: 包含了分片视频的信息</p> <p>vodAddress ->formats: 多中码率对应的码率标题 比如是高清, 标清, 还是流畅等</p> <p>vodAddress ->clips->duration: 该分片视频的时长</p> <p>vodAddress ->clips->title: 该分片视频的名称</p> <p>vodAddress ->clips->urls: 该分片的不同码率的播放地址 其中完整的播放地址为: host 的播放前缀 + clips->urls[i] 并且 urls 中的码率顺序对应着 formats 中的码率标题的顺序</p> <p>比如说: url 中的第一个播放地址为 450K, formats 中的第一个值为标清, 那么 450K 就对应到标清。</p>
playerCodeList	<p>该节目的多个播放器代码</p> <p>playerCodeList包含了多个player, 其中player节点中包含了播放器的名称,播放器代码, 是否是默认播放器等信息</p> <p>playerName:播放器名称</p> <p>playerCode:播放器代码</p> <p>defaultFlag:是否是默认播放器</p>
part	如果是一个剧集类型表示该节目是剧集中的第几集
subTitle	视频的副标题
description	视频描述
programLength	节目时长
playTime	节目首播时间
sourceFrom	来源系统 id 枚举值

	1: 拍客上传 2: Web 上传 3: Ftp 上传 4: 综合入库 5: 在线快编 6: 视频收录 7: 迁移入库 8: 主干入库 9: 素材转码
--	---

4.11 获取音频列表信息

4.11.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 **key**，否则没有访问接口的权限。用户可以根据自己需要通过传入站点名称或者是栏目全路径，分页信息和关键字等参数，来获取到自己所需要的音频节目。返回的结果包含音频的播放地址，播放代码，音频名称，所属栏目，发布状态，音频 ID 等信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：[method= getAudioList](#)

4.11.2 应用示例

➤ 调用获取音频列表信息接口应用举例：

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 [getAudioList](#) 表示该示例调用的是 [getAudioList](#) 这个方法，也就是调获取音频列表信息的接口）

需要传入以下参数：

注：蓝色为选填

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌。
catalogStyle	栏目路径选择传参数的方式 默认值为 0 枚举值：1 和 0 使用 ID 查询时 catalogStyle 值为 1

	使用中文全路径查询时 catalogStyle 值为 0
catalogPath	<p>站点名称或者是栏目名称， 通过传入这个参数来获取某个站点或者是某个栏目下的音频信息，如果只传入的站点名称，则返回该站点下的音频，如果传入的是站点下的某个栏目，则返回该栏目下的音频信息</p> <p>（1）当 catalogStyle 值为 0，则格式为：</p> <p>测试站点@重庆卫视</p> <p>注意：如果传入的某个栏目，必须要按照栏目全路径的方式传入，比如要想获取重庆卫视下的新闻栏目下的音频列表那么传入的就应该为：测试站点@重庆卫视@新闻</p> <p>（2）当 catalogStyle 值为 1，则格式为：</p> <p>catalogPath 则为全路径叶子节点的 id 值，例如：查询 测试站点@重庆卫视@新闻下面的音频列表的时候，则 catalogPath 值为 新闻 栏目的 ID 值；如果查询全站的音频，则 catalogPath 值为 0</p>
getAllData	<p>枚举值 1 和 0 默认值为 0</p> <p>1：表示返回该栏目下的所有音频信息，包括子栏目和子栏目的子栏目下的所有音频。</p> <p>0：表示只返回该栏目下的一级子栏目下的音频信息</p> <p>比如：假如有这样一个栏目层级</p> <p>测试站点</p> <p>---- 重庆卫视</p> <p>---- 新闻</p> <p>---- 娱乐新闻</p> <p>---- 图片</p> <p>---- 山东卫视</p> <p>如果传入的 CatalogPath 为：测试站点@重庆卫视</p> <p>当 getAllData 为 1 时返回：新闻，图片，娱乐新闻这三个栏目下的点播音频信息。</p>

	当 <code>getAllData</code> 为 0 时返回：新闻，图片这两个栏目下的点播音频信息
status	-2:全部状态-1:初稿状态 0:待发布 1:发布 2:转码中 3:转码失败 4:回收站 5:流转中 6:流转异常 7:正在上传 8:取消转码 默认值为 1
pageSize	当获取数据需要以分页的方式来显示的话。 <code>pageSize</code> 作为分页中的每页显示多少条。如果当此字段为空时， <code>pageSize</code> 默认为 10
pageNum	当获取数据需要以分页的方式来显示的话。 <code>pageNum</code> 作为分页中的第几页，如果当此字段为空时， <code>pageNum</code> 默认为 1
startTime	用来查询发布时间段用的开始时间 格式为 yyyy-MM-dd HH:mm:ss
endTime	用来查询发布时间段用的结束时间 格式为 yyyy-MM-dd HH:mm:ss
keywords	用作查询用的关键字，可以为空
sortField	用于返回音频数据排序的排序字段
sort	用于返回音频数据排序的方式

返回的参数格式：

JSON 格式：

```
{
  "total": "2",
  "sort": "DESC",
  "sortField": "PublishDate",
  "pageTotal": "1",
  "pageSize": "10",
  "pageNum": "1",
  "video": [
    {
      "author": "胡夏",
      "catalogId": "2",
      "catalogPath": "不要删除@默认音频栏目",
      "contentSourceId": "90334cc9693941f7995238eade70c789",

```

```

        "createTime": "2013-01-15 17:41:26",
        "fileName": "mp3_128k_mp3_001.mp3",
        "fileSize": "1 MB",
        "id": "1",
        "status": "1",
        "playerCodeList": [
            {
                "defaultFlag": "1",
                "playerCode": "<![CDATA[<script type=\"text/javascript\"
src=\"http://113.142.30.107/bysc/media/player/playerJson/b87872f326e043c592ed0813
8c5f4df2.js\"></script><script
type=\"text/javascript\">createAudioPlayer(\"http://vod.sobeycache.com/vod/2013/0
1/15/90334cc9693941f7995238eade70c789?fmt=mp3_128k_mp3&slice=001\", \"@VIDEOID@\")
</script><div id=\"b87872f326e043c592ed08138c5f4df2\"></div>]]>",
                "playerName": "音频默认播放器"
            },
            {
                "defaultFlag": "0",
                "playerCode": "<![CDATA[<script type=\"text/javascript\"
src=\"http://113.142.30.107/bysc/media/player/playerJson/ccf5e80c2bf34f50b06ee086
22ce417b.js\"></script><script
type=\"text/javascript\">createAudioPlayer(\"http://vod.sobeycache.com/vod/2013/0
1/15/90334cc9693941f7995238eade70c789?fmt=mp3_128k_mp3&slice=001\", \"@VIDEOID@\")
</script><div id=\"ccf5e80c2bf34f50b06ee08622ce417b\"></div>]]>",
                "playerName": "123"
            }
        ],
        "produceTime": "2013-01-21 14:58:28",
        "programLength": [

        ],
        "publishedTime": "2013-01-15 15:08:23",
        "suffix": "mp3",
        "tag": "那些年，我们一起追的女孩 - 回忆",
        "title": "那些年，我们一起追的女孩 - 回忆",
        "url":
"http://vod.sobeycache.com/vod/2013/01/15/90334cc9693941f7995238eade70c789?fmt=mp
3_128k_mp3&slice=001"
    },
    {
        "author": "山野",
        "catalogId": "2",
        "catalogPath": "不要删除@默认音频栏目",
        "contentSourceId": "d95048872acb4018825b51ef6691665c",
    }

```

```

        "createTime": "2013-01-16 14:24:17",
        "fileName": "mp3_128k_mp3_001.mp3",
        "fileSize": "4 MB",
        "id": "6",
        "status": "1",
        "playerCodeList": [
            {
                "defaultFlag": "1",
                "playerCode": "<![CDATA[<script type=\"text/javascript\"
src=\"http://113.142.30.107/bysc/media/player/playerJson/b87872f326e043c592ed0813
8c5f4df2.js\"></script><script
type=\"text/javascript\">createAudioPlayer(\"http://vod.sobeycache.com/vod/2013/0
1/16/d95048872acb4018825b51ef6691665c?fmt=mp3_128k_mp3&slice=001\", \"@VIDEOID@\")
</script><div id=\"b87872f326e043c592ed08138c5f4df2\"></div>]]>",
                "playerName": "音频默认播放器"
            },
            {
                "defaultFlag": "0",
                "playerCode": "<![CDATA[<script type=\"text/javascript\"
src=\"http://113.142.30.107/bysc/media/player/playerJson/ccf5e80c2bf34f50b06ee086
22ce417b.js\"></script><script
type=\"text/javascript\">createAudioPlayer(\"http://vod.sobeycache.com/vod/2013/0
1/16/d95048872acb4018825b51ef6691665c?fmt=mp3_128k_mp3&slice=001\", \"@VIDEOID@\")
</script><div id=\"ccf5e80c2bf34f50b06ee08622ce417b\"></div>]]>",
                "playerName": "123"
            }
        ],
        "produceTime": "2013-01-30 14:58:31",
        "programLength": [

        ],
        "publishedTime": "2013-01-23 15:08:29",
        "suffix": "mp3",
        "tag": "ddf",
        "title": "ddf",
        "url":
"http://vod.sobeycache.com/vod/2013/01/16/d95048872acb4018825b51ef6691665c?fmt=mp
3_128k_mp3&slice=001"
    }
]
}

```

返回的参数说明：

参数	说明
----	----

total	获取音频的总条数
pageTotal	音频通过传入的分页参数统计出来的分页总数
pageSize	每页显示的条数
pageNum	当前处于第几页
sortField	排序的字段名称
sort	排序的方式
title	音频节目的节目名称
tag	音频的标签
catalogId	该音频所属的栏目的 ID
id	音频的 ID（数字类型）
contentSourceId	音频的 GUID（是一个 32 位的唯一标识）
createTime	音频节目的创建时间。
publishedTime	音频节目的发布时间
suffix	音频格式
programLength	音频时长
fileSize	音频大小
author	录音作者
status	-1:初稿状态 0:待发布 1:发布 2:转码中 3:转码失败 4:回收站 5:流转中 6:流转异常 7:正在上传 8:取消转码
url	音频播放地址
playerCodeList	<p>该节目的多个播放器代码</p> <p>playerCodeList包含了多个player，其中player节点中包含了播放器的名称,播放器代码，是否是默认播放器等信息</p> <p>playerName:播放器名称</p> <p>playerCode:播放器代码</p> <p>defaultFlag:是否是默认播放器</p>

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

4.12 获取某个音频信息

4.12.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 key，否则没有访问接口的权限。用户通过传入音频 id 来获取到对应的该音频的信息，返回的结果包含音频的播放地址，播放代码，音频名称，所属栏目，发布状态，音频 ID 等信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：`method= getAudioById`

4.12.2 应用示例

➤ 调用获取某个直播频道信息接口应用举例：

本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 getAudioById 表示该示例调用的是 getAudioById 这个方法，也就是调获取某个音频信息的接口）

需要传入以下参数：

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌。
audioId	查询的音频 id

返回的参数格式：

JSON 格式：

```
{
  "total": "1",
  "video": {
    "author": "胡夏",
    "catalogId": "2",
    "catalogPath": "不要删除@默认音频栏目",
    "contentSourceId": "90334cc9693941f7995238eade70c789",
    "createTime": "2013-01-15 17:41:26",
    "fileName": "mp3_128k_mp3_001.mp3",
    "fileSize": "1 MB",
    "id": "1",
    "status": "1",
    "playerCodeList": [
      {
        "defaultFlag": "1",
```

```

        "playerCode": "<![CDATA[<script type=\"text/javascript\"
src=\"http://113.142.30.107/bysc/media/player/playerJson/b87872f326e043c592ed08138c5f4df2.js
\"></script><script
type=\"text/javascript\">createAudioPlayer(\"http://vod.sobeycache.com/vod/2013/01/15/90334cc
9693941f7995238eade70c789?fmt=mp3_128k_mp3&slice=001\", \"@VIDEOID@\")</script><div
id=\"b87872f326e043c592ed08138c5f4df2\"></div>]]>",
        "playerName": "音频默认播放器"
    },
    {
        "defaultFlag": "0",
        "playerCode": "<![CDATA[<script type=\"text/javascript\"
src=\"http://113.142.30.107/bysc/media/player/playerJson/ccf5e80c2bf34f50b06ee08622ce417b.js
\"></script><script
type=\"text/javascript\">createAudioPlayer(\"http://vod.sobeycache.com/vod/2013/01/15/90334cc
9693941f7995238eade70c789?fmt=mp3_128k_mp3&slice=001\", \"@VIDEOID@\")</script><div
id=\"ccf5e80c2bf34f50b06ee08622ce417b\"></div>]]>",
        "playerName": "123"
    }
],
"produceTime": "2013-01-21 14:58:28",
"programLength": [

],
"publishedTime": "2013-01-15 15:08:23",
"suffix": "mp3",
"tag": "那些年，我们一起追的女孩 - 回忆",
"title": "那些年，我们一起追的女孩 - 回忆",
"url":
"http://vod.sobeycache.com/vod/2013/01/15/90334cc9693941f7995238eade70c789?fmt=mp3_12
8k_mp3&slice=001"
}
}

```

返回的参数说明：

参数	说明
total	获取音频的总条数
title	音频节目的节目名称
tag	音频的标签
catalogId	该音频所属的栏目的 ID
id	音频的 ID（数字类型）

contentSourceId	音频的 GUID （是一个 32 位的唯一标识）
createTime	音频节目的创建时间。
publishedTime	音频节目的发布时间
suffix	音频格式
programLength	音频时长
fileSize	音频大小
author	录音作者
status	-1:初稿状态 0:待发布 1:发布 2:转码中 3:转码失败 4:回收站 5:流转中 6:流转异常 7:正在上传 8:取消转码
url	音频播放地址
playerCodeList	该节目的多个播放器代码 playerCodeList包含了多个player，其中player节点中包含了播放器的名称,播放器代码，是否是默认播放器等信息 playerName:播放器名称 playerCode:播放器代码 defaultFlag:是否是默认播放器

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

4.13 删除视频、音频或电视剧信息

4.13.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的验证码作为参数中的一部分，作为调用其他接口的 key，否则没有访问接口的权限。该接口通过。通过传入需要删除的媒介（视频、音频或电视剧）的 id、媒介的类型以及删除的方式（0：彻底删除，1：删除至回收站）三个参数，删除数据库中的相关数据以及静态文件信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：`method= deleteDataByIds`

4.13.2 应用示例

➤ 调用获取某个直播频道的播放器代码的接口应用举例：

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 `deleteDataByIds` 表示该示例调用的是

`deleteDataByIds` 这个方法，也就是调用删除视频、音频或电视剧信息的接口）

需要传入以下参数：

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌
sourceIDs	要删除的媒介（视频、音频或电视剧的 id），当删除多个的时候用逗号隔开。例如：123,456,7
sourceType	要删除的媒介的类型(5:视频；6：音频；7：电视剧)
delType	删除方式（0：彻底删除；1：删除至回收站）

返回的参数格式：

JSON 格式：

```
{
  "status": "1",
  "message": "删除成功！"
}
```

返回的参数说明：

参数	说明
status	1:删除成功；0：删除失败
delMessage	删除操作返回的具体信息

4.14 获取站点的信息

4.14.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 key，否则没有访问接口的权限。用户可通过该接口获取站点的相关信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：`method= getSiteInfo`

4.14.2 应用示例

➤ 调用获取某个直播频道的播放器代码的接口应用举例：

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 `getSiteInfo` 表示该示例调用的是 `getSiteInfo` 这个方法，也就是调用获取站点信息的接口）

需要传入以下参数：

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌

返回的 JSON 格式：

```
{
  "siteId": "324",
  "name": "测试站点",
  "email": "sobey@qq.com",
  "alias": "sbnbyszd",
  "status": "1",
  "totalSpace": "150.0",
  "usedSpace": "0.0",
  "surplusSpace": "150.0",
  "expiredtime": "2013-07-05 17:12:32",
  "addTime": "2013-04-12 17:12:32"
}
```

返回的参数说明：

参数	说明
siteId	站点 Id
name	站点名称
email	用户名称
alias	用户名称简写
status	用户状态
totalSpace	用户最大空间
usedSpace	已用空间
surplusSpace	剩余空间
expiredtime	到期时间
addTime	创建时间

4.15 获取转码组的信息

4.15.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 key，否则没有访问接口的权限。用户可通过该接口获取转码组的相关信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：`method= getTranscode`

4.15.2 应用示例

➤ 调用获取某个站点的转码组的接口应用举例：

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 `getTranscode` 表示该示例调用的是

`getTranscode` 这个方法，也就是调用获取转码组的接口）

需要传入以下参数：

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌

JSON 格式：

```
[
  {
    "transgroupId": "1",
    "transgroupName": "默认转码组",
    "transgroupAlias": "默认转码组",
    "codes": [
      {
        "id": "1",
        "name": "超清",
        "alias": "1080P"
      },
      {
        "id": "2",
        "name": "高清",
        "alias": "720P"
      },
      {
        "id": "3",
        "name": "标清",
        "alias": "360P"
      },
      {
        "id": "4",
        "name": "流畅",
        "alias": "270P"
      }
    ]
  }
]
```

```
}
]
```

返回的参数说明：

参数	说明
Id	码率 id
name	码率名称
alias	用户名称简写
codes	码率
transgroupIdAlias	转码组描述
transgroupName	转码组名称
transgroupid	码率组 id

4.16 获取播放器列表信息

4.16.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的验证码作为参数中的一部分，作为调用其他接口的 **key**，否则没有访问接口的权限。用户可通过该接口获取各种类型播放器的相关信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：**method= getPlayerList**

4.16.2 应用示例

➤ 调用获取播放器列表的接口应用举例：

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 **getPlayerList** 表示该示例调用的是 **getPlayerList** 这个方法，也就是调用获取播放器列表的接口）

需要传入以下参数：

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌
name	根据播放器名称模糊匹配
description	根据播放器描述模糊匹配
defaultFlag	是否为默认播放器，1 为默认，0 为非默认。

	缺省获取设置所有播放器。
type	类型。5：点播；6：音频；7：电视剧； 8：电视直播；15：广播直播； 16：虚拟直播
pageSize	当获取数据需要以分页的方式来显示的话。pageSize 作为分页中的每页显示多少条，如果当此字段为空时，pageSize 默认为 10
pageNum	当获取数据需要以分页的方式来显示的话。pageNum 作为分页中的第几页，如果当此字段为空时，pageNum 默认为 1
sortField	用于返回播放器数据排序的排序字段
sort	用于返回播放器数据排序的方式

JSON 格式：

```
{
  "total": "1",
  "pageTotal": "0",
  "pageSize": "0",
  "pageNum": "0",
  "player": [
    {
      "defaultFlag": "1",
      "description": "系统默认播放器",
      "guid": "25842f280dcf4560938fcc12fc65c5b6",
      "name": "默认播放器",
      "skin": "0"
    }
  ]
}
```

返回的参数说明：

参数	说明
guid	播放器 id
name	播放器名称
description	播放器描述

defaultFlag	是否默认，1 为默认
type	类型
skin	皮肤

4.17 获取某个播放器信息

4.17.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 key，否则没有访问接口的权限。用户通过传入播放器 id 来获取到对应的该播放器的信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：`method= getPlayerById`

4.17.2 应用示例

➤ 调用获取播放器列表的接口应用举例：

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 `getPlayerById` 表示该示例调用的是 `getPlayerById` 这个方法，也就是调用获取播放器列表的接口）

需要传入以下参数：

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌
guid	所要获取的播放器 id，多个用,(英文逗号)隔开

JSON 格式：

```
{
  "total": "1",
  "pageTotal": "0",
  "pageSize": "0",
  "pageNum": "0",
  "player": [
    {
      "defaultFlag": "1",
      "description": "系统默认播放器",
      "guid": "25842f280dcf4560938fcc12fc65c5b6",
      "name": "默认播放器",
      "skin": "0",

```

```

    "type": "0"
  }
]
}

```

返回的参数说明：

参数	说明
guid	播放器 id
name	播放器名称
description	播放器描述
defaultFlag	是否默认，1 为默认
type	类型。
skin	皮肤

4.18 更新某个视频信息

4.18.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 **key**，否则没有访问接口的权限。用户通过传入视频 ID 及需要修改的视频字段参数来更新 VMS 里对应视频的信息，当需要修改视频抽帧图时，需使用 VMS 提供的图片上传插件，先将图片上传到存储上并获得该图片的地址，然后将地址传回给 VMS。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：**method= updateVideoById**

4.18.2 应用示例

➤ 调用获取播放器列表的接口应用举例：

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 **updateVideoById** 表示该示例调用的是 **updateVideoById** 这个方法，也就是调用获取播放器列表的接口）

需要传入以下参数：

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌
videoId	视频唯一标示 ID
title	需要修改的视频标题

description	需要修改的视频描述
keyframe	需要修改的视频抽帧图地址(如未修改，请传空字段)

返回说明：

Success 更新成功

Faille 更新失败

Other 其他提示

4.19 获取虚拟直播频道列表信息

4.19.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 **key**，否则没有访问接口的权限。用户通过调用 **getVirChannelList** 获取到该站点下的所有虚拟直播频道的信息，包括直播频道的名称，简介等信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：**method= getVirChannelList**

4.19.2 应用示例

➤ 调用获取直播频道列表信息接口应用举例：

本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 **getVirChannelList** 表示该示例调用的是 **getVirChannelList** 这个方法，也就是调获取虚拟直播频道列表信息的接口）

需要传入以下参数：

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌。

返回的参数格式：

JSON 格式：

```
[
  {
    "channelImgUrl":
"http://113.142.30.107:81/ms3.test/upload/Image/live/2015/07/01/bc86650a-1c88-46a6-a549-a74d62266c1d.jpg",
    "id": "1dc9f987ea094c35919ca23b78a4bec8",
    "name": "北京电视台",
```

```

        "playerCode": "<![CDATA[<link
href=\"http://113.142.30.107/ms3.test/media/playerJson//styles/vms_style.css\"
rel=\"stylesheet\" type=\"text/css\"><div class=\"vms-player playerWrapper\"><div
class=\"vms-player-live\"><div class=\"vms-player-panel\"><div
id=\"livePanel2\"></div></div></div><div class=\"vms-player-main\" id=\"flashdiv\"><div
id=\"6cfc4d3738cf4144afe6b7bb8a800de9\"></div></div></div><script
type=\"text/javascript\">var vmsPlayer_callback = function ()
{createLivePlayer(\"1dc9f987ea094c35919ca23b78a4bec8\", \"@WIDTH@\", \"@HEIGHT@\");};
</script><script type=\"text/javascript\"
src=\"http://113.142.30.107/ms3.test/media/playerJson/liveChannel/6cfc4d3738cf4144afe6b7
bb8a800de9.js\"></script>]]>\",
        "summary": "sdfsaf"
    },
    {
        "channellmgUrl":
"http://113.142.30.107:81/ms3.test/upload/Image/live/2015/07/03/430dc6e2-6278-4d6e-ac58
-0dd842119450.jpg",
        "id": "cfcc3d300b4849899604b15cda1f2217",
        "name": "奉贤电视台",
        "playerCode": "<![CDATA[<link
href=\"http://113.142.30.107/ms3.test/media/playerJson//styles/vms_style.css\"
rel=\"stylesheet\" type=\"text/css\"><div class=\"vms-player playerWrapper\"><div
class=\"vms-player-live\"><div class=\"vms-player-panel\"><div
id=\"livePanel2\"></div></div></div><div class=\"vms-player-main\" id=\"flashdiv\"><div
id=\"6cfc4d3738cf4144afe6b7bb8a800de9\"></div></div></div><script
type=\"text/javascript\">var vmsPlayer_callback = function ()
{createLivePlayer(\"cfcc3d300b4849899604b15cda1f2217\", \"@WIDTH@\", \"@HEIGHT@\");};
</script><script type=\"text/javascript\"
src=\"http://113.142.30.107/ms3.test/media/playerJson/liveChannel/6cfc4d3738cf4144afe6b7
bb8a800de9.js\"></script>]]>\",
        "summary": []
    }
]

```

返回的参数说明:

参数	说明
id	虚拟直播频道 ID
name	虚拟直播频道的名称
channellmgUrl	虚拟直播平台的封面图
playerCode	虚拟直播频道的播放器代码。把这段代码可以嵌入到自己的网页中，或者是嵌入到文稿编辑的界面中发布出去。

summary

虚拟直播频道的简介

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

4.20 获取某个虚拟直播频道信息

4.20.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 **key**，否则没有访问接口的权限。用户通过传入直播频道来获取到对应的该直播频道的信息，包括直播频道的名称，简介等信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：**method= getVirChannelById**

4.20.2 应用示例

➤ **调用获取直播频道列表信息接口应用举例：**

本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 getVirChannelById 表示该示例调用的是 getVirChannelById 这个方法，也就是调获取某个直播频道信息的接口）

需要传入以下参数：

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌。
virChannelId	查询的虚拟直播频道 id

返回的参数格式：

JSON 格式：

```
[
{
  "id": "dcbc24fa7ad54e518d0a69820124d0e5",
  "name": "上海卫视",
  "channelImgUrl":
"http://113.142.30.107:81/sbnbyszd/upload/Image/default/2013/04/16/a253c4c1-74c4-4fc4-bd41-69f06f6b2e22.jpg",
  "playerCode": "<link
href='http://113.142.30.196:8060/cszd/media/player/styles/vms_style.css'
rel='stylesheet' type='text/css'><script type='text/javascript'
src='http://113.142.30.196:8060/cszd/media/player/79e4037fbadf43578de5aa3045a9bdb
6.js'></script><script
type='text/javascript'>createLivePlayer('106','PC');</script><div
```

```
class="playerWrapper" style="position:relative; clear:both; width:950px; margin:0 auto; "><div style="float:right; width:300px; _width:298px; "><div style="width:300px; _width:298px; height:485px; background-color:#272727; border:0; "><div id="livePanel2"></div></div></div><div style="float:left" id="flashdiv"><div id="79e4037fbadf43578de5aa3045a9bdb6"></div></div></div> ",
    "summary": "直播简介"
}
]
```

返回的参数说明：

为了方便于第三方解析，此处的格式和直播列表格式一致，只是内部的数组长度为 1

参数	说明
id	虚拟直播频道 ID
name	虚拟直播频道的名称
channellmgUrl	虚拟直播平台的封面图
playerCode	虚拟直播频道的播放器代码。把这段代码可以嵌入到自己的网页中，或者是嵌入到文稿编辑的界面中发布出去。
summary	虚拟直播频道的简介

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

4.21 获取虚拟直播节目单信息

4.21.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 key，否则没有访问接口的权限。用户通过传入直播频道 id、查询时间来获取到对应的该节目单的信息，返回的结果包含节目的开始与结束时间、标题、描述等信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：`method= getVirChannelActList`

4.21.2 应用示例

➤ 调用获取某个虚拟直播频道节目单信息接口应用举例：

本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 getVirChannelActList 表示该示例调用的是 getVirChannelActList 这个方法，也就是调获取某个虚拟直播频道节目单信息的接口）

需要传入以下参数：

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌。
virChannelId	查询的虚拟直播频道 id
startDate	选择查询节目单的开始时间 格式 yyyy-MM-dd
endDate	选择查询节目单的结束时间 格式 yyyy-MM-dd

返回的参数格式：

JSON 格式：

```
[
  {
    "date": "2013-01-29",
    "playBill": [
      {
        "name": "今日新闻坊",
        "length": "00:21:00",
        "id": 785,
        "startTime": "2013-01-29 06:00:00",
        "endTime": "2013-01-29 06:21:00"
      },
      {
        "name": "天气预报",
        "length": "00:05:00",
        "id": 810,
        "startTime": "2013-01-29 19:51:00",
        "endTime": "2013-01-29 19:56:00"
      }
    ]
  },
  {
    "date": "2013-01-30",
    "playBill": [
      {
        "name": "d",
        "length": "00:00:11",
        "id": 863,
        "startTime": "2013-01-30 11:38:49",
        "endTime": "2013-01-30 11:39:00"
      },
      {
```

```

        "name": "风动旖动",
        "length": "00:00:60",
        "id": 864,
        "startTime": "2013-01-30 11:39:00",
        "endTime": "2013-01-30 11:40:00"
    }
}
]

```

返回的参数说明：

参数	说明
name	节目名称
date	节目单时间
playBill	date 当天的节目单
length	节目时长
id	节目 id
startTime	节目开始时间
endTime	节目结束时间

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

4.22 获取所有租户信息

4.22.1 接口说明

接口简介：合作伙伴用户系统通过申请后把邮箱得到的认证码作为参数中的一部分，作为调用其他接口的 key，否则没有访问接口的权限。用户可以查看所有租户的基本信息。

接口地址：[http:// 113.142.30.108:8080/vms/APIServiceReceiver](http://113.142.30.108:8080/vms/APIServiceReceiver)

方法名称：[method= getAccountList](#)

4.22.2 应用示例

➤ 调用获取租户列表信息接口应用举例：

本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

示例：（method 传入 getAccountList 表示该示例调用的是 getAccountList 这个方法，也就是调获取租户信息的接口）

需要传入以下参数：

注：蓝色为选填

参数	说明
partnerToken	通过合作伙伴认证接口返回的 token 令牌。
status	<p>获取租户状态枚举类型 默认值为 1</p> <p>枚举值：1、0 和 -2</p> <p>获取租户状态为 1 的租户 status 值为 1</p> <p>获取租户状态为 0 的租户 status 值为 0</p> <p>获取所有租户时 status 值为-2</p>

返回的参数格式：JSON 格式

```
[
  {
    "ID": "887",
    "email": "yxy@gmail.com",
    "Name": "yszd",
    "Alias": "yszd",
    "Prop1": "1",
    "prop2": {
      "flvVideo": [
        {
          "name": "高清",
          "alias": "720p",
          "format": "h264_800k_flv"
        },
        {
          "name": "标清",
          "alias": "480p",
          "format": "h264_500k_flv"
        },
        {
          "name": "流畅",
          "alias": "360p",
          "format": "h264_300k_flv"
        }
      ],
      "audio": [
        {
          "name": "高品质",
          "alias": "192K",
          "format": "mp3_192k_mp3"
        },
        {
```

```
        "name": "标准",
        "alias": "128K",
        "format": "mp3_128k_mp3"
    },
    {
        "name": "流畅",
        "alias": "64K",
        "format": "mp3_64k_mp3"
    }
],
"tsVideo": [
    {
        "name": "高清",
        "alias": "720p",
        "format": "h264_800k_ts"
    },
    {
        "name": "标清",
        "alias": "480p",
        "format": "h264_500k_ts"
    },
    {
        "name": "流畅",
        "alias": "360p",
        "format": "h264_300k_ts"
    }
],
"mp4Video": [
    {
        "name": "高清",
        "alias": "720p",
        "format": "h264_800k_mp4"
    },
    {
        "name": "标清",
        "alias": "480p",
        "format": "h264_500k_mp4"
    },
    {
        "name": "流畅",
        "alias": "360p",
        "format": "h264_300k_mp4"
    }
]
```



```

    },
    "AddUser": "yxy@gmail.com",
    "AddTime": "2016-08-18 18:46:14.0",
    "ModifyUser": "yxy@gmail.com",
    "ModifyTime": "2016-08-18 18:46:14.0",
    "status": "0",
    "UseSecurityCard": "0",
    "linkMan": "",
    "qq": "916952444",
    "telephone": "",
    "mobilephone": "15209983322",
    "type": "1",
    "expiredTime": "2016-09-02 18:46:14.0",
    "space": "200",
    "flux": "300",
    "useDay": "15",
    "barrageFlag": "1",
    "seniorEdit": "1",
    "verifyCode": "",
    "tenantId": "test"
  }
]

```

返回参数说明：

参数	说明
ID	账户 ID 值，唯一标识
email	用户邮箱
Name	用户名称
Alias	用户英文简称
Prop1	保留字段
prop2	入库码率设置参数
AddUser	添加用户
AddTime	添加时间
ModifyUser	修改人
ModifyTime	修改时间
status	用户状态
UseSecurityCard	用户密保
linkMan	联系人

qq	
telephone	
mobilephone	
type	
expiredTime	用户过期时间
space	用户空间
flux	用户流量
useDay	
barrageFlag	弹幕
seniorEdit	
verifyCode	
tenantId	租户 tenantId

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

5 通知机制

5.1 视频更新通知机制

5.1.1 接口说明

接口简介：当 VMS 里面管理的视频发布或者被推送的时候，可以相关的信息主动通知下游 CMS 平台，让下游平台数据保持最新和最近修改

5.1.2 应用示例

➤ 接口 post 视频和站点等相关信息给下游 CMS 应用举例：

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

POST 传入 JSON 格式：

```
{
  "vodAddress": {
    "isSuccess": "1",
    "message": "播出 URL 信息获取成功!",
    "title": "生活大不同(17)",
    "duration": "15",
    "host": "http://172.16.144.18/vmstest/vod/",
    "subPath": "vmstest/vod/",
  }
}
```

```
"clips": [
  {
    "duration": "15",
    "title": "001",
    "urls": [
      "2015/01/07/403c0848e1f8427bb42bc922e99dc0bd/h264_800k_mp4.mp4",
      "2015/01/07/403c0848e1f8427bb42bc922e99dc0bd/h264_500k_mp4.mp4",
      "2015/01/07/403c0848e1f8427bb42bc922e99dc0bd/h264_300k_mp4.mp4"
    ]
  }
],
"formats": [
  "高清",
  "标清",
  "流畅"
],
"sourceType": "mp4",
"fileType": "mp4"
},
"tsAddress": {
  "title": "生活大不同(17)",
  "isSuccess": "1",
  "message": "播出 URL 信息获取成功!",
  "duration": "15",
  "subPath": "vmstest/vod/",
  "host": "http://172.16.144.18/vmstest/vod/",
  "clips": [
    {
      "duration": "15",
      "title": "001",
      "urls": [
        "2015/01/07/403c0848e1f8427bb42bc922e99dc0bd/h264_800k_mp4.mp4",
        "2015/01/07/403c0848e1f8427bb42bc922e99dc0bd/h264_500k_mp4.mp4",
        "2015/01/07/403c0848e1f8427bb42bc922e99dc0bd/h264_300k_mp4.mp4"
      ]
    }
  ],
  "formats": [
    "高清",
    "标清",
    "流畅"
  ],
  "sourceType": "mp4",
  "fileType": "m3u8"
```

```
},
"mp4Address": {
  "title": "生活大不同(17)",
  "isSuccess": "1",
  "message": "播出 URL 信息获取成功!",
  "duration": "15",
  "subPath": "vmstest/vod/",
  "host": "http://172.16.144.18/vmstest/vod/",
  "clips": [
    {
      "duration": "15",
      "title": "001",
      "urls": [
        "2015/01/07/403c0848e1f8427bb42bc922e99dc0bd/h264_800k_mp4.mp4",
        "2015/01/07/403c0848e1f8427bb42bc922e99dc0bd/h264_500k_mp4.mp4",
        "2015/01/07/403c0848e1f8427bb42bc922e99dc0bd/h264_300k_mp4.mp4"
      ]
    }
  ],
  "formats": [
    "高清",
    "标清",
    "流畅"
  ],
  "sourceType": "mp4",
  "fileType": "mp4"
},
"id": "403c0848e1f8427bb42bc922e99dc0bd",
"classifyType": 0,
"catalogId": 228,
"catalogTree": [
  {
    "catalogId": 218,
    "catalogName": "新建入库一级栏目",
    "treeLevel": 1
  },
  {
    "catalogId": 224,
    "catalogName": "新建入库二级栏目 1",
    "treeLevel": 2
  },
  {
    "catalogId": 225,
    "catalogName": "新建入库三级栏目 1",
```

```

        "treeLevel": 3
    },
    {
        "catalogId": 226,
        "catalogName": "新建入库四级栏目 1",
        "treeLevel": 4
    },
    {
        "catalogId": 227,
        "catalogName": "新建入库五级栏目 1",
        "treeLevel": 5
    },
    {
        "catalogId": 228,
        "catalogName": "新建入库六级栏目 1",
        "treeLevel": 6
    }
],
"createTime": "2015-01-07 17:57:28",
"description": "",
"imageList": [
    {
        "defaultKeyFrame": 0,
        "imagePath": "http://113.142.30.107:81/vmstest/upload/Image/static/2015/01/07/403c0848e1f8427bb42bc922e99dc0bd/defaultKeyFrame.jpg"
    },
    {
        "defaultKeyFrame": 1,
        "imagePath": "http://113.142.30.107:81/vmstest/upload/Image/static/2015/01/07/403c0848e1f8427bb42bc922e99dc0bd/JPG_JPG.jpg"
    }
],
"interfacesTypes": "",
"playerCodeList": [
    {
        "defaultFlag": "1",
        "playerCode": "<![CDATA[<div id=\"10f8963e159a4cb08db5748437476387\"></div><script type=\"text/javascript\">var vmsPlayer_callback = function () {createPlayer(\"video://vid:403c0848e1f8427bb42bc922e99dc0bd\", \"%3Fguid%3D403c0848e1f8427bb42bc922e99dc0bd%26siteId%3D560\", \"2015-01-07\", \"@WIDTH@\", \"@HEIGHT@\")};</script><script type=\"text/javascript\" src=\"http://113.142.30.107/vmstest/media/playerJson/vod/10f8963e159a4cb08db5748437476387.js\"></script>]]>",
        "playerName": "默认播放器"
    }
]

```

```

        "defaultFlag": "0",
        "playerCode": "<![CDATA[<div id=\"d777b89c27a849c4bbc5cd2730988a91\"></div><script ty
pe=\"text/javascript\">var vmsPlayer_callback = function () {createPlayer(\"video://vid:403c0848e1f8427bb
42bc922e99dc0bd\", \"%3Fguid%3D403c0848e1f8427bb42bc922e99dc0bd%26siteId%3D560\", \"2015-01-07\",
\"@WIDTH@\", \"@HEIGHT@\")};</script><script type=\"text/javascript\" src=\"http://113.142.30.107/vmste
st/media/playerJson/vod/d777b89c27a849c4bbc5cd2730988a91.js\"></script>]]>",
        "playerName": "new_4:3"
    },
    {
        "defaultFlag": "0",
        "playerCode": "<![CDATA[<div id=\"544d6f41cd244a39a58b11973d5838bf\"></div><script typ
e=\"text/javascript\">var vmsPlayer_callback = function () {createPlayer(\"video://vid:403c0848e1f8427bb4
2bc922e99dc0bd\", \"%3Fguid%3D403c0848e1f8427bb42bc922e99dc0bd%26siteId%3D560\", \"2015-01-07\", \"
@WIDTH@\", \"@HEIGHT@\")};</script><script type=\"text/javascript\" src=\"http://113.142.30.107/vmstes
t/media/playerJson/vod/544d6f41cd244a39a58b11973d5838bf.js\"></script>]]>",
        "playerName": "test"
    }
],
"programLength": 15,
"publishTime": "2015-01-07 17:57:30",
"status": 1,
"tag": "",
"title": "生活大不同(17)",
"playTime": "",
"customParam": { }
}

```

发送的参数说明:

索贝参数	必填	类型	说明
id	true	String	视频 guid, 32 位字符串
siteId	true	String	站点 id
alias	True	String	站点英文别名
sourceFrom	True	Int	视频来源 1: 拍客上传 2: Web 上传 3: Ftp 上传 4: 综合入库 5: 在线快编

			6: 视频收录 7: 迁移入库 8: 主干入库 9: 素材转码
classifyType	True	Integer	0 无频道 default 枚举值根据在后台定义的 id
catalogId	True	Integer	视频直属栏目 id
catalogTree	true	JSONArray	栏目树
status	true	Integer	视频状态 枚举值: -1:初稿状态 0:待发布 1:发布 2:转码中 3:转码失败 4:回收站 5:流转中 6:流转异常 7:正在上传 8:取消转码
subTitle	false	String	副标题
title	false	String	标题
tag	false	String	视频的标签
description	false	String	视频描述
imageList	true	JSONArray	该视频节目的关键帧图片
programLength	true	Integer	节目时长
createTime	true	String	创建时间
publishTime	false	String	发布时间
interfacesTypes	false	String	下发的类型平台 根据站点配置的下游平台的 code 决定
vodAddress mp4Address tsAddress	true	JSONObject	该节目的多码率分片播放地址 vodAddress -> isSuccess: 1: 获取的播出地址及可用 0: 解析播放URL出现问题, 或者因为转码没有完成未产生可用播

		<p>出URL</p> <p>vodAddress ->message: 获取播出URL的返回信息</p> <p>vodAddress ->duration : 该节目的总时长</p> <p>vodAddress ->title : 视频的名称</p> <p>vodAddress ->host : 视频的播放前缀</p> <p>vodAddress ->subPath: 介于视频播放域名和url间的相对路径目录</p> <p>vodAddress ->clips: 包含了分片视频的信息</p> <p>vodAddress ->formats: 多中码率对应的码率标题 比如是高清, 标清, 还是流畅等</p> <p>vodAddress ->clips->duration: 该分片视频的时长</p> <p>vodAddress ->clips->title: 该分片视频的名称</p> <p>vodAddress ->clips->urls: 该分片的不同码率的播放地址</p> <p>其中完整的播放地址为: host 的播放前缀 + clips->urls[i] 并且 urls 中的码率顺序对应着 formats 中的码率标题的顺序</p> <p>比如说: url 中的第一个播放地址为 450K, formats 中的第一个值为 标清, 那么 450K 就对应到标清。</p>
--	--	---

customParam	false	JSONObject	自定义字段的存放字段
--------------------	-------	------------	------------

catalogTree 的参数说明:

参数	必填	类型	说明
catalogId	true	Integer	栏目 id
catalogName	true	String	栏目名称
treeLevel	true	Integer	栏目层级

imageList 参数说明:

参数	必填	类型	说明
imagePath	True	String	图片的地址

customParam 的参数说明:

参数	必填	类型	说明
key1	false	String	自定义值
Key2	false	String	自定义值

返回参数的 JSON 格式:

```
{
  "id": "7e7e0184c1b13f2bf63bc5b263c5d0a2",
  "partnerCode": "letvCMS",
  "status": 1,
  "message": "ok"
}
```

返回的参数说明:

参数	必填	类型	说明
id	true	String	视频 id
partnerCode	true	String	下游平台的唯一标示比如 乐视: letvCMS
status	true	Integer	处理数据的结果
message	true	String	处理结果的详细信息

5.2 音频更新通知机制

5.2.1 接口说明

接口简介: 当 VMS 里面管理的音频发布或者被推送的时候, 可以相关的信息主动通知

下游 CMS 平台，让下游平台数据保持最新和最近修改

5.2.2 应用示例

➤ 接口 **post** 音频和站点等相关信息给下游 **CMS** 应用举例：

注意：本示例为测试机数据，并非网站实际数据，例子仅展现数据结构。

POST 传入 JSON 格式：

```
{
  "id": "c727ab4e2fd54e32815dffa6a1e9e944",
  "classifyType": 0,
  "catalogId": 2,
  "catalogTree": [
    {
      "catalogId": 2,
      "catalogName": "默认音频栏目",
      "treeLevel": 1
    }
  ],
  "createTime": "2016-10-09 10:16:00",
  "description": "",
  "madeFrom": "Web 上传",
  "imageList": [
    {
      "defaultKeyFrame": 0,
      "imagePath": "http://101.201.66.168:82/cszd/upload/Image/default/2016/09/26/bb5fd1ddd1024649896aeea2434e0706/Cook dog faces_640_400.jpg"
    }
  ],
  "interfacesTypes": "",
  "playerCodeList": [
    {
      "name": "默认播放器",
      "defaultFlag": "1",
      "playerCode": "<div id=\"cb31c1d2413d47879c5c3d61227074e6\"></div><script type=\"text/javascript\">var vmsPlayer_callback = function () {createPlayer(\"video:/id:c727ab4e2fd54e32815dffa6a1e9e944\", \"%3Fguid%3Dc727ab4e2fd54e32815dffa6a1e9e944%26siteId%3D895\", \"2016-10-09\", \"@WIDTH@\", \"@HEIGHT@\")};</script><script type=\"text/javascript\" src=\"http://101.201.66.168:84//cszd/media/playerJson/vod/cb31c1d2413d47879c5c3d61227074e6.js\"></script>"
    },
    {
      "name": "视频播放器",
      "defaultFlag": "0",
      "playerCode": "<div id=\"39ea704d8d6545e99c9e3b92373e4264\"></div><script type=\"text/javascript\">var vmsPlayer_callback = function () {createPlayer(\"video:/id:c727ab4e2fd54e32815dffa6a1e9"
    }
  ]
}
```

```
e944\\",\"%3Fguid%3Dc727ab4e2fd54e32815dffa6a1e9e944%26siteId%3D895\\",\"2016-10-09\\",\"@WIDTH@\\",\"@HEIGHT@\\");</script><script type=\"text/javascript\" src=\"http://101.201.66.168:84//cszd/media/playerJson/vod/39ea704d8d6545e99c9e3b92373e4264.js\"></script>\"

    },
    \"programLength\": 231,
    \"publishTime\": \"2016-10-09 10:16:53\",
    \"status\": 1,
    \"tag\": \"\",
    \"title\": \"李翊君 - 沙漠寂寞\",
    \"playTime\": \"\",
    \"playURL\": {
        \"duration\": 231,
        \"title\": \"李翊君 - 沙漠寂寞\",
        \"host\": \"http://123.56.21.89/cszd/vod/\",
        \"clips\": [
            {
                \"duration\": 231,
                \"index\": \"001\",
                \"urls\": [
                    \"2016/10/09/c727ab4e2fd54e32815dffa6a1e9e944/c727ab4e2fd54e32815dffa6a1e9e944_mp3_192k_mp3.mp3\",
                    \"2016/10/09/c727ab4e2fd54e32815dffa6a1e9e944/c727ab4e2fd54e32815dffa6a1e9e944_mp3_128k_mp3.mp3\",
                    \"2016/10/09/c727ab4e2fd54e32815dffa6a1e9e944/c727ab4e2fd54e32815dffa6a1e9e944_mp3_64k_mp3.mp3\"
                ]
            }
        ]
    },
    \"formats\": [
        \"高品质\",
        \"标准\",
        \"流畅\"
    ],
    \"subPath\": \"cszdod/\",
    \"isSuccess\": \"1\",
    \"message\": \"播出 URL 信息获取成功!\",
    \"ftpUrls\": [
        \"ftp://other:other@123.56.21.11//vod/2016/10/09/c727ab4e2fd54e32815dffa6a1e9e944/c727ab4e2fd54e32815dffa6a1e9e944_mp3_192k_mp3.mp3\",
        \"ftp://other:other@123.56.21.11//vod/2016/10/09/c727ab4e2fd54e32815dffa6a1e9e944/c727ab4e2fd54e32815dffa6a1e9e944_mp3_128k_mp3.mp3\",
        \"ftp://other:other@123.56.21.11//vod/2016/10/09/c727ab4e2fd54e32815dffa6a1e9e944/c727ab4e2fd54e32815dffa6a1e9e944_mp3_64k_mp3.mp3\"
    ]
}
```

```

    ]
  },
  "customParam": { }
}

```

发送的参数说明:

索贝参数	必填	类型	说明
id	true	String	视频 guid, 32 位字符串
classifyType	True	Integer	0 无频道 default 枚举值根据在后台定义的 id
catalogId	True	Integer	音频直属栏目 id
catalogTree	true	JSONArray	栏目树
status	true	Integer	视频状态 枚举值: -1:初稿状态 0:待发布 1:发布 2:转码中 3:转码失败 4:回收站 5:流转中 6:流转异常 7:正在上传 8:取消转码
subTitle	false	String	副标题
title	false	String	标题
description	false	String	音频描述
imageList	true	JSONArray	该音频节目的关键帧图片
programLength	true	Integer	节目时长
createTime	true	String	创建时间
publishTime	false	String	发布时间
interfacesTypes	false	String	下发的类型平台,根据站点配置的CODE相关,比如 ChinaCache :代表蓝汛CDN FastWeb:代表快网CDN Letv:代表乐视CDN
playURL	true	JSONObject	该节目的多码率分片播放地址 playURL->isSuccess:

			<p>1: 获取的播出地址及可用</p> <p>0: 解析播放URL出现问题, 或者因为转码没有完成未产生可用播出URL</p> <p>playURL->message: 获取播出URL的返回信息</p> <p>playURL->duration : 该节目的总时长</p> <p>playURL->title: 音频的名称</p> <p>playURL->host: 音频的播放前缀</p> <p>playURL->subPath: 介于音频播放域名和url间的相对路径目录</p> <p>playURL->clips: 包含了分片音频的信息</p> <p>playURL->formats: 多中码率对应的码率标题 比如是高清, 标清, 还是流畅等</p> <p>playURL->clips->duration: 该分片的时长</p> <p>playURL->clips->title: 该分片的名称</p> <p>playURL->clips->urls: 该分片的不同码率的播放地址</p> <p>其中完整的播放地址为: host 的播放前缀 + clips->urls[i] 并且 urls 中的码率顺序对应着 formats 中的码率标题的顺序</p> <p>比如说: url 中的第一个播放地址为 450K, formats 中的第一个值</p>
--	--	--	---

			为标清，那么 450K 就对应到标清。
customParam	false	JSONObject	自定义字段的存放字段

catalogTree 的参数说明：

参数	必填	类型	说明
catalogId	true	Integer	栏目 id
catalogName	true	String	栏目名称
treeLevel	true	Integer	栏目层级

imageList 参数说明：

参数	必填	类型	说明
imagePath	True	String	图片的地址

customParam 的参数说明：

参数	必填	类型	说明
key1	false	String	自定义值
Key2	false	String	自定义值

返回参数的 JSON 格式：

```
{
  "id": "7e7e0184c1b13f2bf63bc5b263c5d0a2",
  "partnerCode": "letvCMS",
  "status": 1,
  "message": "ok"
}
```

返回的参数说明：

参数	必填	类型	说明
id	true	String	视频 id
partnerCode	true	String	下游平台的唯一标示比如 乐视：letvCMS
status	true	Integer	处理数据的结果
message	true	String	处理结果的详细信息