

# VTC 移动可视云平台及 SDK 产品和服务

## Web 版本接口文档

发布时间：2016 年 3 月

软件版本：V2.4

南京微特喜网络科技有限公司

# VTC 移动可视云平台及 SDK 技术 Web 版本接口文档

版本号：V2.4

文档变更记录		
版本号	完成日期	变更描述
0.1	2014-12-1	Initial version
0.2	2015-2-30	Reworked version
1.0	2015-3-30	First Edition version
2.0	2015-6-26	Second version
2.1	2015-9-2	1. 增加 3.23 视频直播, 3.24 视频观看 --- by gzy 2. 增加 4 客服应用 --- by hyw
2.2	2016-1-4	1. 增加观看视频示例代码 --- by hyw
2.3	2016-3-1	1. 增加根据用户 id 查询视频接口 --- by hyw
2.4	2016-3-29	1. 增加多条件查询视频接口 --- by hyw

## 目录

<b>1 引言</b> .....	<b>4</b>
1.1 概要.....	4
1.2 重要假设条件.....	4
<b>2 运行环境</b> .....	<b>4</b>
2.1 系统架构.....	5
2.2 用户接入管理.....	5
<b>3 SDK 接口规范</b> .....	<b>6</b>
3.1 第三方用户接入.....	6
3.1.1 第三方 id 注册.....	6
3.1.2 第三方 id 登录.....	7
3.1.3 用户更新设备号（仅针对 iphone 用户）.....	7
3.1.4 用户退出登录(该接口同微特喜用户所用接口).....	8
3.1.5 第三方用户管理.....	8
3.2 网页 javascript SDK.....	10
3.2.1 第三方登录接口说明.....	10
3.2.1.1 初始化.....	10
3.2.1.2 登录.....	10
3.2.1.3 注册.....	10
3.2.2 WebIM 接口说明.....	11
3.2.2.1 WebIM 登录.....	11
3.2.2.2 设置状态.....	11
3.2.2.3 发送消息.....	12
3.2.3 视频直播接口说明.....	12
3.2.3.1 登陆.....	12
3.2.3.2 退出.....	12
3.2.3.3 视频直播开始.....	12
3.2.3.4 视频直播停止.....	13
3.2.4 视频观看接口说明.....	13
3.2.4.1 视频观看.....	14
3.2.4.2 根据用户 id 查询视频.....	15
3.2.4.3 根据组合条件查询视频.....	16
<b>4 客服应用</b> .....	<b>17</b>
4.1 客服机器人和人工坐席创建流程.....	17
4.2 PC 端客服使用流程.....	18
4.3 APP 端客服使用流程.....	20
<b>5 版本升级</b> .....	<b>20</b>
5.1 后向兼容.....	20
5.2 互联互通.....	20
<b>6 GLOSSARY 词汇表</b> .....	<b>20</b>
6.1 Acronyms 首字母缩写.....	20

# 1 引言

## 1.1 概要

VTC 移动可视云平台是一个提供 PaaS 层移动流媒体通信服务的云计算平台。平台提供 SDK 关键技术，支持视频分享、IP 电话、视频通信、视频直播等平台级移动流媒体基础功能。用户通过 SDK 技术可在其终端 (APP, iPad, TV) 快速实现类似微信的基础功能，并基于该基础功能开发自身产品和业务。

## 1.2 重要假设条件

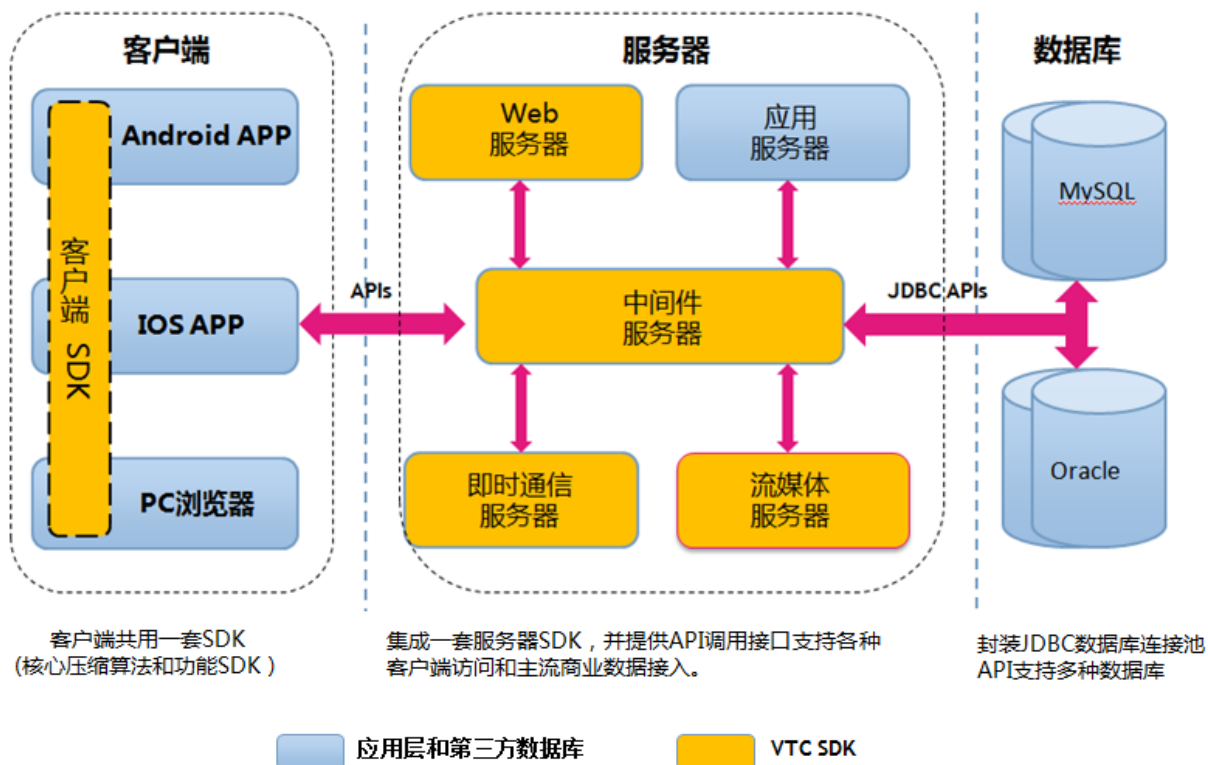
- 用户具有基础的 Android 应用层开发能力
- 用户具备基本的 Android 代码调试、跟踪能力
- 用户具备基础的 Javascript 开发能力
- 用户具备基础的 JQuery 开发能力
- 用户具备基本的 Javascript/JQuery 调试、除错能力

## 2 运行环境

	硬件	软件
终端	Android 手机	OS: Android 4.0 系统以上。 HW: 推荐 2012 年以后的双核以上手机。 带支持 NEON 多媒体加速指令集 (伪硬件模块), "ARMv7 with NEON" or "Cortex A8/A9 "及以上
	iPhone 手机	OS: iOS-7, iOS-8 HW: iPhone4S, iPhone5, iPhone5S, iPhone6, iPhone6P (推荐 iPhone5s)
	PC 机	OS: Windows7, WindowsXP, 浏览器: 要求支持 IE8, Safari(on IOS), Chrome 安装 VTC 插件
服务器	x86CPU/4 核/ 8G RAM / 720G Disk 以上配置	Linux OS: CentOS 6.2 Database: MySQL, Oracle10g Web Server: Apache/Tomcat7
网络	Wi-Fi: 电信 DSL+WLAN	流畅直播: 上、下行最小带宽 > 256 kbps 标清直播: 上、下行最小带宽 > 1.2Mbps
	4G: TD-LTE	

	3G: CDMA-EVDO/WCDMA/	
--	----------------------	--

## 2.1 系统架构



## 2.2 用户接入管理

用户进入 [www.vtc365.com](http://www.vtc365.com) 主页, 点击页面右上角的“注册”进入注册页面。选择“开发者用户”进行注册, 按照流程注册成功并登录后, 即可看到如下页面, 获得 appId 和 appToken (用于 App 开发), 如图所示:

我的控制台	直播大厅	视频大厅	图片大厅	产品下载	服务内容	帮助支持
我的应用	我的账户	用户设置				

应用名称	appId	appToken	状态	到期时间	操作
VTCAPP_802252	10197	gYWQawA2uGsxA5xjczWQ	已激活	无限	管理

创建应用

应用名称:

[添加应用](#)

### 3 SDK 接口规范

SDK 接口分为客户端接口和服务器接口。客户端接口分别为 SDK 与 APP 上层应用之间的接口（外部接口）和 SDK 与服务器间接口（内部接口），目前支持 Android, IOS, 和 Web JS 接口。本文档为服务器接口。服务器接口为与客户端和与客户业务服务器之间的接口，主要体现为 HTTP 接口。

服务器接口中提到的 URL 为相对路径的，如没有特殊说明，其根路径为 `http://www.vtc365.com:8080/LiveVideoServer/`

#### 3.1 第三方用户接入

第三方用户接入包括第三方用户的登录和注册，已经对于服务器开放的接口对第三方所属用户进行管理。管理包括添加用户，删除用户，修改用户密码和 id。

第三方用户接入中，用户注册，用户管理已经群的管理，均需要第三方进行鉴权。用户登录需要第三方用户 id 和用户密码。

鉴权参数：

appId	第三方开发 app 的 id
timestamp	以 GMT 为基础的时间戳，是一个长整型，以毫秒为单位，服务器容忍 1 分钟之内的误差
secret	认证信息，该认证信息生成方法为 <code>md5("appId=&lt;appId&gt;&amp;timestamp=&lt;timestamp&gt;&amp;appToken=&lt;appToken&gt;")</code>

##### 3.1.1 第三方 id 注册

第三方用户可以使用第三方 id 进行注册，注册需要使用第三方开发者的 appId 和 appToken 进行鉴权，为安全起见，微特喜建议第三方开发者将 appToken 存在服务器中，而不是存到 app 中，有 app 向服务器申请鉴权信息，避免 appToken 泄露。

URL: `ajax/thirdPartyRegister.do`

发送方法: GET/POST

数据：

名称	说明
thirdPartyId	用户第三方 id，第三方需保证在其 appId 下，该 id 为唯一。
password	用户的密码，该密码可以和第三方密码一致，也可以是不一样的。推荐使用第三方密码的 md5 散列。
appId	用户所属的第三方开发者 id，该 id 为一数字。
timestamp	以 GMT 为基础的时间戳，是一个长整型，以毫秒为单位。

secret	认证信息
--------	------

类别: struts2

类名: com.vtc.livemedia.struts.ThirdPartyOperate

方法: Register()

返回: JSON

result	OK - 成功 Exceed - 失败, 第三方用户达到上限 其它 - 失败, 并表示失败原因
--------	---

### 3.1.2 第三方 id 登录

第三方用户可以使用第三方 id 进行登录, 登录之前需要服务器进行添加用户, 或者客户端进行第三方注册方可。

URL: ajax/thirdPartyLogin.do

发送方法: GET/POST

数据:

名称	说明
thirdPartyId	用户第三方 id, 第三方需保证在其 appId 下, 该 id 为唯一。
password	用户的密码, 该密码可以和第三方密码一致, 也可以是不一样的。推荐使用第三方密码的 md5 散列。
appId	用户所属的第三方开发者 id, 该 id 为一数字。

类别: struts2

类名: com.vtc.livemedia.struts.ThirdPartyOperate

方法: Login()

返回: XML

result	OK - 成功 IdNotFound - 失败, 该第三方 id 不存在 其它 - 失败, 并表示失败原因
--------	---

### 3.1.3 用户更新设备号 (仅针对 iPhone 用户)

为了能够让 iPhone 用户在客户端不在前台的时候也能收到系统消息和信令, Apple 的 APNS 服务需要知道 iPhone 用户的 DeviceId。

URL: ajax/updateDeviceid.do

发送方法: POST/GET

数据

名称	说明
user.deviceId (可选)	用户 iPhone 的 APNS 设备号, 如果不带该参数, 将清除数据库中用户的设备号。

类别: struts2

类名: com.vtc.livemedia.struts.UserOperate

方法: updateDeviceid()

返回: JSON

<b>result</b>	<p>good - 成功</p> <p>init - 失败, 原因是未登录</p> <p>其它 - 失败, 并表示失败原因</p>
---------------	---

### 3.1.4 用户退出登录(该接口同微特喜用户所用接口)

用户退出登录后, 客户端将不能使用任何服务, 网页端可以浏览视频大厅和图片大厅。iPhone 用户退出登录, 将同时清除用户的 deviceId。

URL: logout.do

发送方法: GET

数据: 无

类别: filter (Tomcat 的过滤器)

类名: com.vtc.livemedia.filter.AutoLoginFilter

方法: doFilter()

返回: 跳转至网页 index.jsp

### 3.1.5 第三方用户管理

第三方服务器可以使用该接口管理自己的开发者 appId 所属的用户。

该管理接口需对请求进行鉴权。

该管理接口对请求数有限制, 目前每个开发者 appId 每秒请求时不能超过一个。

该鉴权方法对于服务器之间的时间差比较敏感, 所以为了防止服务器之间犹豫时间不同而产生认证错误, 该接口还提供获取系统时间的方法, 该获取系统时间方法不受以上次数限制。

URL: ajax/thirdPartyManage.do

发送方法: GET/POST

数据:



名称	说明
action	<p>管理方法：</p> <p>mod: 修改 id 或者密码； add: 注册新用户； del: 删除用户； opt: 取得系统时间。</p> <p>action = mod 时带有参数： thirdPartyId - 第三方 id, 确定用户， 下面的 1) 和 2) 只能有一个</p> <p>1) password - 修改密码（密码是用户真实密码的散列）</p> <p>2) newId - 修改第三方 id</p> <p>action = add 时带有参数： thirdPartyId - 第三方 id, password - 密码</p> <p>action = del 时带有参数： thirdPartyId - 第三方 id</p> <p>action = opt 无参数。</p>
thirdPartyId	用户第三方 id, 第三方需保证在其 appId 下, 该 id 为唯一。
newId	新的第三方 id, 第三方需保证该新 Id 与已有的用户不冲突。
password	用户的密码, 该密码可以和第三方密码一致, 也可以是不一样的。推荐使用第三方密码的 md5 散列。
appId	用户所属的第三方开发者 id, 该 id 为一数字。

类别: struts2

类名: com.vtc.livemedia.struts.ThirdPartyOperate

方法: manage()

返回: XML

result	<p>OK - 成功</p> <p>NotAuthenticate - 鉴权失败</p> <p>Busy - 接口调用过于频繁, 例如上一次调用和此次调用间隔小于 1 秒。</p> <p>Exceed - 增加用户超过上限</p> <p>NotFound - 对于 action=mod 和 del 的情况, thirdPartyId 不存在的情况。</p> <p>Error - 其他错误</p>
sysTimeStamp	服务器系统时间

## 3.2 网页 JAVASCRIPT SDK

网页 javascript SDK 主要提供第三方登录，以及 WebIM 登录以及收发消息的接口。

- 第三方登录 JS SDK: vtc3p.min.js
- WebIM JS SDK : webImLogin.min.js

JS SDK 依赖 JQuery 2.1.3 及更高版本，同时依赖以下两个 javascript

- vtcUtil.js
- base64.js

### 3.2.1 第三方登录接口说明

#### 3.2.1.1 初始化

第三方登录需要初始化 vtc3pman，javascript 代码为

```
var vtc3p = new vtc3pman();
```

其中 vtc3p 变量需要为全局或者在登录时能够获取得到。

#### 3.2.1.2 登录

登录代码为:

```
Vtc3p.login(appid, thirdpartyid, pass, succ_callback, fail_callback);
```

参数:

- appid – 为第三方 id，在第三方开发者申请时得到。
- thirdpartyid – 为第三方用户的 id，在第三方用户注册时提供，未经过第三方注册的用户不能使用本接口登录。
- pass – 为第三方用户登录密码，在第三方用户注册时提供。
- succ\_callback – 当登录成功时的回调函数，用户可以用该回调函数进行一些初始化工作。该回调函数带有一个参数 user，该参数为 JSON Object，带有以下属性：
  - userId – 该用户对应 VTC365 的用户号，该用户号用户 WebIM 的登录。
  - password – 该用户登录用的密码，可用于 WebIM 的登录。
- fail\_callback – 当登录失败时的回调函数，带有一个参数，表示登录失败的原因。
  - IdNotFoud – 表示该第三方 id 不存在，如果要使用的话，必须先进行注册。

#### 3.2.1.3 注册

第三方注册需要使用 appToken 进行鉴权，服务器对于鉴权的时间要求比较严格，所以需要和服务器之间的时间进行同步，如果需要进行注册，请在初始化 vtc3pman 后，调用 vtc3pman.checkTime()进行时间同步，否则注册的鉴权可能失败。

注册代码为:

```
Vtc3p.register(appid, timestamp, secret, thirdpartyid, pass, succ_callback, fail_callback);
```

参数:

- **appid** – 为第三方 id，在第三方开发者申请时得到。
- **timestamp** – 时间戳，从 1970-01-01 00: 00: 00 UTC 开始的毫秒数。以服务器时间为准，需要将当前时间加上 `vtc3pman.timediff`。
- **secret** – 鉴权密码，需要用 **appid**，**timestamp** 和 **appToken** 进行计算。
- **thirdpartyid** – 为第三方用户的 id，在第三方用户注册时提供，未经过第三方注册的用户不能使用本接口登录。
- **pass** – 为第三方用户登录密码，在第三方用户注册时提供。
- **succ\_callback** – 当登录成功时的回调函数，用户可以用该回调函数进行一些初始化工作。该回调函数带有一个参数 **user**，该参数为 **JSON Object**，带有以下属性：
  - **userId** – 该用户对应 VTC365 的用户号，该用户号用户 **WebIM** 的登录。
  - **password** – 该用户登录用的密码，可用于 **WebIM** 的登录。
- **fail\_callback** – 当登录失败时的回调函数，带有一个参数，表示登录失败的原因。

### 3.2.2 WebIM 接口说明

WebIM 需要初始化 `VtcWebIm`，javascript 代码为

```
var webim = new VtcWebIm ();
```

其中 `webim` 变量需要为全局或者在登录以及发送消息时能够获取得到。

#### 3.2.2.1 WebIM 登录

登录代码为：

```
webim.login(userid, pass, handler);
```

参数：

- **userid** – 为 VTC365 用户 id，在第三方登录成功之后可以获得。
- **pass** – 为第三方用户登录密码，在第三方用户注册时提供
- **handler** – 处理 WebIM 事务的处理器，该处理器带有以下成员函数：
  - **handleDisconnect** – 连接断开事件，不带参数。
  - **handleConnected** – 连接事件，不带参数。该事件在 **WebIM** 登录成功后马上触发，所以为了正确的接收消息，该处理函数应该使用 **setStatus()** 上报在线状态。
  - **handleIQSet()** – IQ 请求时间，参数为 IQ 的 XML 文档。
  - **handleConError** – 连接错误时间，带一个参数，为 **Exception** 错误。
  - **handlePresence** – 处理 **Presence** 变动事件，参数为 **Presence** 的 XML 文档。
  - **handleMessage** – 处理正常消息，参数为消息的 XML 文档

#### 3.2.2.2 设置状态

设置状态的代码为：

```
Webim.setStatus(status);
```

参数：

- **status** – 设置为当前用户的状态，可以为任意字符串，服务器不解释该字符串，由对方处理。

### 3.2.2.3 发送消息

发送消息代码为：

```
webim.send(to, msgbody);
```

参数：

- **to** – 接收者的 VTC365 用户 id，不能为其他 id。
- **msgBody** – 发送的文本消息体。

### 3.2.3 视频直播接口说明

视频直播需要初始化 **vtcman**，**vtcvideo** 和 **vtcvideoweb**，**javascript** 代码为：

```
var vtcman= new vtcman ();
```

```
var vtcvideo= new vtcvideo (divObject);
```

```
var vtcvideoweb= new vtcvideoweb ();
```

其中 **vtcman**，**vtcvideo** 和 **vtcvideoweb** 变量需要为全局或者在登录以及发送消息时能够获得到。

#### 3.2.3.1 登陆

登陆代码为：

```
vtcman.login(appld, thirdId, pass, success, fail);
```

参数：

- **appid** – 为第三方 id，在第三方开发者申请时得到。
- **thirdpartyid** – 为第三方用户的 id，在第三方用户注册时提供，未经过第三方注册的用户不能使用本接口登录。
- **pass** – 为第三方用户登录密码，在第三方用户注册时提供。
- **success** – 当登录成功时的回调函数，用户可以用该回调函数进行一些初始化工作。该回调函数带有一个参数 **user**，该参数为 **JSON Object**，带有以下属性：
  - **userId** – 该用户对应 VTC365 的用户号，该用户号用户登录。
  - **password** – 该用户登录用的密码，该用户号用户登录。
- **fail** – 当登录失败时的回调函数，带有一个参数，表示登录失败的原因。

#### 3.2.3.2 退出

退出代码为：

```
vtcman.logout();
```

参数：

NA

#### 3.2.3.3 视频直播开始

视频直播开始代码为：

```
vtcvideoweb.start(success, fail, title, description, save, calltype, content);
```

参数:

- **success** – 返回成功时的回调函数，该回调函数中带有参数：
  - **videoID** – 视频直播的 id;
  - **host** – RTP/UDP 主机地址;
  - **port** – RTP/UDP 端口;
  - **rtspURL** – 视频直播的 RTSP 格式的 url;
  - **url** – 视频的 HLS 格式的 url;用户需在此回调函数中需要调用 `vtcvideo.startVideo(dev, resolution, url)`，该函数参数为：
  - **dev** -- 直播时使用的摄像头，可通过调用函数 `vtcvideo.getDevList()`获取;
  - **resolution** – 分辨率，获取 **resolution** 之前需要通过调用 `vtcvideo.getColorSpaces(dev)`来获取 **colorSpaces**，在得到 **dev** 和 **colorSpaces** 之后，可以通过调用函数 `vtcvideo.getResolutions(dev, colorSpaces)`获取 **resolution**;
  - **url** – 视频的 HLS 格式的 url;
- **fail** – 返回失败时的回调函数，带有一个参数，表示失败的原因;
- **title** – 视频的标题，如果省略，视频的标题缺省为“无标题”;
- **description** – 视频的描述，如果省略，缺省为：“作者有点懒，没有提供该视频的描述信息”;
- **save** – 表示是否存成文件，1 – 存成文件，否则不存文件（缺省不存）;
- **calltype** – 直播类型, DT – 定向直播;
- **shareModel** – 共享模式: 1-公开; 2-私有; 3-圈子; 4-密码
- **content** – 视频类型（默认为 3—音视频）。

### 3.2.3.4 视频直播停止

视频直播停止代码为:

停止直播中需要调用两个函数，先调用 `vtcvideo.stopVideo()`，再调用 `vtcvideoweb.stop (success, fail)`;

`vtcvideo.stopVideo()`的参数:

NA

`vtcvideoweb.stop (success, fail)`参数:

- **success** – 视频停止直播返回成功时的回调函数;
- **fail** – 视频停止直播返回失败时的回调函数，带有一个参数，表示失败的原因;

### 3.2.4 视频观看接口说明

视频观看需要初始化 `vtcvideo` 代码为:

```
var vtcvideo= new vtcvideo (divObject);
```

其中 `vtcvideo` 变量需要为全局或者在登录以及发送消息时能够获取得到。

### 3.2.4.1 视频观看

视频观看代码为:

```
vtcvideo.videoPlay(videoID, playUrl, rtspURL, liveFlag);
```

参数:

- videoID – 视频 ID
- playUrl – 要观看的视频 URL, 这里需要视频的完整的 URL
- rtspURL – 要观看的直播视频 RTSP 格式的 URL(只在观看直播视频的时候使用)
- liveFlag – 是否为直播视频, 1—直播视频, 其他—非直播视频

注:目前非直播视频的观看我们只支持 MP4 和 M3U8 格式。

使用 flash 播放视频实例:

```
<!DOCTYPE>
<html>
<head>
<title>LiveVideo|播放视频</title>
<script src="http://www.vtc365.com:8080/LiveVideoServer/js/jquery-
1.7.1.min.js" type="text/javascript"></script>
<script src="http://www.vtc365.com:8080/LiveVideoServer/js/jwplayer.min.js"
type="text/javascript"></script>
<script src="http://www.vtc365.com:8080/LiveVideoServer/js/swfobject.js"
type="text/javascript"></script>
</head>
<body>
<div>
<div id="player" class="playerBox">缓冲中, 请稍等...</div>
</div>
</body>
<script type="text/javascript">
var host = "http://www.vtc365.com:8080/LiveVideoServer";
function playStream(streamURL) {
var viewer = jwplayer("player");
viewer.setup({
height : 400,
width : 600,
flashplayer : host+"/tools/player.swf",
provider : host+"/tools/AdaptiveProvider.swf",
```

```

file : streamURL,
stretching: 'exactfit',
volume: 100,
skin: host+'/style/skin/snel/snel.xml',
plugins :{'http://www.vtc365.com:8080/LiveVideoServer/js/jwlivestatus.js':{}},
events : {
  onError : function(event) {
    alert(event.message);
  }
};
viewer.play(true);
//视频观看 url
var streamURL = host+"/streams/805076/1451877763084/live.m3u8";
playStream(streamURL);
</script>
</html>

```

### 3.2.4.2 根据用户 id 查询视频

第三方用户可以在服务端使用 httpClient 的方式轮询，可以获取该用户的历史视频或正在直播中的视频。

URL: `ajax/m_listVideoByUid.do`

发送方法: GET/POST

数据:

名称	说明
uid	直播者的用户 id
liveFlag	0:查询历史视频; 1:查询正在直播的视频
appId	用户所属的第三方开发者 id, 该 id 为一数字。
timestamp	当前时间戳
secret	用于鉴权的加密字符串, " appId=" +appId+ " &timestamp=" +timestamp+ " &appToken=" +appToken 的 md5 加密 (32 位小写)

### 3.2.4.3 根据组合条件查询视频

第三方用户可以在服务端使用 httpClient 的方式轮询，可以通过不同条件的组合查询出所需要的视频。

URL: `ajax/listVideoConditional.do`

发送方法: GET/POST

数据:

1. 根据 `groupId` 查询所有直播组下的视频

名称	说明
<code>video.groupId</code>	直播组 id
<code>secret</code>	直播组 token
example	<code>http://www.vtc365.com:8080/LiveVideoServer/ajax/listVideoConditional.do?video.groupId=1&amp;secret=BRPXSI9o7fy9cNt099vv</code>

2. 根据 `appId` 查询所有 `appId` 下的视频

名称	说明
<code>appId</code>	用户创建的 <code>appId</code>
<code>timestamp</code>	时间戳（一般取当前时间）
<code>secret</code>	“ <code>appId=</code> ”+ <code>appId</code> +”&timestamp=”+ <code>timestamp</code> +”&appToken=”+ <code>appToken</code> 的 md5 加密(32 位小写)
example	<code>http://www.vtc365.com:8080/LiveVideoServer/ajax/listVideoConditional.do?appId=11501&amp;timestamp=1459235417475&amp;secret=9779018bb765129407182678a73fb736</code>

3. 根据 `userId` 查询该 `user` 直播的视频以及该 `user` 所属直播组下的所有视频(`video.author`)

名称	说明
<code>video.author</code>	用户 <code>userId</code>
example	<code>http://www.vtc365.com:8080/LiveVideoServer/ajax/listVideoConditional.do?video.author=800101</code>

4. 根据 `liveFlag` 查询

名称	说明
<code>video.liveFlag</code>	默认查所有视频, <code>liveFlag=0</code> : 查历史视频, <code>liveFlag=1</code> : 查当前直播。
example	<code>http://www.vtc365.com:8080/LiveVideoServer/ajax/listVideoConditional.do?video.author=800101&amp;video.liveFlag=1</code>

5. 分页查询



名称	说明
pageNumber	要查询的页码
pageSize	每页的数据
example	http://www.vtc365.com:8080/LiveVideoServer/ajx/listVideoConditional.do?video.author=800101&pageSize=10&pageNumber=1

## 4 客服应用

### 4.1 客服机器人和人工坐席创建流程

步骤 1：在 VTC 官网注册开发者用户并获得 `appId`，详见 2.2 用户接入管理。

步骤 2：点击[管理](#)在跳转到的详情页面左上角点击[客服设置](#)。

步骤 3：修改客服昵称，点击创建客服（客服机器人），如图所示：

步骤 4：点击[创建新席位](#)，创建挂在客服机器人背后的客服坐席账户；或者点击[关联已有席位](#)，把已有用户设定为客服坐席，如图所示：

关联已有席位:

userId:

## 4.2 PC 端客服使用流程

在 4.1 的基础上，执行以下步骤，即可在网页上实现客户咨询和客服坐席应答功能。

步骤 1：在目标网页指定位置（<script></script> 标签内）粘贴以下代码，完成后该页面右下角将会出现在线咨询悬浮框：

注意：（1）请确保粘贴后的代码格式跟文档上严格一致。

（2）将粘贴的代码中第一行 **appId=10000** 换成你创建的应用的 **appId**（4.1 中创建的 **appId**）。

```

/*****          初始化在线咨询悬浮框          *****/
(function() {
    var appId=10000;//将 appId 的值替换成你创建的应用的 appId
    var hostname = "http://www.vtc365.com:8080/LiveVideoServer/";
    var obj_1=document.createElement("div"); //创建一个客户咨询悬浮框
    obj_1.id="com.vtc365.OnlineConsultation";//初始化悬浮框唯一标识
    //设置悬浮框样式（背景，宽度，高度，边框，定位）
    obj_1.setAttribute("style","background: #68b325; position: fixed; width: 200px;height: 37px;"+
    "border-radius: 2px;cursor: pointer;right: 0;bottom: 0;font-size: 18px; line-height: 37px;"+
    "text-align: center; color: #ffffff;z-index:9999;");
    //设置悬浮框上图标及字样
    obj_1.innerHTML="<img style='float:left;margin-left:39px; margin-top:6px;'"+
    "src='"+hostname+"images/mydiv_ico.png' width='28' height='26' />"+
    "<div style='float:left;line-height:37px;margin-left:15px;' >在线咨询</div>";

    document.body.appendChild(obj_1); //将悬浮框加到当前页面
    //点击弹出对话框
    document.getElementById(obj_1.id).onclick=function(){
        window.open(hostname+"chat4help.jsp?appId="+appId, "_blank", "height=650, "+
        "width=600, toolbar =no, scrollbars=no,menubar=no, location=no, status=no");
    };
})();

```

步骤 2：在目标网页指定位置（<script></script> 标签内）粘贴以下代码，该页面左下角将会出现客服登录悬浮框：

```

/*****          初始化坐席登录悬浮框          *****/
(function() {
    var hostname = "http://www.vtc365.com:8080/LiveVideoServer/";
    var obj_2=document.createElement("div");//创建一个坐席登录悬浮框
    obj_2.id="com.vtc365.SeatLogin";//初始化悬浮框唯一标识

```

```

//设置悬浮框样式（背景，宽度，高度，边框，定位）
obj_2.setAttribute("style","background: #68b325; position: fixed; width: 200px; height: 37px;"+
"border-radius: 2px;cursor: pointer;left: 0;bottom: 0;font-size: 18px; line-height: 37px;"+
"text-align: center; color: #ffffff;z-index:9999 ");
//设置悬浮框上图标及字样
obj_2.innerHTML=""+
"<div style='float:left;line-height:37px;margin-left:15px;' >登录坐席</div>";
//将悬浮框加到当前页面
document.body.appendChild(obj_2);
//点击弹出对话框
document.getElementById(obj_2.id).onclick=function(){
    window.open (hostname+"chat4tp.do", "_blank", "height=650, width=960,"+
    " toolbar =no, scrollbars=no,menubar=no, location=no, status=no")
};
})();

```

步骤 3: 客服人员点击登录坐席，弹出登录页面，输入创建坐席时的昵称和密码即可登录，登录完成后弹出对话页面。当有客户点击在线咨询按钮发起询问时，所有客服人员可以在对话框中看到消息，点击客户的 uid 即可回复该客户，当第一个客服人员回复后，该客户将自动被该客服人员接管，实行一对一交流，如图所示：



## 4.3 APP 端客服使用流程

在 4.1 的基础上，执行以下步骤，即可在 APP 上实现客户咨询和客服坐席应答功能。

步骤 1：到 VTC 官网[产品下载](#)页面下载 Android 或 IOS 的 SDK 和 API 文档。

步骤 2：按照对应的 API 文档将 SDK 集成到目标 APP 中，具体请参考[帮助支持](#)页面中的集成指南。

步骤 3：坐席人员使用 4.1 中创建坐席时的昵称和密码登录 APP,即可随时接收客户咨询。

步骤 4：普通用户登录 APP，在通信列表中向客服机器人在线咨询。

**【注】**目前 APP 端 SDK 坐席应答功能正在内测中，将在下一版本推出，同时也将推出独立的移动坐席 APP，敬请期待

## 5 版本升级

### 5.1 后向兼容

### 5.2 互联互通

## 6 Glossary 词汇表

### 6.1 ACRONYMS 首字母缩写

Acronym	Expansion
VTC	Video Technologies Communications
SDK	Software Development Kit
JSON	JavaScript Object Notation
API	Application Interface
RAC	Oracle database Real Application Clusters
RTP	Real Time Protocol
RTSP	Real Time Streaming Protocol
SIP	Session Initiation Protocol
SVN	Subversion management tool