

微吼直播 JS SDK 接入指南

(v1.0.0)



目录

一、	简介.....	- 3 -
二、	SDK 引入、初始化并注入权限验证	- 3 -
1.	依赖库.....	- 3 -
2.	引入 JS 文件.....	- 3 -
3.	初始化及权限验证.....	- 4 -
4.	获取用户参会状态.....	- 6 -
三、	功能详细说明.....	- 7 -
1.	主要调用 API	- 7 -
2.	响应事件 API	- 8 -
3.	SDK 客户端主要错误状态码	- 8 -
4.	主要代码示例.....	- 9 -
附录:	demo.....	- 15 -

一、简介

本文档为了指导开发者更快使用微吼直播的“自助式网络直播服务 SDK”通过引用 JS 的方式开发自己的 PC 网页以及 H5 来对接微吼直播平台，默认读者了解前端开发同时拥有 JavaScript 基础。

目前支持的功能如下：

分类	功能	描述
聊天	观看端聊天	支持在直播时聊天和观看回放时聊天

注：视频、文档以及问答模块目前仍只能使用网页嵌入的方式。

二、SDK 引入、初始化并注入权限验证

1. 依赖库

本 SDK 依赖于 JQuery 库，请在需要引入 JQuery 的页面插入以下代码进行引入：

```
<script
src="http://cnstatic01.e.vhall.com/3rdlibs/jquery/1.11.2/jquery.min.js"
type="text/javascript"></script>
```

2. 引入 JS 文件

在需要调用 SDK 的页面插入以下代码以引入 JS 文件：

```
<script type="text/javascript"
src="http://cnstatic01.e.vhall.com/static/js/webSDK/1.0.0/vhallSDK.js">
</script>
```

目前暂不支持 https

注：支持使用 **AMD/CMD** 标准模块加载规范加载

3. 初始化及权限验证

请使用以下的方法和参数进行 SDK 的初始化以及权限的验证

```
VHALL_SDK.init({  
  
  account : '',//必填, 唯一用户 id  
  
  email : '',//参数必填, 值可为空, 用于与嵌入视频区用户同步  
  
  username : '',//必填, 用户昵称用以在各模块中显示  
  
  roomid : '',//必填, 当前聊天所在活动 id  
  
  appkey : '',//必填, 启用服务唯一验证, API/SDK 权限申请后自行生成  
  
  signedat : '',//必填, 启用服务唯一验证, unix 时间戳  
  
  sign : '',//必填, 启用服务唯一验证, 有效期一小时, 见下面的生成规则  
  
  facedom : '',//可选, 当使用微吼表情时必填, 呼出表情弹窗 dom 属性 如  
  ('#id',' .class'等)  
  
  textdom : '',//可选, 当使用微吼表情时必填, 插入表情文本 dom 属性 如  
  ('#id',' .class'等)  
  
});
```

注: 为确保嵌入页面的用户与 SDK 各模块中的用户是同一个用户, 请尽量传入与视频区嵌入相同的 email 参数, 否则会被统计成两个不同的用户

参数 sign 的生成规则如下:

将 account、email、username、roomid、app_key(注意此处参数名为 app_key 而不是 init 方法中的 appkey)以及 signedat 这些参数:

1	按参数名升序排列
2	然后按参数名 1+参数值+参数名 2+参数值拼接
3	在首尾各加上 secret_key
4	计算 md5 作为 sign

```
<?php

$secret_key = "user_secret_key";

$params = [

    "roomid"=> "123456789",

    "account"=> "accountid",

    "email"=> "user_email",

    "username"=> "username",

    "app_key"=> "user_app_key",

    "signedat"=> "timestamp_now"

];

// 按参数名升序排列

ksort($params);

// 将键值组合

array_walk($params,function(&$value,$key){

    $value = $key . $value;

});

// 拼接,在首尾各加上$secret_key,计算 MD5 值

$sign = md5($secret_key . implode(',',$params) . $secret_key);

// 结果形如

//

$sign=md5("user_secret_keyaccountaccountidapp_keyuser_app_keyemailuser_emailroomid123456789signedattimestamp_nowusernameusernameuser_secret_key");
```

```
// 计算结果

// $sign = 'c641af5046aa0599ab58c57a52a24f83'; //签名有效期只有一个小时
```

初始化主要错误码

错误码	说明
1001	活动不存在
1002	用户昵称不能空
1003	用户 id 不能空
1004	appkey 不存在
1005	签名验证超时
1006	签名错误
1007	当前活动不是该 appkey 下所属活动

对上述内容中所提 appkey、app_key 以及 secret_key 三个参数的说明：

- appkey 和 app_key 均表示添加应用后所得到的 AppKey 列对应的值
- secret_key 表示添加应用后所得到的 SecretKey（注意与 AppSecretKey 区分）

列对应的值

4. 获取用户参会状态

初始化完成后可通过 VHALL_SDK.onReady 获取当前参会用户状态，在 VHALL_SDK.userinfo 对象下会有用户踢出状态，禁言状态，头像等信息，示例代码：

```
VHALL_SDK.onReady = function() {

    console.log(VHALL_SDK.userinfo);

    console.log(VHALL_SDK.room);

};
```

VHALL_SDK.userinfo 和 VHALL_SDK.room 在 consolelog 中返回的内容如下：

```
/**
 * VHALL_SDK.userinfo = {
 *
 * avatar:"//cnstatic01.e.vhall.com/static/images/watch/head50.png" //头像
 *
 * forbidchat:"" //"":未开启全员禁言, 1: 已开启全员禁言
 *
 * is_gag:0 //0: 未禁言, 1: 禁言
 *
 * is_kickout:0 //0: 未踢出, 1: 踢出
 *
 * role:"user"//用户角色
 *
 * userid:"用户参会 id"
 *
 * username:"昵称"
 *
 * }
 *
 * VHALL_SDK.room = {
 *
 * type: 1 // 1: 活动直播中, 2: 预约中, 3: 活动结束
 *
 * }
 */
```

三、功能详细说明

因聊天需要获取用户标识与头像，需提前使用[第三方创建用户接口](#)创建用户，如传入的第三方用户信息未匹配到用户标识，将自动进行创建，并默认为游客以及使用微吼的默认的头像。

注：目前聊天模块只支持观看端，如需要对聊天进行管理操作（禁言、踢出等功能），请前往微吼平台上进行操作。

1. 主要调用 API

API	说明	返回内容
-----	----	------

sendChat({text:''})	聊天消息发送, text: 消息文本, 长度限制为 140 个字符	组装好的聊天数据
vhall_get_live_history_chat_msg()	请求成功回调函数,data 消息体数组	
vhall_get_record_history_chat_msg(curr_page)	回放拉取历史聊天, curr_page: 需拉取数据的页数, 每页 20 条	

2. 响应事件 API

API	说明
onUserOnline	用户上线事件消息
onUserOffline	用户下线事件消息
onChatMessage	直播新消息事件
onSendChatSuccess	发送聊天消息成功
onSendChatError	发送聊天消息失败
onDisadbleChat	直播禁言消息
onPermitChat	直播恢复禁言消息
onForbidchat	直播全员禁言消息, 1 为全员禁言, 0 为恢复全员禁言
onKickout	直播踢出消息
onKickoutrestore	直播恢复踢出消息
on_vhall_live_history_chat_msg	拉取直播历史消息, 拉取 50 条
on_vhall_record_history_chat_msg	拉取回放历史消息, 分页拉取, 每页 20 条

3. SDK 客户端主要错误状态码

错误码	说明
10000	消息体格式不对
10001	消息体为空
10002	当前用户被禁言
10003	聊天输入不能超过 140 个字符
10004	当前已开启全员禁言
20000	接口请求成功
20005	没有数据

4. 主要代码示例

用户上线消息

```
/**
 * [onUserOnline 用户上线]
 * @param {[type]} msg [description]
 * @return {[type]} [description]
 */
VHALL_SDK.onUserOnline = function(msg) {
    console.log(msg);
};
```

用户下线事件消息

```
/**
 * [onUserOffline 用户下线]
 * @param {[type]} msg [description]
 * @return {[type]} [description]
 */
VHALL_SDK.onUserOffline = function(msg) {
    console.log(msg);
};
```

直播新消息事件

```
/**
```

```
* [onChatMessage 直播收到聊天消息]
* @param {[type]} msg [object]
* @return {[type]} [description]
*/
VHALL_SDK.onChatMessage = function(msg) {
    console.log(msg);
};
```

发送聊天消息成功

```
/**
 * [onSendChatSuccess 直播消息发送成功]
 * @return {[type]} [description]
 */
VHALL_SDK.onSendChatSuccess = function(msg) {
    console.log(msg);
};
```

发送聊天消息失败

```
/**
 * [onSendChatError 直播消息发送失败]
 * @return {[type]} [description]
 */
VHALL_SDK.onSendChatError = function(msg) {
```

```
        console.log(msg);  
    };
```

直播禁言消息

```
/**  
 * [onDisadbleChat 直播禁言消息]  
 * @param {[type]} [禁言用户 id]  
 * @return {[type]} [description]  
 */  
VHALL_SDK.onDisadbleChat = function(userid) {  
    console.log(userid);  
};
```

直播恢复禁言消息

```
/**  
 * [onPermitChat 直播恢复禁言消息]  
 * @param {[type]} userid [恢复禁言用户 id]  
 * @return {[type]} [description]  
 */  
VHALL_SDK.onPermitChat = function(userid) {  
    console.log(userid);  
};
```

直播全员禁言消息

```
/**
 * [onForbidchat 直播全员禁言消息]
 * @param {[type]} status [1:开启, 0: 关闭]
 * @return {[type]} [description]
 */
VHALL_SDK.onForbidchat = function(status) {
    console.log(status);
};
```

直播踢出消息

```
/**
 * [onKickout 直播踢出消息]
 * @param {[type]} userid [被踢出用户 id]
 * @return {[type]} [description]
 */
VHALL_SDK.onKickout = function(userid) {
    console.log(userid);
};
```

直播恢复踢出消息

```
/**
 * [onKickoutRestore 直播恢复踢出消息]
```

```
* @param {[type]} userid [恢复踢出用户 id]

* @return {[type]} [description]

*/

VHALL_SDK.onKickoutRestore = function(userid) {

    console.log(userid);

};
```

拉取直播历史消息，拉取 50 条

```
/**

* [on_vhall_record_chat_msg 获取直播历史消息成功回调]

* @param {[arraylist]} msg [数据数组]

* @return {[type]} [description]

*/

VHALL_SDK.on_vhall_live_record_chat_msg = function(msg) {

    console.log(msg);

};
```

拉取回放历史消息，分页拉取，每页 20 条

```
/**

* [on_vhall_record_chat_msg 获取回放历史消息成功回调]

* @param {[arraylist]} msg [数据数组，回放中有 curr_page,total,total_page 有数据]

* @return {[type]} [description]
```

```
*/  
  
VHALL_SDK.on_vhall_record_history_chat_msg = function(msg) {  
  
    console.log(msg);  
  
};
```

获取直播历史消息

```
/**  
  
 * [vhall_get_live_history_chat_msg 获取直播历史消息]  
  
 * @type {[type]}  
  
 */  
  
VHALL_SDK.vhall_get_live_history_chat_msg=function(data){  
  
    console.log(data);  
  
});
```

获取第一页回放历史消息

```
/**  
  
 * [vhall_get_record_history_chat_msg 获取第一页回放历史消息]  
  
 * @type {[type]}  
  
 */  
  
VHALL_SDK.vhall_get_record_history_chat_msg(1);
```

附录：**demo**

demo 地址：<http://cnstatic01.e.vhall.com/static/demo/jssdk.html>

demo 提供了基本的样式，开发者可在 demo 样式的基础上修改成自定义的样式