

获取聊天室在线列表

接口描述

1、通过频道号，获取聊天室在线列表

接口URL

https://apichat.polyv.net/front/userlistExternal

请求方式

GET

接口约束

- 1、接口每次请求最多返回1000条数据，最好每次返回100，分页获取
- 2、返回数据量大的情况下，请求频率间隔大于10秒
- 3、数据中包含虚拟观众的信息

请求参数描述

参数名	必选	类型	说明
roomId	true	String	房间号
page	false	Integer	页码，默认1
len	false	String	每一页条数，默认100，最多返回1000条

参数名	必选	类型	说明
toGetSubRooms	false	Boolean	是否获取子频道的用户，true为获取，false为不获取
callback	false	Function	回调函数，用于解决前端请求跨域问题

返回结果解密：

```
const aesjs = require('aes-js');
const Base64 = require('js-base64');
const md5 = require('md5');

function decrypt(encrypted, sign) {
  const encryptedBytes = aesjs.utils.hex.toBytes(encrypted);
  const key = str2ab(sign, 16);
  const iv = key;
  const aesCfb = new aesjs.ModeOfOperation.cbc(key, iv, 8);
  const decryptedBytes = aesCfb.decrypt(encryptedBytes);

  const decryptedText = aesjs.utils.utf8.fromBytes(decryptedBytes);
  return Base64.decode(decryptedText);
  function str2ab(str, len) {
    const bytes = aesjs.utils.utf8.toBytes(str);
    const bytesLen = bytes.length;
    if (bytesLen > len) {
      return bytes.slice(0, len);
    }
    return (
      [].concat(...(new Array(Math.ceil(len / bytesLen)).fill(bytes)))
    ).slice(0, len);
  }
}

function createApiSign(appSecret, data = {}, type = 'upper') {
  //1、把data里的数据按字典序排列
  //2、按照key+value进行拼接,在首尾分别加上appSecret
  //3、MD5加密,默认大写,当type=lower时,为小写
  let content = appSecret;
  const temp = Object.keys(data).sort();
  temp.forEach((item) => {
    if (item === 'sign') {
      return;
    }
    const v = data[item];
    content += item + (isObject(v) ? JSON.stringify(v) : v);
  });
  content += appSecret;
  const sign = md5(content);
  return type === 'lower' ? sign : sign.toLocaleUpperCase();
}

function isObject(obj){
  return obj !== null && obj instanceof Object;
}

// 1、第一步，访问链接地址，获取返回内容
// http://apichat.polyv.net/front/userlistExternal?roomId=412738&page=1&len=100&hide=0&toGetSubRooms=true

const sign = createApiSign('polyvChatSignForExternal', { roomId: '412738', page: '1', len: '100', hide: '0',
  toGetSubRooms: 'true' }); // 接口请求的参数的签名
//2、第二步，将第一步的返回内容赋值给encrypted进行解析
```

```
const encrypted = 'fd1206646ea5e702852157';
console.log(decrypt(encrypted, sign));
```

响应参数描述

参数名	类型	说明
count	Integer	总人数
userlist	Array	用户对象（具体看下面用户对象说明）

用户对象说明

参数名	类型	说明
actor	String	身份信息，讲师、助教等
banned	Boolean	是否禁言
channelId	String	频道号
clientIp	String	用户ip
nick	String	昵称
pic	String	头像
roomId	String	房间号
uid	String	socket id
userId	String	用户id
userSource	String	信息来源

参数名	类型	说明
userType	String	用户类型 assistant：助教 student：普通观众 slice：普通观众 dummy：虚拟观众 teacher：讲师 guest：嘉宾 attendee：研讨会参会者（频道场景为研讨会场景时才有） manager：直播监控管理员 talent：同台主播（频道场景为MR时才有）

javascript请求示例

```
const formData = {
  roomId: '412738',
  page: 1,
  len: 100,
  hide: 0,
  toGetSubRooms: true,
};
ajax({
  url: 'https://apichat.polyv.net/front/userlistExternal',
  formData,
  type: 'get',
}).then(console.log);
```

响应示例

成功示例

```
{
  "count": 1,
  "userlist": [
    {
      "banned": false,
      "channelId": "412738",
      "clientIp": "",
      "nick": "广州观众/70375",
      "param4": "",
      "param5": "",
      "pic": "http://liveimages.videocc.net/defaultImg/avatar/viewer.png",
      "roomId": "412738",
      "uid": "XX17pssG5JvcNzhbAAAU",
      "userId": "1621838373713",
      "userType": "slice"
    }
  ]
}
```