

前端文档

1、获取聊天室socket实例对象

概述：基于观看页SDK开源项目0.11.0版本为例，其他版本也可参考使用获取到socket实例。本文介绍的聊天室相关EVENT事件均没有经过观看页SDK封装，故需要使用下方代码示例进行发送和监听，其他已封装过的聊天室EVENT可在帮助中心->聊天室模块文档或开源项目中查看使用

```
// [获取聊天室 socket 实例]
(https://help.polyv.net/index.html#/live/js/new_sdk/live_watch_sdk/articles/modules/chat/chat-message?id=classmethoddock_plvchatmodule_getsocket "获取聊天室 socket 实例")
// 代码示例
import { getWatchCore } from '@core/watch-sdk';
const watchCore = getWatchCore();
```

2、发送聊天室事件(TASK_STATUS)

概述：前端主动查询任务进展状态

```
// 代码示例 （建议30S调用一次）
const jsonData = {
  EVENT: 'TASK_STATUS',
};
watchCore.chat.getSocket().emit('interactEvent', JSON.stringify(jsonData), (data: string) => {
  const taskStatusData = JSON.parse(data)
  console.warn('interactEvent-TASK_STATUS接收回调', taskStatusData);
  /**
   * @socket {callback} <interactEvent> TASK_STATUS
   * 获取任务状态
   * @returns {object} taskStatusData
   * @returns {object[]} taskStatusData.taskStatus
   * @returns {string} taskStatusData.taskStatus[].taskId 任务id
   * @returns {?number} taskStatusData.taskStatus[].onlineTime 已在线时长（单位：毫秒）
   * @returns {?number} taskStatusData.taskStatus[].signCount 已完成签到次数
   * @returns {boolean} taskStatusData.taskStatus[].isDone 是否达成
   * @returns {?number} taskStatusData.taskStatus[].speakCount 发言次数
   * @returns {string} taskStatusData.taskStatus[].payload 自定义参数
   * @returns {string} taskStatusData.taskStatus[].activityId 活动id
   */
});
```

3、监听聊天室message类型(onSliceID)

```
// 代码示例
watchCore.chat.getSocket().addEventListener('message', (data: string)=> {
  const objData = JSON.parse(data)

  // 后进入直播间的观众触发
  if(objData.EVENT === 'onSliceID') {
    console.warn('onSliceID', objData)
    /**
     * @socket {on} <message> onSliceID
     * 监听onSliceID事件
     * @returns {...}
     * @returns {?object} objData.data.listenInteractEventPayload或listenInteractEventPayloadForCustomTask 监听互
     动事件的数据，表示正在监听中。创建任务监听时，存在callbackUrl则为listenInteractEventPayloadForCustomTask。；否则为
     listenInteractEventPayload
     */
  }
})
```

4、监听聊天室interactEvent类型(START_LISTEN、TASK_DONE、END_TASK、

REMOVE_TASK)

```
// 代码示例
watchCore.chat.getSocket().addEventListener('interactEvent', (data: string)=> {
  const interactData = JSON.parse(data)

  // 先进入直播间的观众触发
  if(interactData.EVENT === 'START_LISTEN') {
    console.warn('START_LISTEN', interactData)
    /**
     * @socket {on} <interactEvent> START_LISTEN
     * 开始监听互动事件
     * @returns {object} interactData
     * @returns {string} interactData.type 互动类型, 'inAndOut' | 'sign' | 'speak' | 'parent'
     * @returns {?string} interactData.activityId 活动id
     * @returns {?Array} interactData.userTags 用户标签
     */
  }

  // 完成任务触发
  if(interactData.EVENT === 'TASK_DONE') {
    console.warn('TASK_DONE', interactData)
    /**
     * @socket {on} <interactEvent> TASK_DONE
     * 完成任务
     * @returns {object} interactData
     * @returns {string} interactData.EVENT TASK_DONE
     * @returns {string} interactData.taskId 任务ID
     * @returns {string} interactData.activityId 活动ID
     * @returns {string} interactData.roomId 房间ID
     * @returns {string} interactData.taskType 任务类型, speakCount、signCount、onlineTime等
     * @returns {object} interactData.taskData 任务数据
     */
  }

  // 任务结束触发
  if(interactData.EVENT === 'END_TASK') {
    console.warn('END_TASK', interactData)
    /**
     * @socket {on} <interactEvent> END_TASK
     * 任务结束通知
     * @returns {object} interactData
     * @returns {string} interactData.EVENT END_TASK
     * @returns {string} interactData.taskId 任务ID
     * @returns {string} interactData.activityId 活动ID
     */
  }

  // 任务删除触发
  if(interactData.EVENT === 'REMOVE_TASK') {
    console.warn('REMOVE_TASK', interactData)
    /**
     * @socket {on} <interactEvent> REMOVE_TASK
     * 任务删除通知
     * @returns {object} interactData
     */
  }
}
```

```

    * @returns {string} interactData.EVENT REMOVE_TASK
    * @returns {string} interactData.taskId 任务ID
    * @returns {string} interactData.activityId 活动ID
    */
  }
})

```

5、签到成功回调

```

/**
 *      checkInSdk.events.SIGN_IN 开始签到
 *      checkInSdk.events.SIGN_IN_SUCCESS 签到成功
 *      checkInSdk.events.SIGN_IN_FINISH 签到限时结束
 *      checkInSdk.events.STOP_SIGN_IN 讲师结束签到
 *      checkInSdk.events.SIGN_IN_FAIL 签到失败
 */
// 1、在开源项目中找到文件src\components\page-watch-common\interactive-receive\check-in\hooks\use-check-in.ts
// 2、改动参考如下
const signInSuccess = () => {
  console.warn('签到成功')
  beforeRemoveCheckInEntrance()
}

onBeforeMount(() => {
  ...
  checkInSdk.on(checkInSdk.events.SIGN_IN_SUCCESS, signInSuccess);
});

onBeforeUnmount(() => {
  ...
  checkInSdk.off(checkInSdk.events.SIGN_IN_SUCCESS, signInSuccess);
});

```

6、监听聊天室customMessage类型(中奖广播通知)

```

// 代码示例
watchCore.chat.getSocket().addEventListener('customMessage', (data: string)=> {
  console.warn('customMessage', data);
});

```

后端文档

7、创建互动监听事件

[创建互动监听事件](#)

8、删除互动监听事件

删除互动监听事件

- 1、END_TASK下发会带上任务id或活动id。
- 2、删除会发送 REMOVE_TASK 事件

9、广播中奖消息接口

并发限制：1秒内，所有请求的userIds的长度加起来不能超过8万

根据用户ID进行广播消息

10、广播开奖接口

发送自定义 消息