

聊天室抢答功能说明

聊天室抢答功能说明

主要说明聊天室抢答功能在观看端以及讲师端如何使用。

1、连接聊天室时，需要带上token和version参数。

[点击获取token说明文档](#)

聊天室连接说明：

```
io.connect('ws://chat.polyv.net', {
  'query': {
    version: '2.0', // version必须大于等于2.0
    token, // token通过接口获取，看上面的链接
  },
  'transports': ['websocket']
});
```

2、监听开始抢答事件

- START_ANSWER_QUICKLY

参数说明

参数名称	参数说明	参数类型
EVENT	事件名称，值为 START_ANSWER_QUICKLY	string
limitTime	抢答时间	number
sessionId	该场直播的sessionId	string
timestamp	发起抢答时的时间戳	number
quicklyId	抢答的id	string
overTime	抢答剩余时间	number

示例

```
socket.on('message', function(jsonData, callback) {
  const data = JSON.parse(jsonData);
  if(data.EVENT === 'START_ANSWER_QUICKLY') {
    // 执行相关操作
  }
})
```

3、学员提交抢答信息

- SUBMIT_ANSWER_QUICKLY

参数说明

参数名称	参数说明	参数类型	是否必填
EVENT	事件名称，写死 SUBMIT_ANSWER_QUICKLY	string	是
quicklyId	抢答id	string	是
useTime	抢答使用的时间	number	是
organization	学员所属机构	string	否

示例

```
// 需要发送的参数，为json字符串
const sendData = JSON.stringify({
  EVENT: 'SUBMIT_ANSWER_QUICKLY',
  quicklyId: '436499f0-17d9-11ea-9ba9-6f8eb568fbc',
  useTime: 2
});
socket.emit('message', sendData , function(d) {
  const { code, message } = JSON.parse(d); // code 为 请求结果状态
  if (code === 200) {
    console.log('抢答成功');
  } else {
    console.log(`抢答失败，失败原因: ${message}`)
  }
});
```

4、获取抢答状态

开始抢答，各个端收到socket消息的时间可能不同，比如pc端比移动端先收到开始抢答的socket，这时时间就会产生偏差。所以需要不断从服务器获取抢答的剩余时间，保证时间准确性。

- 建议1秒获取一次。
- GET_ANSWER_QUICKLY_STATUS

参数说明

参数名称	参数说明	参数类型
EVENT	事件名称，写死 GET_ANSWER_QUICKLY_STATUS	string

示例

```
socket.emit(
  'message',
  JSON.stringify({EVENT: 'GET_ANSWER_QUICKLY_STATUS'}),
  function(d) {
    const { code, data, message } = JSON.parse(d);
    if (code === 200) {
      const { overTime, answerNum } = data; // overTime为抢答剩余时间，answerNum为已抢答人数
    } else {
      console.log(`获取抢答状态失败，失败原因:${message}`)
    }
  }
)
```

5、讲师端发送开始抢答事件

讲师在推流端点击抢答图标，弹出窗口，输入时间，点击开始抢答。

- START_ANSWER_QUICKLY

参数说明

参数名称	参数说明	参数类型
EVENT	事件名称，写死 START_ANSWER_QUICKLY	string
limitTime	抢答的时间	number
sessionId	该场直播的sessionId（如果只是测试，可以传入随机字符串）	string

示例

```
socket.emit('message', JSON.stringify({
  EVENT: 'START_ANSWER_QUICKLY',
  limitTime: 30,
  sessionId: 'test'
}), function(d) {
  const { code, message } = JSON.parse(d);
  if (code === 200) {
    console.log('发起抢答成功');
  } else {
    console.log(`发起抢答失败，失败原因:${message}`);
  }
});
```

6、结束抢答

讲师发起抢答后，页面有个停止抢答按钮，讲师可以主动提前停止抢答

- STOP_ANSWER_QUICKLY

参数说明

参数名称	参数说明	参数类型
EVENT	事件名称，写死 STOP_ANSWER_QUICKLY	string
quicklyId	抢答id	string

示例

```
socket.emit('message', JSON.stringify({
  EVENT: 'STOP_ANSWER_QUICKLY',
  quicklyId: 'test',
}), function(d) {
  const { code, message } = JSON.parse(d);
  if (code === 200) {
    console.log('抢答停止成功成功');
  } else {
    console.log(`抢答结束失败, 失败原因:${message}`);
  }
});
```