

Websocket platform message transfer specification

Websocket 平台消息传输规范

制 订 日 期：2020 年 08 月 09 日

最后修订日期：2020 年 08 月 20 日

目录

- Websocket platform message transfer specification..... 1
- Websocket 平台消息传输规范..... 1
 - 描述..... 3
 - 修订记录..... 3
 - 消息加密..... 4
 - 认证授权..... 4
 - 心跳..... 4
 - 连接流程..... 4
 - MsgTypeEnum 消息类型枚举..... 5
 - DataTypeEnum 数据类型枚举(业务)..... 5
 - 消息传输-model-TextMsg..... 6
 - 消息传输-model- TextData..... 6
 - 消息传输-model- Member..... 6
 - 消息传输-model-Room..... 7
 - 消息传输-群聊(datatype=1)-JoinGroupChat..... 7
 - 传输示例-基础-骨架..... 7
 - 传输示例-基础-心跳..... 8
 - 传输示例-群聊(datatype=1)-加入群聊..... 8
 - 响应示例-基础-骨架..... 8
 - 响应示例-基础-心跳..... 9
 - 响应示例-群聊(datatype=1)-加入群聊..... 9
 - 响应示例-群聊(datatype=2)-离开群聊..... 10
 - 响应示例-连接被拒绝-重复加入房间..... 11

描述

略

修订记录

修订人	修订类型	修订日期
王树明	新增	2020 年 08 月 09 日
修订原因	创建文档	
修订详情描述	创建文档	

修订人	修订类型	修订日期
王树明	新增	2020 年 08 月 14 日
修订原因	增加传输示例，便于使用者更加理解	
修订详情描述	<div>1. 传输示例-基础-骨架</div> <div>2. 传输示例-基础-心跳</div> <div>3. 传输示例-群聊(datatype=1)-加入群聊</div> <div>4. 增加 DataTypeEnum 值，2-离开群聊/房间</div> <div>5. 增加 Room 模型字段，newLeaveMember，新离开的成员</div> <div>6. 响应示例-群聊(datatype=2)-离开群聊</div>	

修订人	修订类型	修订日期
王树明	新增	2020 年 08 月 20 日
修订原因	增加响应示例，便于使用者更加理解	
修订详情描述	<div>1. 响应示例-连接被拒绝-重复加入房间</div>	

消息加密

由于消息内容可能会很长，且字节大小不确定，所以不会采用非对称加密算法。会采用对称加密算法，具体哪种还未选定(大概率会采用 AES)，先占位。

认证授权

目前服务器还未实行任何认证及授权流程，任何人都可以连接，实行后会具体描述，先占位。

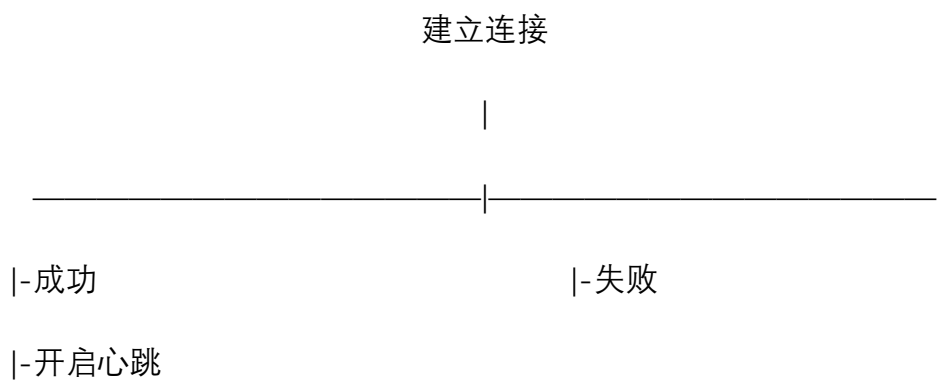
心跳

心跳，是即时通讯项目必不可少的一道（流程）工序。是目前为止最简单且有效的检验客户端是否存活的依据之一。心跳频率不宜过长，也不宜过短，过短会对服务端造成压力，过长会造成检测不及时。

本项目心跳频率为 2min(120s)，即客户端需每隔以上频率内就向服务端发送一个心跳包，包内容具体见：目录-传输示例-基础-心跳

需要注意的是：一定时间内服务端收不到心跳包，则认为该客户端已处于消亡状态，会将该客户端踢出并断开连接，节约服务器资源。

连接流程



|-发送消息

|-保持心跳

|-Error

|-关闭连接、关闭心跳

MsgTypeEnum 消息类型枚举

Type	Describe
1	客户端保持心跳
2	第一次(或重连)初始化连接
3	聊天消息
4	消息签收 (目前暂未使用到)
5	拉取好友 (目前暂未使用到)
...	...

DataTypeEnum 数据类型枚举(业务)

Type	Describe
1	加入群聊/房间
2	离开群聊/房间
...	...

消息传输-model-TextMsg

Field	Type	Description
msgType	Integer	消息类型，对应 MsgTypeEnum
textData	TextData	文本数据
extendFields	Object	扩展字段(备用)

消息传输-model- TextData

Field	Type	Description
dataType	Integer	内容类型，对应 DataTypeEnum
data	T	数据内容，此数据内容会随着 dataType 改变而改变，不同的 dataType 对应不同的 data

消息传输-model- Member

Field	Type	Description
id	String	Id，唯一标识
personName	String	姓名
personNum	String	编号

nickName	String	昵称
headImgUrl	String	头像

消息传输-model-Room

Field	Type	Description
number	String	房间编号
name	String	房间名称
desc	String	房间描述
owner	Member	房主
createTime	Date	创建日期
members	List<Member>	房间成员列表
newJoinMember	Member	新加入的成员
newLeaveMember	Member	新离开的成员

消息传输-群聊(datatype=1)-JoinGroupChat

Field	Type	Description
from	Member	发送人/加入人
roomNumber	String	群编号

传输示例-基础-骨架

```
1. var textMsg = {
2.     msgType : 此处省略,           //消息类型, 对应 MsgTypeEnum
3.     textData : {
4.         dataType: 此处省略,       //内容类型, 对应 DataTypeEnum
```

```

5.         data: {                                //数据内容，此数据内容会随着 dataType 改变而改变，不同
           的 dataType 对应不同的 data
6.         }
7.     }
8. };

```

传输示例-基础-心跳

```

1. var textMsg = {
2.     msgType : 1,                                //消息类型，对应 MsgTypeEnum，此处为心跳
3.     textData : {}
4. };

```

传输示例-群聊(datatype=1)-加入群聊

```

1. var textMsg = {
2.     msgType : 2,                                //第一次(或重连)初始化连接
3.     textData : {
4.         dataType: 1,                            //加入群聊
5.         data: {                                //数据内容，此数据内容会随着 dataType 改变而改变，不同的 dataType 对应不
           同的 data
6.             from: {id:"1jt5325h83rf5h9ue6jk8r3h8jk2", personName:"张三
           ", personNum:"XS00001", nickName:"老三", headImgUrl:"/abc/efg/h.jpg"},
7.             roomNumber: "E00001"
8.         }
9.     }
10. };

```

响应示例-基础-骨架

```

1. {
2.     "status":1,
3.     "msg":"Join successfully",
4.     "resultBody":{
5.         "msgType":3,
6.         "textData":{
7.             "dataType":1,
8.             "data":{

```



```
9.         }
10.     }
11. }
12. }
13. // 同样，传输内容分类型那响应内容也分类型，原因是便于客户端自己根据类型编写业务逻辑
```

响应示例-基础-心跳

```
1. // 心跳包只需客户端向服务端发送，告知服务端某个客户端还活着，不要踢出我，不要与我断开连接。
2. // 服务端不做任何处理，也不会有任何响应。
```

响应示例-群聊(datatype=1)-加入群聊

```
1. {
2.     "status":1,
3.     "msg":"Join successfully",
4.     "resultBody":{
5.         "msgType":3,
6.         "textData":{
7.             // 除了骨架外，以下消息是 Room 模型数据，即整个房间的相关属性，编号、名称、成员列表等
8.             "dataType":1,
9.             "data":{
10.                 "number":"E00001",
11.                 "name":"房间名称",
12.                 "desc":"房间描述",
13.                 "owner":{
14.                     "headImgUrl":"/abc/efg/h.jpg",
15.                     "id":"1jt5325h83rf5h9ue6jk8r3h8jk1",
16.                     "nickName":"老三",
17.                     "personName":"张三",
18.                     "personNum":"XS00001"
19.                 },
20.                 "createTime":1597368648105,
21.                 "members":[
22.                     {
23.                         "headImgUrl":"/abc/efg/h.jpg",
24.                         "id":"1jt5325h83rf5h9ue6jk8r3h8jk1",
25.                         "nickName":"老三",
26.                         "personName":"张三",
```

```

27.             "personNum": "XS00001"
28.         },
29.         {
30.             "headImgUrl": "/abc/efg/h.jpg",
31.             "id": "1jt5325h83rf5h9ue6jk8r3h8jk1",
32.             "nickName": "老三",
33.             "personName": "张三",
34.             "personNum": "XS00001"
35.         }
36.     ],
37.     "newJoinMember": {
38.         "headImgUrl": "/abc/efg/h.jpg",
39.         "id": "1jt5325h83rf5h9ue6jk8r3h8jk1",
40.         "nickName": "老三",
41.         "personName": "张三",
42.         "personNum": "XS00001"
43.     }
44. }
45. // 除了骨架外，以上消息是 Room 模型数据，即整个房间的相关属性，编号、名称、成员列表等
46. }
47. }
48. }

```

注：首次加入房间的响应消息中 TextData 内 Room 房间对象 `members` 拥有全部在线成员列表，之后的每次有人加入房间的响应消息中 TextData 内 Room 房间对象 `members` 为 `null` 仅拥有 `newJoinMember` 新加入的 Member 对象，需要前端自行组装至自己维护的在线列表内。

原因：考虑到消息包大小的问题，如果不区分，全部都直接返回整个房间的列表，数据量会根据房间的人数逐渐增大，考虑到大课堂的问题，还是不太可行，会有传输压力，甚至超过最大字节长度，导致断连。

响应示例-群聊(datatype=2)-离开群聊

```

1. {
2.     "status": 1,
3.     "msg": "Leave successfully",
4.     "resultBody": {
5.         "msgType": 3,
6.         "textData": {
7.             // 除了骨架外，以下消息是 Room 模型数据，即整个房间的相关属性，编号、名称、成员列表等
8.             "dataType": 2,
9.             "data": {
10.                 "number": "E00001",
11.                 "name": "房间名称",

```

```

12.         "desc": "房间描述",
13.         "createTime": 1597405290076,
14.         "newLeaveMember": {
15.             "headImgUrl": "/abc/efg/h.jpg",
16.             "id": "1jt5325h83rf5h9ue6jk8r3h8jk4",
17.             "nickName": "老三",
18.             "personName": "张三",
19.             "personNum": "XS00001"
20.         },
21.         "owner": {
22.             "headImgUrl": "/abc/efg/h.jpg",
23.             "id": "1jt5325h83rf5h9ue6jk8r3h8jk2",
24.             "nickName": "老三",
25.             "personName": "张三",
26.             "personNum": "XS00001"
27.         }
28.     }
29.     // 除了骨架外，以上消息是 Room 模型数据，即整个房间的相关属性，编号、名称、成员列表等
30. }
31. }
32. }

```

注：每次有人离开房间的响应消息中 TextData 内 Room 房间对象 `members` 为 `null`，仅拥有 `newLeaveMember` 新离开的 Member 对象，需要前端自行组装至自己维护的在线列表内。

响应示例-连接被拒绝-重复加入房间

```

1. {
2.     "status": 0,
3.     "msg": "连接被拒，您已在其他页面或客户端加入该房间，请先将其退出！",
4.     "resultBody": ""
5. }
6. // 解释：重复加入某个房间，不会导致之前已加入的房间被强制退出。相反，会提示加入房间失败。
7. // 如需在当前位置（页面、客户端）加入房间，需将之前加入的房间正常退出。

```