总线接口定义有两种，一实一虚，实的规划为实际通信的内容，虚的规划为对实的理化解析，像虚拟机一样，支撑对实际程序的运行。

虚的规划，目的是为了实现通信双方的自动应答。

先实后虚，本文先定义实的总线接口，从实际需求出发，逐步增加功能定义。

# 客户端和websocket server的通信协议

\* JSON格式

\* C:客户端 S:服务端

\* 当C向S发消息时，callback通常是一个随机的数字，S回应C时只需原样返回callback就可以了。

只有当S主动向C发消息时使用特定的callback值，且S不需要等待C的回应。

---------------------------------------------------------------------------------------------

## HTML5连接注册

**C -> S:**

'{

"type": "register",

"content": {

"data": {

"userId": "U000000001",

"username": "lucky",

"clientId": "H000000001",

"clientName": "clientName",

"clientCategory": "category"

},

"callback": 45

}

}'

**S -> C:**

'{

"callback": 42,

"data": "true",

"message": "Text"

}'

统一叫UA+两位类別+000001吧，名叫什么适配器(照明是LT，楼宇叫BA);

客户端是A+两位+7位序号;

\*说明：适配器连接服务器成功后，向服务器注册适配器信息

\*本例：向服务器注册适配器的ID、名称、URL，加上服务器自动识别的适配器IP和端口号，形成关联

## HTML5向服务器发送数据

**C -> S:**

'{

"type": "clientSend",

"pointer": [

"clientId1",

"clientId2"

],

"content": {

"sourcetype": "html5",

"data": [

{

"name": "点名1",

"value": "值"

},

{

"name": "点名1",

"value": "值"

}

],

"callback": 21

}

}'

**S -> C:**

'{

"callback": 21,

"data": "true",

"message": "text"

}'

\*说明：本接口为适配器向服务器发送变化了的数据，"适配器ID"本身也设置为管道名，向订阅该适配器ID的客户端发送变化了的点值。

\*本例：点名1、点名2变化了的值发送到服务器，服务器对该适配器ID的相关点值进行更新

## 服务器向HTML5发送数据(控制)

**S-> C:**

'{

"type": "serverSend",

"pointer": [

"clientId1",

"clientId2"

],

"content": {

"desttype": "html5",

"data": [

{

"name": "点名1",

"value": "值"

},

{

"name": "点名1",

"value": "值"

}

],

"callback": 21

},

"callbackid": 22

}'

**C -> S:**

'{

"callback": 21,

"data": "true",

"message": "Text"

}'

\*说明：本接口为服务器向客户端发送变化了的数据。

\*本例：如前端把输出点名1、点名2变化了的值发送到服务器，服务器作为中转，向适配器ID输出控制值。

## HTML5订阅实时数据（订阅请求）

**C -> S:**

'{

"type": "rtsub",

"pointer": [

"clientId1",

"clientId2"

],

"content": {

"data": {

"clientId": "客户端ID",

"subscribeTopic": "控制器",

"fileCategory": "文件类型",

"subscribeSubitem": [

"adapter1.controller1",

"adapter1.controller2",

"adapter2.controller1",

"adapter2.controller2"

]

},

"callback": 21

}

}'

**S -> C:**

'{

"callback": 21,

"data": "true",

"message": "Text"

}'

\*说明：本接口为批量处理接口

\*举例：h5前端的编号是H0BA0001，有2个空调的数据要展示，它订阅的主题是：网络控制器，文件类型是：元素文本，文本串是：<网络控制器1><网络控制器2>。元素文本指，后面都是一个个元素，每个元素用<>包起来。

比如用户订阅的是“某红外报警”的主题，文本类型就是记录文本，就采用json格式吧，如报警源、报警时间、报警内容等等。

文件类型的种类：元素文本、记录文本、二进制、音频、音视频。

订阅时，文件类型是元素文本的，这些元素文本就是订阅子项。其它类型没有订阅子项。

订阅数据库的设计：客户端编号、订阅主题、文件类型、订阅子项

订阅主题中，适配器、网络控制器、监控点，这三种主题定死了，其它主题自定义。

报警接口

WebSocketServer-->Html （报警时间，报警系统，报警级别，报警原因）

{

"type":"Alarm",

"Alarmtime": "2018:04:13:16:30:41",

"Level": "1",

"System": "BA",

"Data": {

"pointid": "100000000000000209",

"pointname": "风机过滤网",

"alarmvalue": "1",

"alarmcontent": "风机过滤网报警"

}

}