信令通道协议

目录

[1：头文件 2](#_Toc519774364)

[2：tcp传输的数据头定义 2](#_Toc519774365)

[3：json定义 3](#_Toc519774366)

[4：信令组成 4](#_Toc519774367)

[1.1 Tcp链路使用 数据头+json数据的方式 4](#_Toc519774368)

[1.2 Websocket使用json数据的方式 4](#_Toc519774369)

[5 ：TCP设备与服务器交互 4](#_Toc519774370)

[5.1交互时序图 4](#_Toc519774371)

[5.2请求session： 4](#_Toc519774372)

[5.3登陆： 5](#_Toc519774373)

[5.4心跳 5](#_Toc519774374)

[5.5 上传告警 5](#_Toc519774375)

[5.6 处理用户发送的操作指令 5](#_Toc519774376)

[5.7代码参考 6](#_Toc519774377)

[6：TCP客户端与服务器交互 6](#_Toc519774378)

[6.1交互时序图 6](#_Toc519774379)

[6.2请求session： 6](#_Toc519774380)

[6.3登陆： 6](#_Toc519774381)

[6.4心跳 7](#_Toc519774382)

[6.5 发送操作指令 7](#_Toc519774383)

[6.6 接收告警 7](#_Toc519774384)

[6.7代码参考 7](#_Toc519774385)

[7：WEBSOCKET客户端与服务器交互 8](#_Toc519774386)

[7.1 特别注意： 8](#_Toc519774387)

[7.2 交互时序图 8](#_Toc519774388)

[7.3 登陆 8](#_Toc519774389)

[7.4 心跳 9](#_Toc519774390)

[7.5 获取设备 9](#_Toc519774391)

[7.6 发送操作指令 10](#_Toc519774392)

[7.7 接收告警 10](#_Toc519774393)

[7.8代码参考 11](#_Toc519774394)

# 1：头文件

XDefineSignalingChannelStruct.h

Json 使用的jsoncpp

a\_signaling\_channel\_json\_reader\_builder.h

经典服务器与物联网服务器在于端口号不同和设备id不同

经典服务器：

端口号：

设备id：纯数字

物联网服务器：

端口号：

设备id：数字与字母混合

# 2：tcp传输的数据头定义

enum signaling\_channel\_cmd\_def

{

SINGNALING\_CHANNEL\_CMD\_REDIRECTION\_DEVICE = 0,

SINGNALING\_CHANNEL\_CMD\_REDIRECTION\_USER,

SINGNALING\_CHANNEL\_CMD\_REDIRECTION\_CLASSIC\_GET\_DEVICES,

SINGNALING\_CHANNEL\_CMD\_LOGIN = 20, //登陆

SINGNALING\_CHANNEL\_CMD\_LOGIN\_NEED\_SESSION, //key s -> c

SINGNALING\_CHANNEL\_CMD\_HEARTBEAT, //心跳

SINGNALING\_CHANNEL\_CMD\_LOGOUT, //注销

//

SINGNALING\_CHANNEL\_CMD\_IOT\_DEVICE, //IOT的添加修改删除

SINGNALING\_CHANNEL\_CMD\_IOT\_DEVICE\_SENSORS, //IOT的传感器

SINGNALING\_CHANNEL\_CMD\_TRANS, //透明传输

SINGNALING\_CHANNEL\_CMD\_GPS,

SINGNALING\_CHANNEL\_CMD\_ALARM,

};

enum signaling\_channel\_cmd\_result

{

SINGNALING\_CHANNEL\_RESULT\_OK = 0,

SINGNALING\_CHANNEL\_RESULT\_ERROR

};

struct signaling\_channel\_head\_t

{

uint32\_t cmd; //signaling\_channel\_cmd\_def

uint32\_t result; //signaling\_channel\_cmd\_result

uint32\_t session; //

uint32\_t datalen; //数据长度

};

struct signaling\_channel\_device\_login\_t

{

char userdes[128];

char uuiddes[128];

};

struct signaling\_channel\_user\_login\_t

{

char userdes[128];

char pwddes[128];

};

# 3：json定义

“header”节点：必须，必须的关键字：command，result，deviceuuid，msguuid

"command" : 命令：已定义的命令有"getdevices"，"adddevice"，"modifydevice"，"deletedevice"，"ptz"，"setparam"，"getparams"，"getsensors"，“trans”

"result" : 操作结果

"deviceuuid" : 设备id

"msguuid" : 消息ID

"info"节点，可选，"info"内的项目可选

例如：

{

"header" :

{

"command" : "cmd",

"result" : 0,

"deviceuuid" : "WWDSDJ9982HH88",

"msguuid" : "777777"

},

"info": {

"pwd": "222",

"user": "111",

}

}

# 4：信令组成

## 1.1 Tcp链路使用 数据头+json数据的方式

signaling\_channel\_head\_t + json\_data

## 1.2 Websocket使用json数据的方式

服务器会校验json头并转换成合适的数据包与设备或者客户端交互

# 5 ：TCP设备与服务器交互

## 5.1交互时序图

## 5.2请求session：

发送：signaling\_channel\_head\_t

cmd： SINGNALING\_CHANNEL\_CMD\_LOGIN\_NEED\_SESSION

服务器返回：

返回：signaling\_channel\_head\_t

cmd：SINGNALING\_CHANNEL\_CMD\_LOGIN\_NEED\_SESSION

session：服务器填充了session

## 5.3登陆：

发送：signaling\_channel\_head\_t + signaling\_channel\_device\_login\_t

signaling\_channel\_head\_t .cmd：SINGNALING\_CHANNEL\_CMD\_LOGIN

signaling\_channel\_user\_login\_t. uuiddes:使用des混合加密设备ID与服务器返回的session

服务器返回：

返回：signaling\_channel\_head\_t

cmd：SINGNALING\_CHANNEL\_CMD\_LOGIN

result：SINGNALING\_CHANNEL\_RESULT\_OK

## 5.4心跳

发送：signaling\_channel\_head\_t

cmd： SINGNALING\_CHANNEL\_CMD\_HEARTBEAT

服务器返回：

返回：signaling\_channel\_head\_t

cmd：SINGNALING\_CHANNEL\_CMD\_HEARTBEAT

## 5.5 上传告警

发送：signaling\_channel\_head\_t+ json数据

cmd： SINGNALING\_CHANNEL\_CMD\_ALARM

json数据：

服务器返回：

暂时不关心服务器的返回。

## 5.6 处理用户发送的操作指令

接收：signaling\_channel\_head\_t + json数据

cmd： SINGNALING\_CHANNEL\_CMD\_TRANS

json数据：

返回操作结果：signaling\_channel\_head\_t + json数据

cmd： SINGNALING\_CHANNEL\_CMD\_TRANS

json数据：请注意：json的msguuid必须填充为接受的json指令的msguuid，客户端用于区分是否自己发布的指令。

## 5.7代码参考

\_IncludeDevice 下的 XDeviceSignaling.cpp

# 6：TCP客户端与服务器交互

## 6.1交互时序图

## 6.2请求session：

发送：signaling\_channel\_head\_t

cmd： SINGNALING\_CHANNEL\_CMD\_LOGIN\_NEED\_SESSION

服务器返回：

返回：signaling\_channel\_head\_t

cmd：SINGNALING\_CHANNEL\_CMD\_LOGIN\_NEED\_SESSION

session：服务器填充了session

## 6.3登陆：

发送：signaling\_channel\_head\_t + signaling\_channel\_user\_login\_t

signaling\_channel\_head\_t .cmd：SINGNALING\_CHANNEL\_CMD\_LOGIN

signaling\_channel\_user\_login\_t. userdes:使用des混合加密用户名与服务器返回的session

signaling\_channel\_user\_login\_t. pwddes:使用des混合加密密码与服务器返回的session

服务器返回：

返回：signaling\_channel\_head\_t

cmd：SINGNALING\_CHANNEL\_CMD\_LOGIN

result：SINGNALING\_CHANNEL\_RESULT\_OK

## 6.4心跳

发送：signaling\_channel\_head\_t

cmd： SINGNALING\_CHANNEL\_CMD\_HEARTBEAT

服务器返回：

返回：signaling\_channel\_head\_t

cmd：SINGNALING\_CHANNEL\_CMD\_HEARTBEAT

## 6.5 发送操作指令

发送：signaling\_channel\_head\_t + json数据

cmd： SINGNALING\_CHANNEL\_CMD\_TRANS

json数据：请注意，必须填充msguuid，使用唯一算法

返回：signaling\_channel\_head\_t + json数据

cmd： SINGNALING\_CHANNEL\_CMD\_TRANS

json数据：请注意：必须判读json的msguuid是否为自己填充的

## 6.6 接收告警

接收：signaling\_channel\_head\_t+ json数据

cmd： SINGNALING\_CHANNEL\_CMD\_ALARM

json数据：

发送：

不向服务器返回结果。

## 6.7代码参考

\_XInternetProjectClient\\_Include下的 XClientSignaling.cpp

# 7：WEBSOCKET客户端与服务器交互

## 7.1 特别注意：

请注意：请人工确保该账号下的设备都在同一个服务器上，否则请不要提供该服务给用户。

## 7.2 交互时序图

## 7.3 登陆

发送：

{

"header": {

"command": "login",

"result": 0,

"deviceuuid": "",

"msguuid": ""

},

"info": {

"user": "13800138000", 用户名

"pwd": "13800138000"密码

}

}

服务器返回：

{

"header": {

"command": "login",

"deviceuuid": "",

"msguuid": "",

"result": 0 成功返回0

}

}

## 7.4 心跳

发送

{

"header" :

{

"command" : "heartbeat",

"deviceuuid" : "",

"msguuid" : "",

"result" : 0

}

}

服务器返回

{

"header" :

{

"command" : "heartbeat",

"deviceuuid" : "",

"msguuid" : "",

"result" : 0

}

}

## 7.5 获取设备

发送

{

"header": {

"command": "getdevices",

"result": 0,

"deviceuuid": "",

"msguuid": ""

}

}

服务器返回

{

"header" :

{

"command" : "getdevices",

"deviceuuid" : "",

"msguuid" : "",

"result" : 0

},

"info" :

[

{

"name" : "3001",

"online" : 0,

"rtmpip" : "47.96.249.142",

"rtmpuuid" : "46d6a42a7ed84576a0489fafae814d5a",

"uuid" : "1"

}

]

}

## 7.6 发送操作指令

发送

服务器返回

{

"header": {

"command": "getdevices",

"result": 0,

"deviceuuid": "",

"msguuid": ""

}

}

## 7.7 接收告警

服务器返回

## 7.8代码参考

XInternetProjectClient\XWebSocketSite\classic