API of MoteChat

YPCloud Inc.

2018/04/24

Command Brief

- Init: Initial motechat
- PublishXrpc: Publish XRPC function
- IsolatedXrpc: Publish isolated XRPC function
- StartSession: Start session with device center
- EndSession: End session with device center
- CallXrpc: Call function of the other device by XRPC
- SendXmsg: Send message to the other device by Xmsg
- GetDeviceInfo: Get the information of my device
- SetDeviceInfo, Set the device information of my device
- Nearby: search nearby device
- on: Set event handler

Command

Init: Initial motechat

Input:
conf: the configuration object for init. { “AppName”:””, “IOC”:””, “DCenter”:””, “AppKey”:””, “UseWeb”:”” }
AppName: the name of motebus MMA
IOC: the MMA of IOC
DCenter: the MMA of device enter
AppKey: the key string of app
UseWeb: can be 'websocket', 'ajax', or ''
cb: callback({ErrCode, ErrMsg, result})

Example:
var conf = { “AppName”:””, “IOC”:””, “DCenter”:””, “AppKey”:””, “UseWeb”:”” }
conf.AppName = ‘myfunc’;
conf.DCenter = ‘dc@boss.ypcloud.com:6788’;
conf.AppKey = ‘YfgEeop5’;

Doc:

```
var mchat = require('motechat');
mchat.Init(conf, function(result){
        console.log('init result=%s', JSON.stringify(result));
}
```

PublishXrpc: Publish XRPC function

Input:

app: the name of function

func: the user function entry which is published

cb, callback({ErrCode, ErrMsg})

Example:

```
var XrpcMcService = {
        "echo": function(head, body){
                console.log("xrpc echo: head=%s", JSON.stringify(head));
                if ( typeof body == 'object')
                        sbody = JSON.stringify(body);
                else
                        sbody = body;
                console.log("xrpc echo: body=%s", sbody);
                return {"echo":body};
        }
}
mchat.PublishXrpc( XrpcMcService, function(result){
   console.log('motechat publish: result=%s', JSON.stringify(result));
});
```

IsolatedXrpc: Publish isolated XRPC function

Input:

app: the name of function

func: the user function entry which is published

cb, callback({ErrCode, ErrMsg})

Example:

```
var XrpcMcSecService = {
    "echo": function(head, body){
        console.log("xrpc echo: head=%s", JSON.stringify(head));
        if ( typeof body == 'object')
            sbody = JSON.stringify(body);
        else
            sbody = body;
        console.log("xrpc echo: body=%s", sbody);
        return {"echo":body};
    }
}
mchat.IsolatedXrpc( XrpcMcSecService, function(result){
  console.log('motechat isolated: result=%s', JSON.stringify(result));
});
```

StartSession: Start session with device center

Input:

data: the information for session, { “EiToken”:””, “SToken”:”” }

EiToken: device token

SToken: app token

cb: callback({ErrCode, ErrMsg, result})

Example:

```
var mydev = {“EiToken”:”8dilCCKj”,”SToken”:” baTi52uE”};
mchat.StartSession(mydev, function(result){
  console.log('StartSession result=%s', JSON.stringify(result));
});
```

Note: At first time of the device, EiToken and SToken is empty.

EndSession: End session with device center

Input:

data: the information for session, { “SToken”:”” }

SToken: app token

cb: callback({ErrCode, ErrMsg})

Example:

```
var mydev = {”SToken”:” baTi52uE”};
mchat.EndSession(mydev, function(result){
  console.log('EndSession result=%s', JSON.stringify(result));
});
```

CallXrpc: Call function of the other device by XRPC

Input:

xrpc: xrpc control object, { “SToken”:””, “Target”:””, ”Func”:””, ”Data”:{} }

SToken: app token

Target: the target name of function

Func: the function name

Data: the data object for function

cb: callback({ErrCode, ErrMsg}) or callback(reply)

Example:

```
var target = ‘myEi’;
var func = ‘echo’;
var data = {“time”:”2018/4/24 10:12:08”};
var xrpc =
{"SToken":mydev.SToken,"Target":target,"Func":func,"Data":data};
mchat.CallXrpc( xrpc, function(reply){
  console.log('CallSession reply=%s', JSON.stringify(reply));
});
```

SendXmsg: Send message to the other device by Xmsg

Input:

xmsg: xmsg control object, { “SToken”:””, “From”:””, “Target”:””,”Data”:{},
“WaitReply”: 0 }

SToken: app token

From: DDN of source device

Target: can be DDN, EiName, EiType or EiTag of destination device

Data: the data which want to be sent
WaitReply: The wait time of reply, by sec.
cb: callback({ErrCode,ErrMsg}) or callback(reply)

Example:

```
var target = 'myEi';
var data = {"message":"Hello World"};
var ddn = GetSocketAttr('ddn', socket.id);
var stoken = GetSocketAttr('stoken', socket.id);
var xmsgctl =
{"SToken":stoken,"From":ddn,"Target":target,"Data":data,"WaitReply":12};
mochat.SendXmsg(xmsgctl, function(reply){
  console.log('sendxmsg reply=%s', JSON.stringify(reply));
});
```

GetDeviceInfo: Get the information of my device

Input:
data: the input data object, { "SToken":"" }
SToken: app token
cb: callback({ErrCode, ErrMsg}) or callback(reply)

Example:

```
var data = {"SToken":mydev.SToken};
mchat.GetDeviceInfo(data, function(result){
  console.log('GetDeviceInfo result=%s', result);
});
```

SetDeviceInfo, Set the device information of my device

Input:
data: input data object, { "SToken":"", "EdgeInfo":{} }
SToken: app token
EdgeInfo: {"EiName":"","EiType":"","EiTag":"","EiLoc":""}
cb: callback({ErrCode, ErrMsg}) or callback(reply)

Example:

```
var info = {"EiName":"myEi","EiType":".ei","EiTag":"#my","EiLoc":""};
var data = {"SToken":mydev.SToken,"EdgeInfo":info};
mchat.SetDeviceInfo(data, function(result){
    console.log('SetDeviceInfo result=%s', result);
});
```

Nearby: search nearby device

Input:

data: input data object, { "SToken":"" }

SToken: app token

cb: callback({ErrCode, ErrMsg}) or callback(reply)

Example:

```
var data = {"SToken":mydev.SToken};
mchat.Nearby(data, function(result){
    console.log('Nearby result=%s', result);
});
```

on: Set event handler

Input:

stype: "message" is for getxmsg, "state" is for state changed

cb: the user routine entry

Output:

return is boolean (true or false)

Example:

```
var InmsgRcve = function(ch, head, from, to, msgtype, data){
    console.log('InmsgRcve: channel=%s, from=%s, to=%s, msgtype=%s,
    data=%s', ch, JSON.stringify(from), to, msgtype, JSON.stringify(data));
}
Var InState = function(state){
```

```
    console.log('InState=%s', state);
}
mochat.on('message',InmsgRcve);
mochat.on('state', InState);
```