YPCLOUD

API of MoteChat

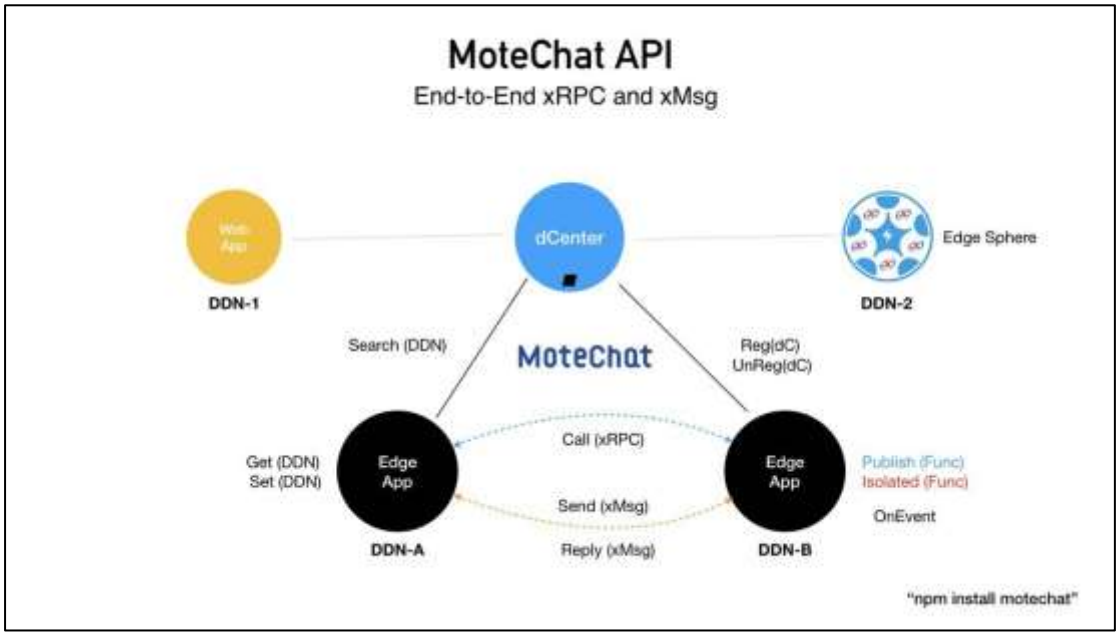
YPCloud Inc.

Last Updated : 2018/05/21

MoteMsg Cloud

Summary

System Diagram



List of API

Command	Description
Open	Open motechat
Close	Close motechat
Publish	Publish function
Isolated	Publish isolated function
Reg	Register to device center
UnReg	Un-register from device center
Call	Call function of another device
Send	Send message to another device
Get	Get the information of my device
Set	Set the device information of my device
Search	Search nearby device
OnEvent	Set event handler

Function List

Command	Function
Open	mChat.Open()
Close	mChat.Close()
Publish	mChat.Publish()
Isolated	mChat.Isolated()
Reg	mChat.Reg()
UnReg	mChat.Unreg()
Call	mChat.Call()
Send	mChat.Send()
Get	mChat.Get()
Set	mChat.Set()
Search	mChat.Search()
OnEvent	mChat.OnEvent()

Command

Open

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 center

UCenter: the MMA of user center

AppKey: the key string of app

UseWeb: can be 'websocket', 'ajax', or ''

reg: the information of register (option, the info of reg to DC)

EiToken: device token

SToken: app token

WIP: WAN IP

LIP: LAN IP

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

Example 1:

```
var conf = { "AppName":"", "IOC":"", "DCenter":"","UCenter":"", "AppKey":"", "UseWeb":"" }
conf.AppName = 'myfunc';
conf.DCenter = 'dc@boss.ypcloud.com:6788';
conf.AppKey = 'YfgEeop5';
var mChat = require('motechat');
mChat.Open(conf, function(result){
        console.log('init result=%s', JSON.stringify(result));
}
```

Example 2: reg to DC directly

```
var conf = { "AppName":"", "IOC":"", "DCenter":"", "AppKey":"", "UseWeb":"" }
conf.AppName = 'myfunc';
conf.DCenter = 'dc@boss.ypcloud.com:6788';
conf.AppKey = 'YfgEeop5';
var reginfo = {"EiToken":"8dilCCKj","SToken":"baTi52uE","WIP":"","LIP":""};
```

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

Close

Input:

cb: callback({ErrCode, ErrMsg })

Publish

Input:

app: the name of name

func: the user function entry which is published

cb: callback({ErrCode, ErrMsg})

Example:

```
var app = 'motechat';
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.Publish( app, XrpcMcService, function(result){
   console.log('motechat publish: result=%s', JSON.stringify(result));
});
```

Isolated

Input:

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.Isolated( XrpcMcSecService, function(result){
    console.log('motechat isolated: result=%s', JSON.stringify(result));
});
```

Reg

Input:

data: the information for registration, { "EiToken":"","SToken":"","WIP":""}

EiToken: device token

SToken: app token

WIP: WAN ip (empty means the same as dc)

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

Example:

```
var mydev = {"EiToken":"8dilCCKj","SToken":"baTi52uE","WIP":""};
mChat.Reg(mydev, function(result){
    console.log('StartSession result=%s', JSON.stringify(result));
});
```

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

UnReg

Input:

data: the information for registration, { "SToken":"" }

SToken: app token

cb: callback({ErrCode, ErrMsg})

Example:

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

Call

Input:

xrpc: xrpc control object, { "SToken":"", "To":"", "Topic":"", "Func":"", "Data":{},"SendTimeout": 6,"WaitReply": 12 }

SToken: app token

Topic: topic of app

To: the target name, DDN or topic address of function

Func: the function name

Data: the data object for function

SendTimeout: Integer, Timeout of send message, by sec.

WaitReply: Integer, The wait time of reply, by sec.

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

reply:

{"IN":{"From":{"DDN":"","Name":"","Type":"","Uid":"","Topic":""},"To":{"DDN":"","Name":"","Type":"","Topic":""},"State":{"ErrCode":0,"ErrMsg":"OK","By":""}},"Reply":{"ErrCode":0,"ErrMsg":"OK"}}

Example 1:

```
var to = 'myEi';
var func = 'echo';
var data = {"time":"2018/4/24 10:12:08"};
var t1 = 6;
var t2 = 12;
var xrpc = {"SToken":mydev.SToken, "To":to,"Func":func,"Data":data,
```

```
"SendTimeout":t1, "WaitReply":t2};
mChat.Call( xrpc, function(reply){
   console.log('CallSession reply=%s', JSON.stringify(reply));
});
```

Example 2:

```
var topic = 'steven.bot';
var to = 'einodered';
var func = 'echo';
var data = {"time":"2018/4/24 10:12:08"};
var t1 = 6;
var t2 = 12;
var xrpc = {"SToken":mydev.SToken,
"Topic":topic ,"To":to,"Func":func,"Data":data, "SendTimeout":t1,
"WaitReply":t2};
mChat.Call( xrpc, function(reply){
console.log('CallSession reply=%s', JSON.stringify(reply));
});
```

Example 3:

```
var stopic = 'steven.bot';
var target = 'einodered';
var dtopic = 'sally.bot';
var to = {"Target":target,"Topic":dtopic};
var data = {"message":"Hello World"};
var t1 = 6;
var t2 = 12;
var xrpc = {"SToken":mydev.SToken,
"Topic":topic ,"To":to,"Func":func,"Data":data, "SendTimeout":t1,
"WaitReply":t2};
mChat.Call( xrpc, function(reply){
console.log('CallSession reply=%s', JSON.stringify(reply));
});
```

Note:

Topic address:

```
{"Target":"(target)","Topic":"(topic)"}
```

{"Target":"(target)","Topic":{(topic)}}
{"DDN":"(DDN)","Topic":"(topic)"}
{"DDN":"(DDN)","Topic":{(topic)}}

Send

Input:

xmsg: xmsg control object, { "SToken":"", "Topic":"", "To":"","Data":{},
"SendTimeout": 6,"WaitReply": 12 }

SToken: app token

Topic: the app topic

To: can be DDN, EiName, EiTag or Topic Address of destination device

Data: the data which want to be sent

SendTimeout: Integer, Timeout of send message, by sec.

WaitReply: Integer, The wait time of reply, by sec.

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

reply:

{"IN":{"From":{"DDN":"","Name":"","Type":"","Uid":"","Topic":""},"To":{"DDN":"","Name":"","Type":"","Topic":""},"State":{"ErrCode":0,"ErrMsg":"OK","By":""}},"Reply":{"ErrCode":0,"ErrMsg":"OK"}}

Example 1:

```
var stoken = mydev.SToken;
var to = 'myEi';
var data = {"message":"Hello World"};
var t1 = 6;
var t2 = 12;
var xmsgctl = {"SToken":stoken, "To":to,"Data":data,
"SendTimeout":t1,"WaitReply":t2};
mChat.Send(xmsgctl, function(reply){
    console.log('sendxmsg reply=%s', JSON.stringify(reply));
});
```

Example 2:

```
var stoken = mydev.SToken;
```

```
var topic = 'steven.bot';
var to = 'myEi';
var data = {"message":"Hello World"};
var t1 = 6;
var t2 = 12;
var xmsgctl = {"SToken":stoken,"Topic":topic,"To":to,"Data":data,
"SendTimeout":t1,"WaitReply":t2};
mChat.Send(xmsgctl, function(reply){
    console.log('sendxmsg reply=%s', JSON.stringify(reply));
});
```

Example 3:

```
var stoken = mydev.SToken;
var stopic = 'steven.bot';
var target = 'einodered';
var dtopic = 'sally.bot';
var to = {"Target":target,"Topic":dtopic};
var data = {"message":"Hello World"};
var t1 = 6;
var t2 = 12;
var xmsgctl = {"SToken":stoken,"Topic":stopic,"To":to,"Data":data,
"SendTimeout":t1,"WaitReply":t2};
mChat.Send(xmsgctl, function(reply){
console.log('sendxmsg reply=%s', JSON.stringify(reply));
});
```

Note:

Topic address:

```
{"Target":"(target)","Topic":"(topic)"}
{"Target":"(target)","Topic":{(topic)}}
{"DDN":"(DDN)","Topic":"(topic)"}
{"DDN":"(DDN)","Topic":{(topic)}}
```

Get

Input:

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

Example:

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

Set

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.Set(data, function(result){
    console.log('SetDeviceInfo result=%s', result);
});
```

Search

Input:

data: input data object, { "SToken":"","Keyword":""}
SToken: app token
Keyword: keyword for search
cb: callback({ErrCode, ErrMsg}) or callback(reply)

Example:

```
var data = {"SToken":mydev.SToken, "Keyword":"#test"};
mChat.Search(data, function(result){
    console.log('Search result=%s', result);
});
```

OnEvent

Input:

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

cb(ch, inctl, data): the user routine entry

ch: channel (xrpc or xmsg)

inctl: IN control data {"From":"","To":""}

data: user data

Output:

return is boolean (true or false)

Example:

```
var InmsgRcve = function(ch, inctl, data, retcb){
    console.log('InmsgRcve: channel=%s, from=%s, to=%s, data=%s', ch,
    JSON.stringify(inctl.From), JSON.stringify(inctl.To), JSON.stringify(data));
    if ( typeof retcb == 'function') retcb({"ErrCode":0, "ErrMsg":"OK"})
}
Var InState = function(state){
    console.log('InState=%s', state);
}
mChat.OnEvent('message',InmsgRcve);
mChat.OnEvent('state', InState);
```

Error Code

No	Code	Message
1	0	OK
2	-10199	in error
3	-10101	in: open XRPC error
4	-10102	in: XRPC not open
5	-10103	in: motebus not open
6	-10104	in: send error
7	-10105	in: open XMsg error
8	-10106	in: invalid data
9	-10299	motechat error
10	-10201	motechat: no DC setting
11	-10202	motechat not open
12	-10203	motechat: invalid data
13	-10204	motechat: invalid stoken
14	-10205	motechat: no rcve function
15	-10306	motechat: no matched DDN
16	-10207	motechat: no DDN
17	-10399	wsocket error
18	-10301	wsocket: invalid data
19	-10302	wsocket: no matched DDN