



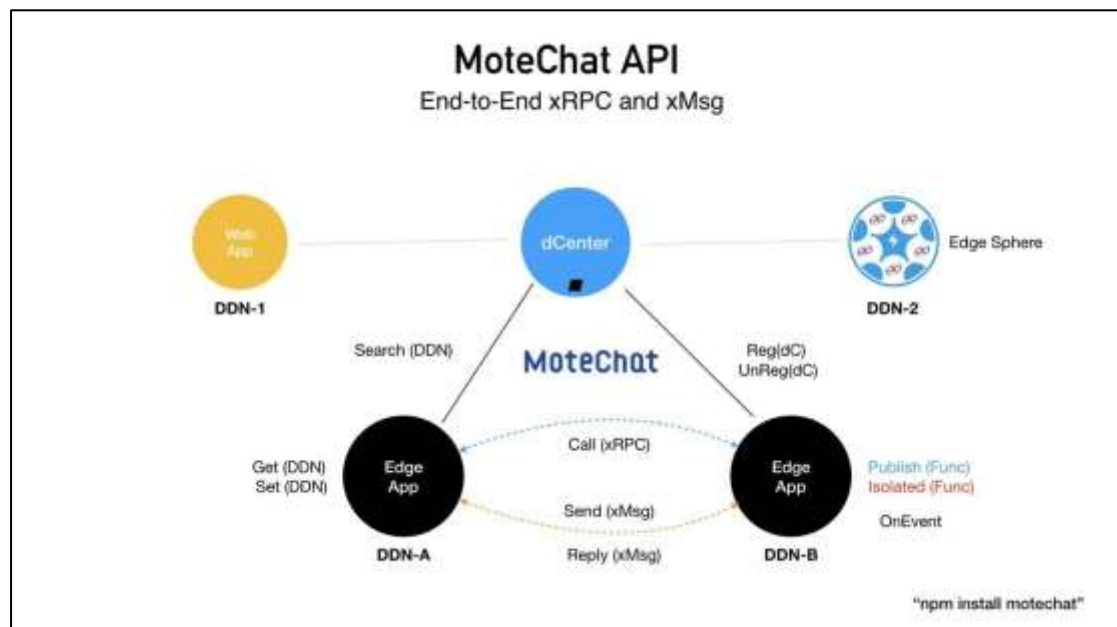
## API of MoteChat

YPCloud Inc.  
Copyright © 2018, 2019  
Last Updated : 2019/06/12

## Summary

System Diagram .....	3
List of API.....	4
Function List.....	5
Command.....	6
Open.....	6
Close.....	7
Publish.....	7
Isolated.....	8
Reg.....	9
UnReg.....	9
Call.....	9
Send .....	10
Get.....	11
Set .....	11
Search.....	12
Nearby.....	12
mbCall .....	13
mbSend .....	13
OnEvent.....	14
Error Code .....	15

# 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
mbCall	Call function of another device by motebus
mbSend	Send message to another device by motebus
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": "",  
"AppKey": "", "DCenter": "", "IOC": "", "MotebusGW": "", "Heartbeat": "",  
"UseWeb": "" }  
    AppName: the name of motebus MMA  
    AppKey: the key string of app  
    DCenter: the MMA of device center  
    IOC: the MMA of IOC  
    MotebusGW: the ip and port of motebus gateway, default is 127.0.0.1:6161  
    Heartbeat: enable heartbeat function, true or false, default is true  
    UseWeb: can be 'websocket' for web app or '' for app  
reg: the information of register for auto reg ( option, the info of reg to DC )  
    EiToken: device token  
    SToken: app token  
    WIP: WAN IP  
    LIP: LAN IP  
ei: the information of edge device for auto reg ( option, the info of ei )  
    EiName: device name  
    EiType: device type  
    EiTag: device tag  
    EiLoc: device location  
cb: callback( {ErrCode, ErrMsg, result} )

### Example 1:

```
var conf = { "AppName": "", "AppKey": "", "DCenter": "", "IOC": "",  
"MotebusGW": "127.0.0.1:6161", "UseWeb": "" }  
conf.AppName = 'myfunc';  
conf.DCenter = 'dc@dc.yplcloud.com:6789';  
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":""," "AppKey":""," "DCenter":""," "IOC":"","  
  "MotebusGW":"127.0.0.1:6161", "UseWeb":"" }  
conf.AppName = 'myfunc';  
conf.DCenter = 'dc@dc.ypcloud.com:6789';  
conf.AppKey = 'YfgEeop5';  
var reginfo = {"EiToken":"8diICCKj","SToken":"baTi52uE","WIP":""," "LIP":""};  
var mChat = require('motechat');  
mChat.Open(conf, reginfo, function(result){  
  console.log('init result=%s', JSON.stringify(result));  
})
```

### Example 3: reg to DC and set EI info directly

```
var conf = { "AppName":""," "AppKey":""," "DCenter":""," "IOC":"","  
  "MotebusGW":"127.0.0.1:6161", "UseWeb":"" }  
conf.AppName = 'myfunc';  
conf.DCenter = 'dc@dc.ypcloud.com:6789';  
conf.AppKey = 'YfgEeop5';  
var ei = {"EiName":"aifunc","EiType":".func","EiTag":"#ai","EiLoc":""}  
var reginfo = {"EiToken":"8diICCKj","SToken":"baTi52uE",  
  "WIP":""," "LIP":""," "EiInfo":ei};  
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":"","DDN":"","Topic":"","Func":"","  
"Data":{},"SendTimeout": 6,"WaitReply": 12 }  
SToken: app token  
DDN: DDN of device  
Topic: topic of app

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

## Send

Input:

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

SToken: app token

DDN: DDN of device

Topic: the app topic

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 token = mydev.SToken;
var ddN = 'kvGuHVUy';
var topic = 'flow/msg';
var data = {"message": "Hello World"};
var t1 = 6;
var t2 = 12;
var xmsgctl = {"SToken": token, "DDN": ddN, "Topic": topic, "Data": data,
"SendTimeout": t1, "WaitReply": t2};
mChat.Send(xmsgctl, function(reply){
    console.log('sendxmsg reply=%s', JSON.stringify(reply));
});
```

## 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  
Local: search local [ true | false ], default is false  
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);
});
```

## Nearby

### Input:

data: input data object, { "SToken":""}  
SToken: app token  
Local: search local [ true | false ], default is false  
cb: callback( {ErrCode, ErrMsg} ) or callback(reply)

### Example:

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

## mbCall

### Input:

xrpc: xrpc control object, { "MMA":""," "Func":""," "Data":{},"SendTimeout":6,"WaitReply": 12 }

MMA: mma of target device

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)

### Example 1:

```
var mma = 'hello@192.168.1.10';
var func = 'echo';
var data = {"time":"2018/4/24 10:12:08"};
var t1 = 6;
var t2 = 12;
var xrpc = {"MMA":mma, "Func":func,"Data":data, "SendTimeout":t1,
"WaitReply":t2};
mChat.mbCall( xrpc, function(reply){
    console.log('mbCall reply=%s', JSON.stringify(reply));
});
```

## mbSend

### Input:

xmsg: xmsg control object, { "MMA":""," "Data":{},"SendTimeout":6,"WaitReply": 12 }

MMA: mma of target device

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(result)

### Example 1:

```

var mma = 'hello@192.168.1.10';
var data = {"degree":30};
var t1 = 6;
var t2 = 12;
var xmsg = {"MMA":mma, "Data":data, "SendTimeout":t1, "WaitReply":t2};
mChat.mbSend( xmsg, function(result){
    console.log('mbSend reply=%s', JSON.stringify(result));
});

```

## OnEvent

### Input:

stype: "message" is for getxmsg, "state" is for state changed  
 cb: callback handler

### Output:

return is boolean ( true or false )

### Example:

```

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

```

## 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 token
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 socket id