YPCLOUD

API of MoteChat

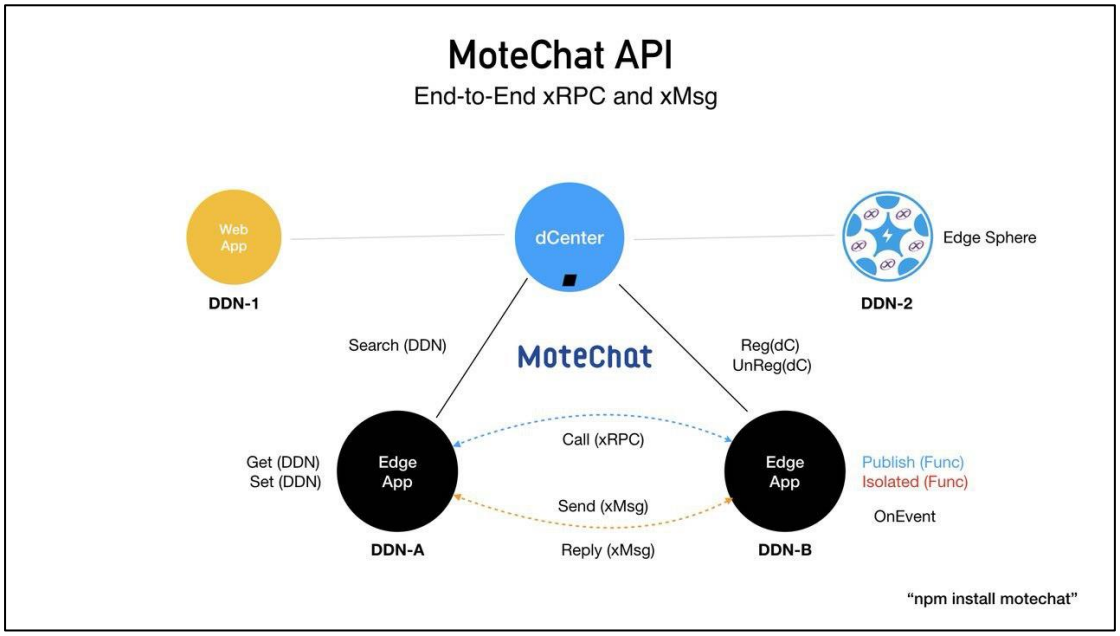
YPCloud Inc.

Last Updated : 2018/04/24

MoteMsg Cloud

Summary

System Diagram



List of API

Command	Description
Open	Open 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
Reply	

Function List

Command	Function
Open	mChat.Open()
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()
Reply	

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

Publish

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

Isolated

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

Reg

Input:

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

EiToken: device token

SToken: app token

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

Example:

```
var mydev = {"EiToken":"8dilCCKj","SToken":"baTi52uE"};
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":"", "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.Call( xrpc, function(reply){
    console.log('CallSession reply=%s', JSON.stringify(reply));
```

```
});
```

Send

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};
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

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: 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);
}
mChat.OnEvent('message',InmsgRcve);
mChat.OnEvent('state', InState);
```

Reply