# CC Api 接口文档

## 第一章：整体结构

### 1：接口调用方法

调用入口:

URL：http://cc.qiuhubang.com/model/action.json

### 2：常规请求数据结构（POST），每个接口需要如下参数

#### 1：提交参数

{

time : Integer,// 时间戳

version : Float, // 版本号

os : String, // 系统

udid : String, // 设备ID

token : String, // 认证码，暂时可以为空

auth\_session: String, // 会话(未登录传空字符串)

sign : String // 签名，暂时可以为空

params ：String //具体接口中所指定的业务参数 json字符串

}

#### 2：签名

调用接口时对所有POST数据（除sign字段）进行签名认证

A：将所有字段（除SIGN字段外）按键值字典序排列。

如将foo=1,bar=2,baz=3 排序为bar=2,baz=3,foo=1，参数名和参数值链接后，得到拼装字符串bar2baz3foo1

B：将应用对应密钥附在上一步字符串尾后MD5加密：

格式为：md5(bar2baz3foo1secret)

C：将上一步最终得到的字符串作为sign字段的值

#### 3：数据加密

对params进行3des加密，暂时不做

### 3：常规返回数据结构（JSON）

{

code : String, // 状态码(200 为成功,201为content空， 300以上为失败)

msg : String, // 消息(失败时为异常消息)

content : Json, // 返回内容(无为空字符串)

}

在提交一些数据时，服务器可能需要对数据进行校验，如某些条件校验不通过，则通过 content 下的 errors 来告知客户端哪些字段有问题。errors 为 Map<String,String> 类型，键是字段名称，值是错误消息。如果提交成功，并且content为空时，code返回201，不为空时code为200。

另，所有涉及到日期、时间的地方，请求和返回均使用时间戳。

## 第二章：用户api

### 1：个人注册接口

#### 第一步：输入邮箱，获取验证码：

**请求：**

协议：HTTP

方法：POST

地址：user/captcha.json

参数：{

email: 邮箱

}

**返回：**

类型：JSON

数据：content: {

captcha: 验证码

}

注释：由于没有做邮件模块，这个接口目前每次返回一个固定数字：523123

#### 第二步：输入第一步的验证码，第一步的邮箱，用户名，密码：

**请求：**

协议：HTTP

方法：POST

地址：user/register.json

参数：{

name: 名称,必选

email: 邮箱,必选 // UI上不需要用户输入，通过第一步带过来

password: 密码,必选

captcha: 验证码,必选 //如果UI上有专门输入验证码验证邮箱的步骤，程序带过来提交

}

**返回：**

类型：JSON

数据：content: {

auth\_session: 会话 // 每个接口都要上传的，没登录前空

}

### 2：登录接口

**请求：**

协议：HTTP

方法：POST

地址：user/login.json

参数：{

email: 邮箱,

password: 密码

}

**返回：**

类型：JSON

数据：content: {

auth\_session: 会话

}

### 3：登出接口

**请求：**

协议：HTTP

方法：POST

地址：user/logout.json

参数：{}

**返回：**

类型：JSON

数据：content: ""

### 4：关联状态接口

**请求：**

协议：HTTP

方法：POST

地址：user/openlogin.json

参数：{

open\_id: 关联ID,

open\_type: 关联类型(qq,weibo,weixin)

}

**返回：**

类型：JSON

数据：content: {

### 5：补全信息接口(同 [我的.修改信息](#_修改信息接口)接口)

**请求：**

协议：HTTP

方法：POST

地址：user/openregister.json

参数：{

**open\_id: 关联ID,**

**open\_type: 关联类型,**

name: 名称,必填

email: 邮箱,必填，后续验证码生效

password: 密码,

gender: 性别(0女1男),

title: 头衔,

company: 工作单位,

summary: 个人简介,

ico\_file: 头像(上传或者base64数据流都支持),

ico\_category：1/2 1表示文件，2表示字符流

ico\_type: 头像类型(png,jpg...)

}

**返回：**

类型：JSON

数据：content: {

errors: 错误列表,

auth\_session: 会话(如果成功)

}

**注：**以上请求数据中除auth\_session外都是选填，例如需要修改密码则name、email等不要发送。state: 0未关联,1已关联**(未关联则需要补全信息)**,

auth\_session: 会话(如果已关联)

}

### 6：补充个人信息

协议:http

方法：post

地址：user/persionalinfo.json

参数：

{

name:姓名 (必选项)

gender:性别

title: 职位

company:工作单位

summary:个人简介

}

返回：

Code：200,201，。。。

## 第三章：我的设置 相关接口

### 1：获取信息接口

**请求：**

协议：HTTP

方法：POST

地址：setinfo/myinfo.json

参数：{}

**返回：**

类型：JSON

数据：content: {

ico\_url: 头像URL,

name: 名称,

email: 邮箱,

gender: 性别(0女1男),

title: 头衔,

company: 工作单位,

summary: 个人简介

}

### 2：修改信息接口

**请求：**

协议：HTTP

方法：POST

地址：setinfo/setmyinfo.json

参数：{

name: 名称,

email: 邮箱,

password: 密码,

gender: 性别(0女1男),

title: 头衔,

company: 工作单位,

summary: 个人简介,

ico\_file: 头像(android上传,ios用base64),

ico\_type: 头像类型(png,jpg...)

}

**返回：**

类型：JSON

数据：content: {

errors: 错误列表

}

**注：**以上请求数据中除auth\_session外都是选填，例如需要修改密码name、email等不要发送。

### 3：所有消息开关接口

**请求：**

协议：HTTP

方法：POST

地址：setinfo/allnoticeswitch.json

参数：{

switch: 0关, 1开

}

**返回：**

类型：JSON

数据：content: ""

### 4：群组消息开关接口

**请求：**

协议：HTTP

方法：POST

地址：setinfo/groupnoticeswitch.json

参数：{

switch: 0关, 1开

}

**返回：**

类型：JSON

数据：content: ""

### 5：陌生人发起临时聊天开关接口

**请求：**

协议：HTTP

方法：POST

地址：setinfo/interimchatswitch.json

参数：{

switch: 0关, 1开

}

**返回：**

类型：JSON

数据：content: ""

### 6：版本升级接口

**请求：**

协议：HTTP

方法：POST

地址：setinfo/checkversion.json

参数：{}

**返回：**

类型：JSON

数据：content: {

state: 0没有新版本,1强制升级,2推荐升级

new: 最新版本号,

url: 下载URL

}

### 7：关于CC

Webview: http://cc.qiuhubang.com/setinfo/aboutus.html

### 8：服务条款

Webview: http://cc.qiuhubang.com setinfo/termsofservice.html

## 第四章：消息相关接口

## 第五章：人脉相关接口

## 第六章：mqtt使用UI展示