

口信 1680 企业移动通信平台

二次开发接口

2017-04-08

目录

第一章 企信通平台介绍	3
1.1. 简介	3
1.2. 缩略语清单	3
1.3. 修订时间	3
第二章 API 接口说明 (HTTP 接口)	4
2.1. 发送短信接口	4
2.2. 发送动态短信接口	5
2.3. 发送批量动态短信接口	7
2.4. 余额接口	9
2.5. 状态报告接口	10
2.6. 上行接口	10
第三章 API 接口说明 (CMPP 接口)	12
3.1. CMPP 概述	12
3.2. CMPP 连接返回	12
3.3. CMPP 提交返回	12
第四章 环境要求	14
4.1. 网络拓扑图	14
4.2. 环境要求	14
4.3. 流量控制	14
4.4. 并发控制	15
4.5. 其它	15
附录	16
密码 MD5 对照	16
短信提交 DEMO (JAVA 版)	16
短信提交 DEMO (SHELL 版)	16
短信提交 DEMO (C/C++版)	17
短信提交 DEMO (Objective-C 版)	18
短信提交 DEMO (C#版)	18
短信提交 DEMO (PHP 版)	19
短信提交 DEMO (PERL 版)	19
短信提交 DEMO (PYTHON 版)	19
短信提交 DEMO (RUBY 版)	20
短信提交 DEMO (LUA 版)	20

第一章 企信通平台介绍

1.1. 简介

企信通平台是我公司自主研发的系统，用于实现终端客户短信发送的管理平台，本文档为第三方平台使用 API 接口，与本企信通平台进行对接而完成。本接口采用传统 HTTP 方式和 CMPP 规范，出于数据安全性和数据容量的考虑，建议采用 CMPP 中方法连接本平台进行数据传输。

1.2. 缩略语清单

客户端：本文中特指短信平台企业终端客户端。

服务端：本文中特指短信平台企业网关服务端。

1.3. 修订时间

2011-01-03	完成基本内容
2013-08-10	完善 V5 版本说明
2015-01-12	完善 V7 版本说明
2016-12-30	完善 V8 版本说明，增加 CMPP 服务端等功能

第二章 API 接口说明（HTTP 接口）

2.1. 发送短信接口

提交地址：<http://api.1680.me/sendXSms.do>

send 接口提交方式：GET/POST

功能：发送短信

输入参数	描述
username	登陆名称（必填）
password	登陆密码（必填，明文或 32 位 MD5 密码）
mobile	手机号，多个手机号为用半角，分开，如 13899999999, 13688888888 (最多 1000 个，必填)
content	发送内容（必填）
dstime	定时时间，为空时表示立即发送（选填）
productid	产品 id (必填)
xh	扩展的小号（选填）

样例：

```
http://api.1680.me/sendXSms.do?username=kuaihuolin&password=888888&mobile=13911000000&content=中国&dstime=&productid=61341&xh=12345
```

其中：content 参数，采用下列方式处理：

```
// String content = "汉字";
```

```
// content = URLEncoder.encode(content, "UTF-8");
```

返回参数	描述
-1	登陆名称或者登陆密码不正确
0	失败
1, xxxxxxxx	1 代表发送短信成功, xxxxxxxx 代表消息编号
2	余额不够

3	黑词审核中
4	出现异常，人工处理中
5	提交频率太快
6	有效号码为空
7	短信内容为空
8,xxxxxxx	一级黑词,逗号后面为具体黑词内容
9	没有 url 提交权限
10	发送号码过多
11	产品 ID 异常
-11	用户信息异常
12	必填参数异常
13	产品余额异常
-14	IP 认证异常
-15	签名内容为空

2.2. 发送动态短信接口

提交地址：<http://api.1680.me/sendDIYSms.do>

send 接口提交方式：POST

功能：发送动态短信

输入参数	描述
username	登陆名称（必填）
password	登陆密码（必填，明文或 32 位 MD5 密码）
productid	产品 id(具体明细请联系客服)
params	动态短信的参数组。单次提交参数组的数目不能超过1000个。编码为UTF-8格式 每一组参数之间用英文“;”间隔 每一组参数内部用英文“,”间隔，最多六个参数，其中第一个参数为手机号码，第二个参

	<p>数为模板中第一个变量，第三个参数为模板中第二个变量，以此类推。</p> <p>* 例如：</p> <p>* 13800210000, 李先生, 100110198809030019, 北京市, 100010, 2013-01-01; 13500210000, 王先生, 100110199809030019, 北京市, 100010, 2013-01-15</p> <p>表明提交了两个变量组，其中手机号码分别为：13800210000 和 13500210000。模板变量分别为：“李先生, 100110198809030019, 北京市, 100010, 2013-01-01” 和 “王先生, 100110199809030019, 北京市, 100010, 2013-01-15”</p> <p>注：格式不符的参数，系统自动过滤掉。</p>
<p>content</p>	<p>短信模板。其中对应参数中的六个变量，第一个变量用“{PHONE}”来替代，第二个变量用“{A}”来替代，第三个变量用“{B}”来替代，第四个变量用“{C}”来替代，第五个变量用“{D}”来替代，第六个变量用“{E}”来替代。</p> <p>*例如：</p> <p>“{A}, 你好！，您的身份证号是{B}，位于{C}，邮编{D}请你于{E}日参加活动, 手机号是{PHONE}。”该短信中具有六个变量参数。编码为 UTF-8 格式。</p>
<p>dstime</p>	<p>定时时间，可以为空, 为空时表示立即发送</p>
<p>xh</p>	<p>扩展的小号（选填）</p>

样例：

http://api.1680.me/sendDIYSms.do

username=user1&password=123456¶ms=13800210000, 李先生, 100110198809030019, 北京市, 100010, 2013-01-01;13500210000, 王先生, 100110198809030019, 北京市, 100010, 2013-01-15&content={A}, 你好!, 您的身份证号是 {B}, 位于 {C}, 邮编 {D} 请你于 {E} 日参加活动, 手机号是 {PHONE}。&dstime=2010-12-30 17:48:21&productid=520411

注: 请使用 POST 方式提交, 避免提交失败

返回参数	描述
1	代表发送动态短信成功
0	失败
-1	登陆名称或者登陆密码不正确
-2	余额不够
-3	黑词审核中
-4	变量参数出现异常
-5	提交频率太快
-6	有效号码为空
-7	短信内容为空
-8	一级黑词
-9	没有 url 提交权限
-10	发送号码过多
-11	产品 ID 异常
-12	参数异常
-14	IP 认证失败
-15	签名内容为空

2.3. 发送批量动态短信接口

提交地址: <http://api.1680.me/sendPackSms.do>

send 接口提交方式: POST

功能: 发送动态短信

输入参数	描述
username	登陆名称 (必填)
password	登陆密码 (必填, 明文或 32 位 MD5 密码)
productid	产品 id(具体明细请联系客服)
content	短信内容格式如下: 手机号 发送内容\$手机号 发送内容\$手机号 发送内容... (每次提交最多 1000 个内容)
dstime	定时时间, 可以为空, 为空时表示立即发送
xh	扩展的小号 (选填)

样例:

<http://api.1680.me/sendPackSms.do>

username=user1&password=123456&content=13011001111 | 发 送 内 容
1\$13011001112 | 发 送 内 容 2\$13011001113 | 发 送 内 容 3\$13011001114 | 发 送 内 容
4&dstime=2010-12-30 17:48:21&productid=520411

注: 请使用 POST 方式提交, 避免提交失败

返回参数	描述
1	代表发送动态短信成功
0	失败
-1	登陆名称或者登陆密码不正确
-2	余额不够
-3	黑词审核中
-4	变量参数出现异常

-5	提交频率太快
-6	有效号码为空
-7	短信内容为空
-8	一级黑词
-9	没有 url 提交权限
-10	发送号码过多
-11	产品 ID 异常
-12	参数异常
-14	IP 认证失败
-15	签名内容为空

2.4. 余额接口

提交地址: <http://api.1680.me/balance.do>

Finance 接口提交方式: GET

功能: 查询短信剩余数量

输入参数	描述
username	登陆名称
password	登陆密码
productid	产品 ID

样例:

<http://api.1680.me/balance.do?username=userccc&password=XXXX&productid=677678>

返回参数	描述
XXXXX	XXXXX 代表该产品的剩余条数
-2	失败
-1	登陆名称或者登陆密码不正确

-5	提交频率太快
-9	没有 url 提交权限
-14	IP 认证失败

2.5. 状态报告接口

状态报告接口提交方式：GET

功能：获取网关发送结果，0 代表成功，非 0 代表失败

方式：推送方式

此方式需要用户在我平台填写下行地址

例如：<http://220.194.54.161:9010/index.aspx>

我平台每次只返回一组内容，客户被动接收状态报告并解析之

推送 URL 格式：

`msgid=xxxxxx&mobile=xxxxxxx&status=xxxxxxx`

2.6. 上行接口

上行接口提交方式：GET

功能：获取上行内容

方式：推送方式

此方式需要用户在我平台中填写上行地址

例如：<http://220.194.54.161:9010/index.aspx>

推送 URL 格式：

```
mobile=xxxx&content=xxxxx&xh=xxxxx;
```

一次推送一个号码，mobile 指上行号，content 指上行内容，xh 代表扩展号码。

注：上行内容是经过 `URLEncoder.encode(content, "UTF-8")` 编码过的内容，接受时请使用 `URLDecoder.decode(content, "utf-8")` 来进行解码。

第三章 API 接口说明（CMPP 接口）

3.1. CMPP 概述

CMPP 协议规范为中国移动通信集团公司企业规范，简称 CMPP，本接口版本为 2.0 版；旨在为企业 提供 CMPP2.0 接入规范。

该接口严格参照协议规范，除后面要说明的特殊返回值外，其它返回均为协议原默认返回值，详细内容请参见《中国移动通信互联网短信网关接口协议 2.0》。

3.2. CMPP 连接返回

返回参数	描述
0	正确
1	消息结构错：消息结构错
2	非法源地址
3	认证错
4	版本太高
5	账号暂停

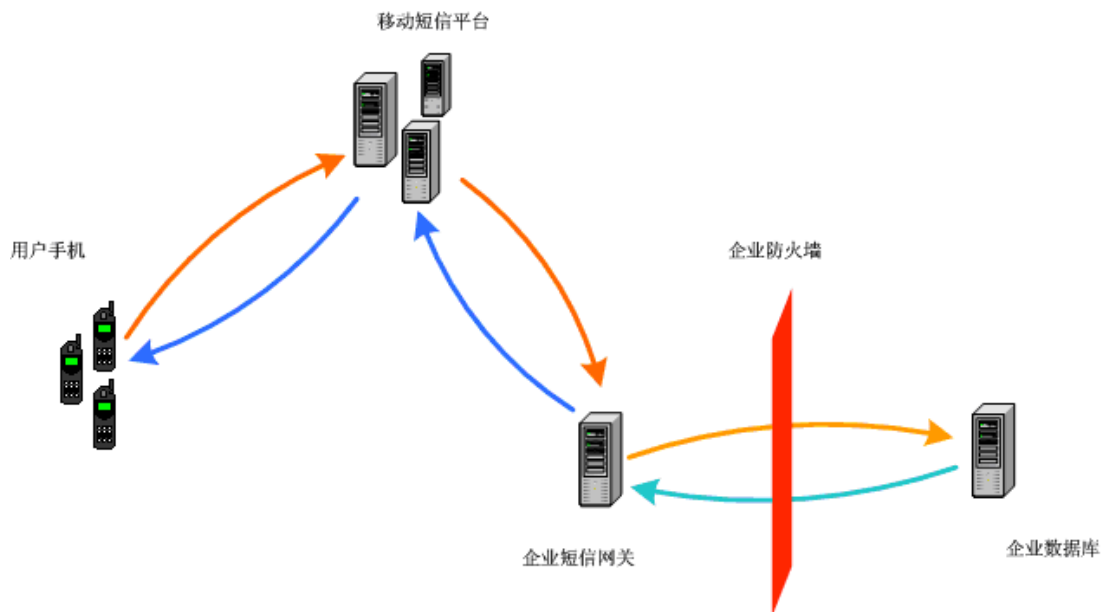
3.3. CMPP 提交返回

返回参数	描述
0	正确
1	消息结构错：消息结构错
2	命令字错
3	账号暂停
4	消息长度错
5	资费代码错
6	超过最大信息长

7	业务代码错
8	流量控制错
9	Submit 中 SPID 和登陆用户 ID 不匹配
10	用户所用产品未分配
11	接收短信的 MSISDN 号错
12	提交内容含有一级黑词，拒绝发送
13	余额不足拒绝发送
14	长号码错误

第四章 环境要求

4.1. 网络拓扑图



4.2. 环境要求

本节将详细说明正确发送/接收短信的前提条件：

确保企业服务器端可以通过 http 上网

确保企业服务器端有固定 IP，如需要，可对企业方的 IP 进行鉴权认证

4.3. 流量控制

接口对每个帐号进行了下发 (MT) 的控制，流量控制参数为：总条数/秒。

在算法上，一个 MT 发送包中的总条数=手机号码数量*(消息内容长度/70)

流量控制原理是：根据当前 MT 包的条数，控制下个 MT 包的发送时间。

例如：帐号 A 的流量是 10，第一次提交的 MT 发送包中的总条数是 5 条，那么：第 2 次提交的 MT 发送包（最多 100 条）的时间必须在，收到第一条响应包时间的 5000ms 后。

4.4. 并发控制

对于每个帐号，建议不超过五个并发。建议使用长连接提交地址。

4.5. 其它

所有参数区分大小写；所有内容均采用 UTF8 编码

1、被动接收上行 sp 的下行报告，1603 是通道号

<http://220.194.54.161:9801/1603/mr>

2、被动接收上行 sp 的上行报告，1603 是通道号

<http://220.194.54.161:9801/1603/mo>

3、特殊状态说明

返回参数	描述
KX:1000	代表下行提交失败
KX:2000	代表通道网关不存在 下行失败
KX:3000	代表扣量状态报告
KX:4000	代表同样内容，特定时间段下行次数限制

附录

密码 MD5 对照

[明文][123456]

[MD5][JAVA-DigestUtils.md5Hex][e10adc3949ba59abbe56e057f20f883e]

短信提交 DEMO (JAVA 版)

```
public static void main(String[] args) throws IOException {
    String content = "短信提交 JAVA 示例代码";
    StringBuffer sb = new StringBuffer("http://api.1680.me/sendXSms.do?");
    sb.append("username=用户名");
    sb.append("&password=密码");
    sb.append("&productid=产品 ID");
    sb.append("&mobile=13911111111,13811111111");
    sb.append("&content="+URLEncoder.encode(content, "utf-8"));
    sb.append("&dstime=");
    sb.append("&xh=11");
    URL url = new URL(sb.toString());
    HttpURLConnection connection = (HttpURLConnection) url.openConnection();
    connection.setRequestMethod("POST");
    BufferedReader in = new BufferedReader(new InputStreamReader(url.openStream()));
    String inputline = in.readLine();
    System.out.println(inputline);
}
```

短信提交 DEMO (SHELL 版)

```
//使用 linux curl 工具发送 get 请求
curl
"http://112.5.196.74:8081/sendXSms.do?username=username&password=password&productid=productid&mobile=13900000000&content=`echo '发送指令' | tr -d '\n' | xxd -plain | sed 's/\(..\)/%\1/g'`"
```


短信提交 DEMO (C/C++版)

```
//GET 方式如下: 使用 libhttp
#include "ghttp.h"
#include "http_hdrs.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
int netGet(char* url, char* params, int timeout, char **result, int result_len) {
    ghttp_request *request = NULL;
    request = ghttp_request_new();
    if(params!=NULL&&strlen(params)>0)
    {
        char tmp[1024];
        strcpy(tmp, url);
        if(strchr(tmp, '?') == NULL)//url 不存在
        {
            strcat(tmp, "?") ;
        }
        strcat(tmp, params) ;
        printf("%s\n", tmp);
        ghttp_set_uri(request, tmp);
    }else{
        ghttp_set_uri(request, url);
    }
    ghttp_set_type(request, ghttp_type_get); //get 方法
    ghttp_set_header(request, http_hdr_Connection, "close");
    char timeout_str[10];
    sprintf(timeout_str, "%d", timeout);
    ghttp_set_header(request, http_hdr_Timeout, timeout_str);
    ghttp_prepare(request);
    ghttp_process(request);
    *result = ghttp_get_body(request);
    result_len = ghttp_get_body_len(request);
    ghttp_request_destroy(request);
    return 0;
}
int main(int argc, char *argv[]) {
    char *result;
    int len;
    result=(char*)malloc(sizeof(char*)*8096);
    memset(result, 0, sizeof(char*)*8096);
    char
                                                                    param[2048]=
"username=username&password=password&productid=productid&mobile=mobile&content=zhi
ling";
    printf("%s\n%d\n", result, len);
    netGet("http://112.5.196.74:8081/sendXSms.do",param, 5000, &result, len);
    printf("%s\n%d\n", result, len);
    return 0;
}
```

短信提交 DEMO (Objective-C 版)

```
//POST 方式 prepar request
NSString          *urlString          =          [NSString
stringWithFormat:@"http://112.5.196.74:8081/sendXSms.do"];
NSMutableURLRequest *request = [[NSMutableURLRequest alloc] init] autorelease];
[request setURL:[NSURL URLWithString:urlString]];
[request setHTTPMethod:@"POST"];
//set headers
NSString *contentType = [NSString stringWithFormat:@"text/plain"];
[request addValue:contentType forHTTPHeaderField: @"Content-Type"];
//create the body
NSString          *content          =
@"username=username&password=password&productid=productid&mobile=mobile&content=zhi
ling";
NSData *data = [content dataUsingEncoding:NSUTF8StringEncoding];
[request setHTTPBody:data];
//get response
NSHTTPURLResponse* urlResponse = nil;
NSError *error = [[NSError alloc] init];
NSData *responseData = [NSURLConnection sendSynchronousRequest:request
returningResponse:&urlResponse error:&error];
NSString *result = [[NSString alloc] initWithData:responseData
encoding:NSUTF8StringEncoding];
NSLog(@"Response Code: %d", [urlResponse statusCode]);
if ([urlResponse statusCode] >= 200 && [urlResponse statusCode] < 300) {
    NSLog(@"Response: %@", result);
    //here you get the response
}
}
```

短信提交 DEMO (C#版)

```
//GET 方式提交
string content="发送指令";
string          uri          =
"http://112.5.196.74:8081/sendXSms.do?username=username&password=password&productid=
productid&mobile=13900000000";
//uri = uri + "&dstime="+ Server.UrlEncode("2015-01-01 14:41:00");
uri= uri+ "&content="+ Server.UrlEncode(content);
WebClient wc = new WebClient();
Stream st = wc.OpenRead(uri);
StreamReader sr = new StreamReader(st);
string res = sr.ReadToEnd();
sr.Close();
st.Close();
Console.WriteLine(res);
```

短信提交 DEMO (PHP 版)

//GET 方式如下:

```
<?PHP
$content="发送指令";
$url="http://112.5.196.74:8081/sendXSms.do?";
$url.="username=username";
$url.="&password=password";
$url.="&productid=productid";
$url.="&mobile=mobile";
//url.="&dstime=".urlencode("2015-01-01 14:41:00");
$content=urlencode($content);
$url.="&content=$content";
$res = file_get_contents($url);
print $res;
?>
```

短信提交 DEMO (PERL 版)

//GET 方式如下:

```
#!/usr/bin/perl
use strict;
use LWP::Simple;
use URI::Escape;
my
$url="http://112.5.196.74:8081/sendXSms.do?username=username&password=password&productid=productid&mobile=13900000000&content=";
my $content = "发送指令";
$url = $url.uri_escape($content);
my $res = get($url);
print $res;
```

短信提交 DEMO (PYTHON 版)

//GET 方式如下:

```
#coding=utf-8
from urllib import quote
import urllib2
url
"http://112.5.196.74:8081/sendXSms.do?username=username&password=password&productid=productid&mobile=13900000000&content="
content="发送指令"
content=quote(content)
url = url + content
req = urllib2.Request(url)
res_data = urllib2.urlopen(req)
res = res_data.read()
print res
```

短信提交 DEMO (RUBY 版)

//GET 方式如下:

```
require "open-uri"
uri =
' http://112.5.196.74:8081/sendXSms.do?username=username&password=password&productid
=productid&mobile=13900000000&content='
content = '发送指令'
content = URI::escape(content)
uri = uri + content
html_response = nil
open(uri) do |http|
  html_response = http.read
end
puts html_response
```

短信提交 DEMO (LUA 版)

//GET 方式如下: 使用 luasocket

```
local http = require("socket.http")
local ltn12 = require("ltn12")

function http.get(u)
  local t = {}
  local r, c, h = http.request{
    url = u,
    sink = ltn12.sink.table(t)}
  return r, c, h, table.concat(t)
end

url =
" http://112.5.196.74:8081/sendXSms.do?username=username&password=password&productid
=productid&mobile=mobile&content=zhiling"
r, c, h, body=http.get(url)
if c~= 200 then
  return
end
print(body)
```