融合·通信云 短信接口技术规范

版本号	日期	修改人	审阅人	摘要
V1.0	2010-10-20	-	-	初稿
V1.5	2012-12-17	-	-	12修订版
V2.0	2013-06-17	-	-	13修订版
V2.1	2016-06-03	-	-	16修订版
V2.2	2016-06-29	-	-	修正状态报告推送方式
V2.5	2021-09-20	-	-	修正 json 收发方式

1. 接口概况

常见问题回答

==**这么多接口普通接入xml需要开发那个？**==

> 一般接入只需开发 ==4.1.1、4.3.1== 即可。

==**这么多接口普通接入json需要开发那个？**==

> 一般接入只需开发 ==4.1.2、4.3.2== 即可。

==**状态报告可以推送给客户么?**==

> 可以.客户需要提供ip,端口.我们会以json格式推送状态报告。数据格式请参考 ==4.7== 进行开发。

==**下发消息和返回的状态报告如何对应？**==

> 下发消息中用户端生成的==usermsgid==键值(不超过32位，由数字和英文字母组成的字符串组合）,可与状态报告中的==usermsgid==进行对应，即可完成下发消息与状态报告的一一对应。

==**下发短信内容应该采取什么编码格式？**==

> 必须为utf-8格式，不支持gbk等其他格式，并且需要进行base64加密。

==**可以多进程或者多线程获取状态报告和上行消息么？**==

> 不可以，只能用一个线程获取状态报告和上行。获取状态报告时候，如果有上行，会同时返回状态报告和上行消息。

基本流程：

```
sequenceDiagram
客户平台->>融合云平台: 请求(HTTP POST+XML)
融合云平台-->>客户平台: 应答(HTTP POST+XML)
```

状态报告推送流程:

```
sequenceDiagram
融合云平台->>客户平台: 请求(HTTP POST+JSON)
客户平台-->>融合云平台: 应答(HTTP POST+JSON)
```

测试服务信息

```
http://123.56.14.50:9191/adc_posthandler_test       #北京
```

用户名密码都是123456,测试用签名:【测试】

测试服务只能用于检测程序是否正确，不能下发短信。

2. 接口通信方式

各客户平台与融合云平台通过HTTP POST XML或JSON 进行数据的交互。

3. 消息格式

消息格式分为HTTP消息头和消息体两部分。消息体为完整的XML或JSON。注意消息体编码要求：utf-8。

4. 接口说明

4.1. 发送短信（客户平台->融合云平台）

4.1.1 XML

发送请求消息参数

消息头

```
POST /url HTTP/1.1
Host: xxx.xxx.xxx.xxx
Connection: close
Content-Type: text/xml; charset=utf-8
Content-Length: length #消息体(xml)长度
Action: "submitreq" #注意submitreq两边的双引号要加
```

消息体

```
<?xml version="1.0" encoding="utf-8"?>
<Body>
    <user>用户名</user>
    <password>密码</password>
    <version>1.2</version>
    <submit>
        <usermsgid>唯一标识，不超过32位，由数字和英文字母组成的字符串，由用户生成，该值将与状态报告中的usermsgid一一对应</usermsgid>
        <desttermid>目的号码,用户手机号码</desttermid>
        <srctermid>下发源号码的扩展号码</srctermid>
        <msgcontent>base64(utf-8)</msgcontent>
        <signid>默认填0</signid>
        <desttype>1移动，2联通，3电信</desttype>            #不知道的话填0
        <needreply>0,不需要状态报告 1,需要</needreply>
    </submit>
    <submit>
…
    </submit>
</Body>
```

发送结果消息参数

消息头

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/xml; charset=utf-8
Content-Length: length #消息体(xml)长度
```

消息体

```
<?xml version="1.0" encoding="utf-8"?>
<Body>
    <result>0成功，其他失败</result>
</Body>
```

4.1.2 json

发送请求消息参数

消息头

```
POST /url HTTP/1.1
Host: xxx.xxx.xxx.xxx
Connection: close
Content-Type: text/json; charset=utf-8
Content-Length: length #消息体(xml)长度
Action: "submitreq" #注意submitreq两边的双引号要加
```

消息体

```
{
   "user":"用户名",
   "password":"密码",
   "submit":[
      {
         "srctermid":"下发源号码的扩展号码",
         "desttermid":"目的号码,用户手机号码",
         "msgcontent":"base64(utf-8)",
         "usermsgid":"唯一标识，不超过32位，由数字和英文字母组成的字符串，由用户生成，该值
将与状态报告中的usermsgid一一对应",
         "desttype":"1移动，2联通，3电信"              #不知道的话填0
      }
   ]
}
```

发送结果消息参数

消息头

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/json; charset=utf-8
Content-Length: length #消息体(xml)长度
```

消息体

```
{
   "result":"0:成功，其他失败"
}
```

4.2. 查询余额（客户平台->融合云平台）

4.2.1 xml

查询请求消息参数

消息头

```
POST /url HTTP/1.1
Host: xxx.xxx.xxx.xxx
Connection: close
Content-Type: text/xml; charset=utf-8
Content-Length: length #消息体(xml)长度
Action: "checkbalances" #注意checkbalances两边的双引号要加
```

消息体

```
<?xml version="1.0" encoding="utf-8"?>
<Body>
    <user>用户名</user>
    <password>密码</password>
    <version>1.2</version>
</Body>
```

查询结果消息参数

消息头

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/xml; charset=utf-8
Content-Length: length #消息体(xml)长度
```

消息体

```
<?xml version="1.0" encoding="utf-8"?>
<Body>
    <result>0或大于0为移动剩余金额，其他为错误</result>
</Body>
```

4.3. 获得上行短信、状态报告（客户平台->融合云平台）

4.3.1 xml

获取请求消息参数

消息头

```
POST /url HTTP/1.1
Host: xxx.xxx.xxx.xxx
Connection: close
Content-Type: text/xml; charset=utf-8
Content-Length: length #消息体(xml)长度
Action: "getreportsms2" #注意getreportsms2两边的双引号要加
```

消息体

```
<?xml version="1.0" encoding="utf-8"?>
<Body>
    <user>用户名</user>
    <password>密码</password>
    <version>1.2</version>
</Body>
```

获取结果消息参数

消息头

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/xml; charset=utf-8
Content-Length: length #消息体(xml)长度
```

消息体

```
<?xml version="1.0" encoding="utf-8"?>
<Body>
    <result>1表示有数据，0表示没有数据，其余为失败</result>
    <deliver>
        <srctermid>源号码,用户手机号码</srctermid>
        <desttermid>目的号码，长号码</desttermid>
        <msgcontent>base64(utf-8)上行短信内容</msgcontent>
        <srctype>1移动，2联通，3电信</srctype>
        <receivedatetime>接收时间格式:YYYY-MM-DD HH24:MI:SS</receivedatetime>
    </deliver>
    .
    .
    .
    最长到350条，上行一次最多50条、状态报告最多300条
    .
    .
    .
    <report>
        <msgid>空，保留字段</msgid>
        <srctermid>源号码,用户手机号码</srctermid>
        <desttermid>目的号码，长号码</desttermid>
        <msgcontent>状态报告结果，base64编码格式，DELIVRD为成功 </msgcontent>

        <srctype>1移动，2联通，3电信</srctype>
        <usermsgid>用户发送时定义的usermsgid</usermsgid>
        <receivedatetime>接收时间格式:YYYY-MM-DD HH24:MI:SS</receivedatetime>
    </report>
    .
    .
    .
</Body>
```

4.3.2 json

获取请求消息参数

消息头

```
POST /url HTTP/1.1
Host: xxx.xxx.xxx.xxx
Connection: close
Content-Type: text/json; charset=utf-8
Content-Length: length #消息体(xml)长度
Action: "getreportsms2" #注意getreportsms2两边的双引号要加
```

消息体

```
{
   "user":"用户名",
   "password":"密码"
}
```

获取结果消息参数

消息头

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/json; charset=utf-8
Content-Length: length #消息体(xml)长度
```

消息体

```
{
   "result":"1表示有数据，0表示没有数据，其余为失败",
   "deliver":[
      {
         "srctermid":"源号码,用户手机号码",
         "desttermid":"目的号码，长号码",
         "msgcontent":"base64(utf-8)上行短信内容",
         "srctype":"1移动，2联通，3电信",
         "receivedatetime":"接收时间格式:YYYY-MM-DD HH24:MI:SS"
      },
      {
         ...
      },
      {
         ...
      }
   ],
   "report":[
      {
         "msgid":"空，保留字段",
         "srctermid":"源号码,用户手机号码",
         "desttermid":"目的号码，长号码",
         "msgcontent":"状态报告结果，base64编码格式，DELIVRD为成功 ",

         "srctype":"1移动，2联通，3电信",
         "usermsgid":"用户发送时定义的usermsgid",
         "receivedatetime":"接收时间格式:YYYY-MM-DD HH24:MI:SS"
      },
      {
         ...
      },
      {
         ...
      }
   ]
}
```

4.4. 状态报告推送(融合云平台->客户平台)

发送请求消息参数

消息头

```
POST /url HTTP/1.1
Host: xxx.xxx.xxx.xxx
Connection: close
Content-Type: application/json; charset=utf-8
Content-Length: length #消息体(json)长度
```

消息体

> 最长到350条，上行一次最多50条、状态报告最多300条

```
{"report": [
    ["123456","13812345678", "1069000000", "REVMSVZSRA==", "1", "012345678901", "2016-06-29 11:16:00"],
    ["123456","13812345679", "1069000000", "REVMSVZSRA==", "1", "012345678902", "2016-06-29 11:16:00"]
]
}
列表的格式顺序:[接口账户名,手机号码,源号码,状态报告结果(base64),所属运营商,usermsgid,网关接收时间]
{"mo": [
    ["123456','13812345678", "1069000000", "REVMSVZSRA==", "1", "2016-06-29 11:16:00"],
    ["123456','13812345679", "1069000000", "REVMSVZSRA==", "1", "2016-06-29 11:16:00"]
]
}
列表的格式顺序是[接口帐户名,手机号,目的号码,用户回复内容(base64),所属运营商,网关接收时间]
```

发送结果消息参数

消息头

```
HTTP/1.1 200 OK
Connection: close
Content-Type: application/json; charset=utf-8
Content-Length: length #消息体(json)长度
```

消息体

```
{"result":"0"}
```

5. 返回状态码

返回码	说明
0	成功
-1	消息个数大于100
-2	消息内容不是base64格式
-3	没有消息内容
-4	没有目的号码
-5	desttype填写错误
-6	手机号码不正确
-8	needreply填写错误
-9	余额不足
-10	数据库准备失败
-11	用户名密码错误
-12	xml格式错误
-13	Action填写错误
-14	消息内容加上签名长度大于469字
-15	signid填写错误，未找到对应签名
-16	联通全网发送失败
-17	联通全网下发源号码配置错误，与管理员确认
-18	该签名为指定号码下发，不支持长号码扩展
-19	短信内容含有敏感词汇
-20	usermsgid长度超过32位
-21	usermsgid不是由数字和英文字母组成
-23	msgcontent短信内容中的签名错误
-24	模板编号错误
-25	发送内容不匹配模板
-26	消息头中没有Content-Length

java demo 仅供参考 请根据本单位实际情况编写

```
package com.cmcc.tj.res;
import it.sauronsoftware.base64.Base64;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.StringReader;
import java.io.UnsupportedEncodingException;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLConnection;
import java.util.HashMap;
import java.util.Map;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.NodeList;
import org.xml.sax.InputSource;
import org.xml.sax.SAXException;

public class demo {
    public static String[] sendSms(String phoneNumber, String smsContent) {
        System.out.println("sendSms begin");
//        String url = "http://211.137.171.46:9191/adc_posthandler_new";
        String url = "http://111.30.60.54:9191/adc_posthandler_test";
//此处修改为最终发送地址
        String result = testPost(url, phoneNumber, smsContent);
        System.out.println(result);
        if (result == ""){
            String[] arr = {"-1", "null err"};
            return arr;
        }else{
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            try {
                DocumentBuilder builder = factory.newDocumentBuilder();
                Document doc = builder.parse(new InputSource(new
StringReader(result)));
                Element root = doc.getDocumentElement();
                NodeList books = root.getChildNodes();
                String nl =  books.item(0).getFirstChild().getNodeValue();
                System.out.println(nl);
                if (nl.equalsIgnoreCase("0")){
                    return null;
                }else{
                    return nl.split(":");
```

```
                }
            } catch (ParserConfigurationException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (SAXException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (IOException e) {
                // TODO Auto-generated catch blockwebservice.putheader("
                e.printStackTrace();
            }
        }
        return null;
    }
    public static String testPost(String urlStr, String phoneNumber, String
smsContent) {
        String line = "";
        try {
            URL url = new URL(urlStr);
            URLConnection con = url.openConnection();
            con.setDoOutput(true);
            con.setRequestProperty("Pragma", "no-cache");
            con.setRequestProperty("Cache-Control", "no-cache");
            con.setRequestProperty("Connection", "close");
            con.setRequestProperty("Content-Type", "text/xml; charset=utf-8");
            con.setRequestProperty("Action", "\"submitreq\"");
//此处按照接口文档说明按收发实际正确配置

            String xmlInfo = getXmlInfo(phoneNumber, smsContent);
            System.out.println("urlStr=" + urlStr);
            System.out.println("xmlInfo=" + xmlInfo);
            if (xmlInfo == ""){
                return "";
            }else{
                con.setRequestProperty("Content-Length", Integer.toString(new
String(xmlInfo.getBytes("UTF-8")).length()));
                OutputStreamWriter out = new
OutputStreamWriter(con.getOutputStream());
                out.write(new String(xmlInfo.getBytes("UTF-8")));
                out.flush();
                out.close();
                BufferedReader br = new BufferedReader(new
InputStreamReader(con.getInputStream()));
//                System.out.println(br.readLine());
                for (line = br.readLine(); line != null; line = br.readLine()) {
                    break;
                }
            }
        } catch (MalformedURLException e) {
            e.printStackTrace();
            line = "";
```

```
        } catch (IOException e) {
            e.printStackTrace();
            line = "";
        }
        return line;
    }

    private static String getXmlInfo(String phoneNumber, String smsContent) {
        StringBuffer msg = new StringBuffer();
        msg.append(smsContent);
        String msgutf8;
        try {
            msgutf8 = new String(msg.toString().getBytes("UTF-8"));
        } catch (UnsupportedEncodingException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            msgutf8 = "";
        }
        if (msgutf8 != ""){
            StringBuilder sb = new StringBuilder();
            sb.append("<?xml version=\"1.0\" encoding=\"utf-8\"?><Body>");
            sb.append("<user>123456</user><password>123456</password>
<version>1.2</version><submit><usermsgid>123456789012</usermsgid><desttermid>" +
phoneNumber + "</desttermid><srctermid></srctermid><msgcontent>" + getBase64(msgutf8)
+ "</msgcontent><signid></signid><desttype>0</desttype><needreply>1</needreply>
</submit></Body>");            //此处改成正式配给的账号密码
            return sb.toString();
        }else{
            return "";
        }
    }
    public static String getBase64(String str) {
        String encoded = Base64.encode(str, "UTF-8");
        return encoded;
    }

    public String getFromBase64(String s) {
        String decoded = Base64.decode(s, "UTF-8");
        return decoded;
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        String[] results = sendSms("13888888888", "测试下发【测试】");
//此处改成目的手机号码和实际短信内容
        System.out.println(results);
        if (results == null) {
            System.out.println(results);
        }else{
            for (int i = 0 ; i <results.length ; i++ ) {
                  System.out.println("--"+results[i]);
```

```
            }
        }
    }
}
```

python demo 仅供参考 请根据本单位实际情况编写

```python
#coding=utf-8

import base64
import httplib
from xml.sax.saxutils import unescape
import urllib

ip = "123.56.14.50"
port = 9191
mobile = "13888888888"
src_term = "" # 测试时候填空
msg = base64.encodestring("这是测试!【测试】")

def sendsms():
    try:
        XmlMessage = """<?xml version="1.0" encoding="UTF-8"?><Body>
<user>123456</user><password>123456</password><version>1.2</version>%s</Body>
        """.replace('\r', '').replace('\n', '')
        submit = "<submit><usermsgid>012345678912</usermsgid>
<desttermid>%s</desttermid><srctermid>%s</srctermid><msgcontent>%s</msgcontent>
<signid></signid><desttype>0</desttype><tempid>1</tempid><needreply>1</needreply>
</submit>"
        submit = submit %(mobile, src_term, msg)
        XmlMessage = XmlMessage % submit
        conn = httplib.HTTPConnection(ip, port)
        conn.putrequest("POST", "/adc_posthandler_test",0,1)
        conn.putheader("Content-Type", "text/xml; charset=utf-8")
        conn.putheader("Content-Length", "%d" % len(XmlMessage))
        conn.putheader("Connection", 'Close')
        conn.putheader("Pragma", "no-cache")
        conn.putheader("Action", '"submitreq"')
        conn.endheaders()
        conn.send(XmlMessage)
        response = conn.getresponse()
        resp = response.read()
        resp = unescape(resp.replace('\n','').replace('\r',''))
        print resp
    except Exception, e:
        print e
    finally:
        if conn:
            conn.close()

if __name__ == '__main__':
    sendsms()
```

php demo 仅供参考 请根据本单位实际情况编写

```php
<?php

//下面的demo，实现了简单的发送功能。

//首先检测是否支持curl
if (!extension_loaded("curl")) {
    trigger_error("对不起，请开启curl功能模块！", E_USER_ERROR);
}

//构造xml
$xmldata=
'<?xml version="1.0" encoding="utf-8"?><Body><user>123456</user>
<password>123456</password><version>1.2</version><submit>
<usermsgid>L6z6CGj2tK9T</usermsgid><desttermid>17610063060</desttermid>
<srctermid>1451</srctermid>
<msgcontent>5rWL6K+V54q25oCB5oql5ZGK44CQ5rWL6K+V44CR</msgcontent><signid></signid>
<desttype>0</desttype><needreply>1</needreply></submit></Body>';

//初始一个curl会话
$curl = curl_init();

//设置url
curl_setopt($curl, CURLOPT_URL,"http://123.56.14.50:9191/adc_posthandler_test");

//设置发送方式：post
curl_setopt($curl, CURLOPT_POST, true);
//注意：'Action: "submitreq"' 里面必须有双引号，其他Action也类似
curl_setopt($curl, CURLOPT_HTTPHEADER, array('Pragma: no-cache', 'Connection: close',
'Content-Type: text/xml; charset=utf-8', 'Content-Length: ' . strlen($xmldata),
'Action: "submitreq"'));


//设置发送数据
curl_setopt($curl, CURLOPT_POSTFIELDS, $xmldata);

//TRUE 将curl_exec()获取的信息以字符串返回，而不是直接输出
curl_setopt($curl, CURLOPT_RETURNTRANSFER, TRUE);

//执行cURL会话 ( 返回的数据为xml )
$return_xml = curl_exec($curl);

//关闭cURL资源，并且释放系统资源
curl_close($curl);

echo $return_xml;
exit;
?>
```