

中国移动代理服务器 MAS V2.0

JAVA API 接口插件

开发手册



深圳市嘉讯软件有限公司

2008 年 9 月

目录

1.	引言.....	4
1.1.	编写目的.....	4
1.2.	背景说明.....	4
1.3.	相关术语与缩略语解释.....	4
2.	API 结构说明	4
3.	运行环境.....	5
4.	开发步骤.....	5
4.1.	在 Mas 服务器新建一个企业应用	5
4.2.	启动企业应用.....	6
4.3.	开发时需要引入的包.....	6
4.4.	接口开发及使用举例.....	6
4.4.1.	APIClient 工厂类-ApiClientFactory 类	7
4.4.1.1.	创建短信 API 客户端-createSmsApiClient 方法	7
4.4.2.	短信接口.....	7
4.4.2.1.	短信 API 客户端主要操作类-SmsApiClient 类.....	7
4.4.2.1.1.	设置自动重连服务器-setTimingConnect 方法	7
4.4.2.1.2.	设置自动重连服务器相隔时间-setConnectTime 方法.....	7
4.4.2.1.3.	设置与服务连接超时时长-setConnectTimeout 方法.....	8
4.4.2.1.4.	设置发送超时时长-setSendTimeout 方法	8
4.4.2.1.5.	启动 API-start 方法	8
4.4.2.1.6.	获得网关连接状态-getConnStatusIAGW 方法.....	8
4.4.2.1.7.	取消短信发送- cancelSms 方法	8
4.4.2.1.8.	获得短信队列中未发送的条数-getStat 方法.....	8
4.4.2.1.9.	获得扩展服务代码长度- getXCodeLength 方法	8
4.4.2.1.10.	计算短信条数和字数-getSmsCount 方法	8
4.4.2.1.11.	获得预览的短信内容-getSmsContentPreview 方法 错误！未定义书签。	
4.4.2.1.12.	短信 API 退出-loginOut 方法	8
4.4.2.1.13.	获得 MAS 短信提交的限制数量-getDestAddrsLimit 方法	9
4.4.2.1.14.	发送短信-sendSms 方法	9
4.4.2.2.	短信回调对象处理类-SmsApiClientHandler 类	10
4.4.2.2.1.	发送响应通知-notifySmsSendResponse 方法 错误！未定义书签。	
4.4.2.2.2.	状态报告通知-notifySmsDeliveryStatus 方法.....	10
4.4.2.2.3.	MO 短信通知-notifySmsReception 方法.....	10
4.4.2.2.4.	通知取消成功的手机号码-notifyCancelSmsList 方法.....	11
4.4.2.3.	短信开发客户端代码举例.....	11
4.4.3.	彩信接口.....	14
4.4.3.1.	彩信 API 客户端主要操作类-MmsApiClient 类	14
4.4.3.1.1.	设置是否自动重连到服务器-setTimingConnect 方法	14
4.4.3.1.2.	设置自动重连服务器相隔时间-setConnectTime 方法.....	14

4.4.3.1.3.	设置与服务连接超时时长-setConnectTimeout 方法.....	14
4.4.3.1.4.	设置发送超时时长-setSendTimeout 方法.....	14
4.4.3.1.5.	启动 API -start 方法	14
4.4.3.1.6.	彩信 API 退出-loginOut 方法	14
4.4.3.1.7.	获得网关连接状态-getConnStatusIAGW 方法.....	14
4.4.3.1.8.	获得 MAS 彩信提交的限制数量-getDestAddrsLimit 方法	15
4.4.3.1.9.	发送彩信-sendMms 方法	15
4.4.3.1.10.	获取该次发送还未提交到彩信网关的数量-getStat 方法	15
4.4.3.1.11.	取消彩信发送-canelMms 方法	16
4.4.3.2.	彩信服务回调对象进行处理类-MmsApiClientHandler 类	16
4.4.3.2.1.	发送响应通知-notifyMmsSendResponse 方法.....	16
4.4.3.2.2.	状态报告通知-notifyMmsDeliveryStatus 方法	16
4.4.3.2.3.	MO 彩信通知-notifyMmsReception 方法	17
4.4.3.2.4.	通知取消成功的手机号码-notifyCanelMmsMobile 方法	错误!未定义书签。
4.4.3.3.	彩信多媒体消息内容格式定义说明	17
4.4.3.3.1.	彩信多媒体内容类型.....	17
4.4.3.3.2.	彩信多媒体内容加密.....	17
4.4.3.3.3.	彩信多媒体内容编码.....	17
4.4.3.3.4.	彩信多媒体内容支持附件的类型.....	18
4.4.3.3.5.	彩信多媒体消息内容示例	18
4.4.3.3.6.	彩信多媒体消息具体加密前后内容示例	18
4.4.3.4.	彩信开发客户端代码举例.....	20
4.4.4.	API 异常错误对应常量	22
5.	接口描述.....	23
5.1.	短信接口描述.....	23
5.1.1.	接口说明.....	23
5.1.2.	流程说明.....	23
5.1.3.	主要类说明.....	24
5.2.	彩信接口描述.....	24
5.2.1.	接口说明.....	24
5.2.2.	流程说明.....	24
5.2.3.	主要类说明.....	25
6.	API 短信 DEMO 使用说明.....	25
6.1.1.	使用说明.....	25
6.1.2.	主要命令说明.....	25
7.	注意事项.....	27
8.	参考文档.....	27

1. 引言

1.1. 编写目的

本手册详细描述了中国移动代理服务器(MAS)短信、彩信 API 的开发步骤以及注意事项。

其阅读对象为需要 MAS 开发、工程支持人员、集成开发人员、SI 厂商开发人员、企业客户 IT 开发人员。

1.2. 背景说明

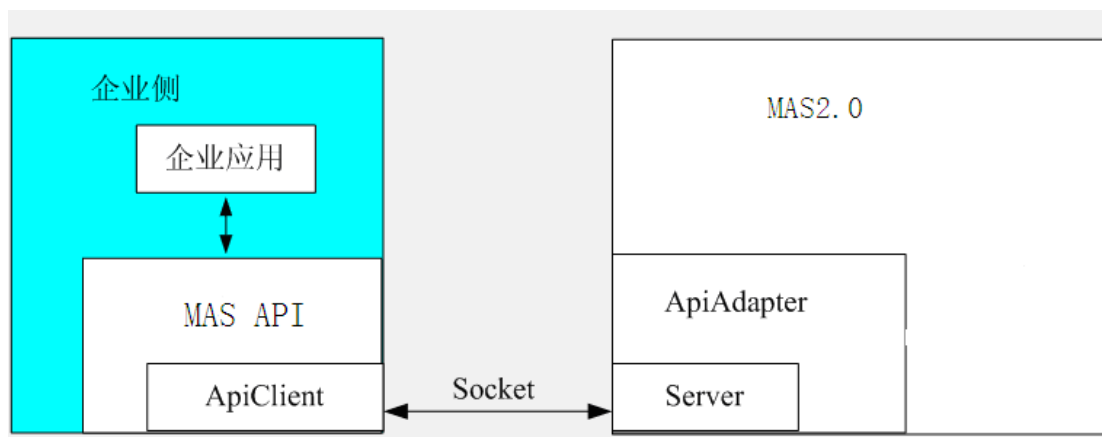
近年来移动业务高速发展，MAS 业务也开展的如火如荼，但同时许多集团客户有许多自己的 IT 系统，他们如何把自己的 IT 系统也“移动”化起来呢？MAS 做为移动代理服务器，提供了许多手段来供企业 IT 系统接入（如 WS 接口、DB 接口、WAP 适配），但此次 API 又为这许多手段之外再增加了一种方式，并且这种方式更为高效、更稳定、扩展性更强。

本手册适用于产品“中国移动代理服务器 MAS” V2.0 版本。只描述了以 JAVA 语言客户端来接入 MAS 的内容，其它语言开发客户端可以参考本手册。

1.3. 相关术语与缩略语解释

缩写词	英文解析	中文解析
MAS	Mobile Agent Server	移动代理服务器
HTTP	Hypertext Transfer Protocol	超文本传输协议
HTTPS	Secure HTTP	加密的 HTTP 协议
SOAP	Simple Object Access Protocol	简单对象访问协议
UCA	Unified Communication Agent	统一通信代理

2. API 结构说明



企业应用通过 APIClient 来与 MAS 通信，底层采用 Socket，这样可以保证高性能与稳定性。

API 方式通信适配插件接口封装了以下移动通信能力：

- 短信
- 彩信

此外，还提供基座与企业应用之间的插件管理功能接口。

MAS 服务器应提供配置工具，指定对于某个集团客户应用系统或应用插件，通过 API 方式通信适配插件或数据库方式通信适配插件接入 MAS 服务器。

这是当应用作为插件的身份来接入 MAS 时才考虑的开发环境。当应用不作为插件而做为集团客户应用的身份来接入时不需关注，只需通过标准的 API 接口即可。

3. 运行环境

- MAS2.0 基座运行环境
- MAS 服务平台环境
- JDK1.5 及以上

4. 开发步骤

4.1. 在 Mas 服务器新建一个企业应用

- 运行 MAS2 服务器，登录到系统管理->“插件管理”中新建一个企业应用

在开发完成后准备调试前，先运行 MAS2.0 服务器，然后在插件管理里新建一个应用或插件。

应用类型：	<input checked="" type="radio"/> 企业应用 <input type="radio"/> 应用插件	
显示类别：	内部信息	
标识 *：	API插件	
名称 *：	API插件	
密码 *：	*****	
通信接口方式：	API	
类型：	ERP	
应用IP地址 *：	127.0.0.1	
短信应用：	<input checked="" type="radio"/> 是 <input type="radio"/> 否 扩展号码 *：123 匹配格式 <input type="radio"/> 精确 <input checked="" type="radio"/> 模糊 MO命令字 *：123 (多个用;号隔开) 业务代码：	
彩信应用：	<input checked="" type="radio"/> 是 <input type="radio"/> 否 扩展号码：222 <input type="radio"/> 精确 <input checked="" type="radio"/> 模糊	
WAP应用：	<input type="radio"/> 是 <input checked="" type="radio"/> 否	
USSD应用：	<input type="radio"/> 是 <input checked="" type="radio"/> 否	
LBS应用：	<input type="radio"/> 是 <input checked="" type="radio"/> 否	
CPU门限值：	警戒 0.5 严重 0.7 高危 0.9	
内存门限值：	警戒 0.5 严重 0.7 高危 0.9	
硬盘门限值：	警戒 0.5 严重 0.7 高危 0.9	
<input type="button" value="保存"/> <input type="button" value="取消"/>		

图 4-1 新增插件页面

注意：新建时“通信接口方式”选择“API”。

4.2. 启动企业应用

<input type="checkbox"/>	标识	名称	类型	通信接口	短信应用	彩信应用	WAP应用	USSD应用	LBS应用	状态	开关	安装
<input type="checkbox"/>	API插件	API插件	ERP	API	短信子号：123 MO命令字：123	彩信子号：222	否	否	否	需要注册	<input checked="" type="button" value="运行"/>	否
<input type="checkbox"/>	API202020XXXX02	彩信API测试插件	ERP	API	短信子号：5056 MO命令字：MMS115 MMS116	彩信子号：666	否	否	否	正常	<input type="button" value="暂停"/>	否
<input type="checkbox"/>	P0200000000000100	默认插件	INDUSTRY	InnerSpecific	短信子号：00000 MO命令字：DEFAULTMO	彩信子号：	是	是	是	正常	<input type="button" value="暂停"/>	

图 4-2 插件管理页面

要新建的插件 API 接口可用，需要点击“开关”列的运行按钮。否则在使用 API 接口通信时可能会报 POL0906 错误。当然要确保 API 接口能正常使用，还需要管理平台相应的业务能力。

4.3. 开发时需要引入的包

客户端开发时，要引入 mas-api.3.x.jar 包。

4.4. 接口开发及使用举例

4.4.1. APIClient 工厂类-ApiClientFactory 类

描述：APIClient 工厂类，用来创建短信 APIClient 与彩信 APIClient。

4.4.1.1. 创建短信 API 客户端-createSmsApiClient 方法

功能：创建短信 API 客户端，一个 SmsApiClient 绑定了一个 Socket 连接。

参 数	描 述	备 注
smsHandler	短信服务回调接口，如状态报告通知、MO 短信接收、发送响应等	
masIP	MAS 服务器 IP 地址	
masApiPort	MAS 服务器 API 适配监听端口	MAS 默认一般是 61616，具体可在 MAS 上修改
apiID	API 的标识，新建 API 时指定的	见图 4-1
password	API 登录的密码，新建 API 时指定	见图 4-1

4.4.1.2. 创建彩信 API 客户端- createMmsApiClient 方法

功能：创建彩信 API 客户端，一个 MmsApiClient 绑定了一个 Socket 连接。

参 数	类 型	描 述	备 注
mmsHandler	MmsApiClientHandler	彩信服务回调接口，如状态报告通知、MO 彩信接收等	
masIP	String	EMA 服务器 IP 地址	
masApiPort	int	EMA 服务器彩信 API 适配监听端口	EMA 默认一般是 61617，具体可在 EMA 上修改
apiID	String	API 的标识，新建 API 时指定的	见图 4-1
password	String	API 登录的密码，新建 API 时指定	见图 4-1

4.4.2. 短信接口

4.4.2.1. 短信 API 客户端主要操作类-SmsApiClient 类

短信 API 客户端主要操作接口，通过 ApiClientFactory 类来创建期实例，一个 SmsApiClient 对象代表了与服务器连接，在正常与服务器通信时需要先调其 start 方法来初始化。

4.4.2.1.1. 设置自动重连服务器-setAutoConnect 方法

功能：设置是否自动重连到服务器。

4.4.2.1.2. 设置自动重连服务器相隔时间-setReConnectInterval 方

法

功能：设置自动重连服务器相隔时间(单位：秒)，默认为 30 秒

4.4.2.1.3. 设置与服务连接超时时长-setConnectTimeout 方法

功能：设置与服务连接超时时长，单位：millisecond，需要在调用 start 方法前设置方可生效。

4.4.2.1.4. 设置发送超时时长-setSendTimeout 方法

功能：设置发送超时时长，单位：millisecond，需要在调用 start 方法前设置方可生效。

4.4.2.1.5. 启动 API-start 方法

功能：启动 API，该方法会自动连接到 MAS 服务器并进行登录。所有的发送、接收等操作都需要在 start 成功后方可使用。

4.4.2.1.6. 获得网关连接状态-getConnStatusIAGW 方法

功能：获得网关连接状态。

4.4.2.1.7. 取消短信发送- cancelSms 方法

功能：取消短信发送。

4.4.2.1.8. 获得短信队列中未发送的条数-getStat 方法

功能：获得短信队列中未发送的条数。

4.4.2.1.9. 获得扩展服务代码长度- getXCodeLength 方法

功能：获得扩展服务代码(插件短信扩展号码+流水号)长度。

4.4.2.1.10. 计算短信条数和字数-getSmsCount 方法

功能：计算短信条数和字数。

4.4.2.1.11. 短信 API 退出-loginOut 方法

功能：短信 API 退出。

4.4.2.1.12. 获得 MAS 短信提交的限制数量-getDestAddrsLimit 方法

功能：获得 MAS 当前短信提交的限制数量。

4.4.2.1.13. 发送短信-sendSms 方法

功能：发送短信。SmsSendRequest 对象参数说明如下：

字 段	类 型	描 述	备 注
appId	String	插件 ID	API 插件标识 ID
destAddrs	List<String>	目的地址，即手机号码集合	一次发送的最大长度，具体可通过 getDestAddrsLimit 方法获得
xCode	String	短信子码+流水号	
message	String	短信内容	
msgFormat	MsgFmt	短信内容编码格式	ASCII: ASCII 字符。 UCS2G: USC2 格式的 UniCode 字符。 B18030: GB18030 格式的中文字符。 GB2312: GB2312 格式的中文字符。 Binary: 二进制短信，用十六进制字符串。
type	SmsType	短信类型	Normal: 普通短信 Instant: 免提短信 Long: 长短信 Structured: 二进制短信 WapPush: WapPush 短信
priority	int	短信优先级。大于 0 的正整数，数字越小优先级越高。	
targetURL	String	推送 URL	发送 Wappush 短信时有效，其它类型短信时忽略
isNeedReport	boolean	是否需要状态报告	
validTime	int	短信存活期，单位秒	开始时间以 MAS 收到 API 提交时间为准，validTime 默认为 0 时，存活期为 24 小时。
linkedID	String	保留字段	
返回值	SmsSendResponse	本次发送响应	

其 SmsSendResponse 参数说明如下

字 段	类 型	描 述	备 注
-----	-----	-----	-----

requestID	String	提交到 MAS 服务平台标识	对应 submitID
code	String	错误编码	
desc	String	错误描述	

4.4.2.2. 短信回调对象处理类-SmsApiClientHandler 类

描述：由企业应用来实现，该接口类主要是对短信服务的一些回调对象进行处理，如接收状态报告、MO 短信等。

4.4.2.2.1. 状态报告通知-notifySmsDeliveryStatus 方法

功能：状态报告通知。如果在发送时指定不要状态报告将不会调用到此方法

字 段	类 型	描 述	备 注
submitID	String	提交到 MAS 服务平台标识	sendSms 方法返回的值
deliveryInfo	Report	状态报告对象	

其 Report 参数说明如下：

字 段	类 型	描 述	备 注
address	String	提交的手机号码	
sendTime	long	提交到网关的时间	
detailResult	String	规范状态报告描述	可以忽略此项
time	long	状态报告收到时间	
sendResult	int	提交到网关的返回响应的结果	此值为 0，表示正确提交到网关
smsType	int	短信类型	
stat	String	提交网关成功后，网关返回的状态报告的状态码	sendResult 为 0 时此值有效，stat 为“DELIVRD”，表示手机正确收到短信，其它表示发送失败，具体描述看 errDesc
msgID	String	短消息 ID	可不用关心
errDesc	String	错误描述	

注：一条状态报告成功与否有以下几种情况

1. sendResult 如果为 0，则看 stat，stat 为“DELIVRD”则表示发送成功，stat 为其它则表示发送失败，errDesc 为错误描述
2. sendResult 不为 0，则表示发送失败，不用看 stat，直接看 errDesc 错误描述

4.4.2.2.2. MO 短信通知-notifySmsReception 方法

功能：MO 短信通知。其 sms 参数说明如下

字 段	类 型	描 述	备 注
destID	String	MO 流水号	一般跟发送时的 xCode 对应
mobile	String	上行的手机号码	
content	ErrCode	短信内容	

msgFormat	MsgFmt	短信内容编码	ASCII: ASCII 字符。 UCS2G: USC2 格式的 UniCode 字符。 B18030: GB18030 格式的中文字符。 GB2312: GB2312 格式的中文字符。 Binary: 二进制短信，用十六进制字符串。
msgID	String	消息 ID	

4.4.2.2.3. 通知取消成功的手机号码-notifyCancelSmsList 方法

功能：通知取消成功的手机号码，其参数说明如下

字 段	类 型	描 述	备 注
submitID	String	提交到 MAS 服务平台标识	
cancelMobileList	List<String>	取消成功的手机号码集合	

4.4.2.3. 短信开发客户端代码举例

由图 4-1 所示，新建了 API 插件，API 插件标识为“API 插件”，API 插件密码“123123”，下面代码演示 API 客户端发送短信及相关方法。

短信开发客户端相关方法调用代码 example
<pre> SmsApiClient smsApiClient = null; long mobile = 138000000001; String appId="API插件";//API插件标识ID String appPwd="123123";//API插件密码 String content=发送API短信"; //短信内容 String xcode="222";//短信扩展码 int masApiPort=61616; String masIP="192.168.0.238"; //SmsApiClientHandlerImpl是由自己实现的 SmsApiClientHandler smsHandler = new SmsApiClientHandlerImpl(); try { smsApiClient = new SmsApiClientImpl(smsHandler, masIP, masApiPort, appId, appPwd); </pre>

```
//设置是否自动重连到服务器(可以不需要设置)
smsApiClient.setAutoConnect(true);

//设置自动重连服务器相隔时间(单位: 秒), 默认为30秒(可以不需要设置)
smsApiClient.setReConnectInterval(60);

//设置与服务连接超时时长, 单位: millisecond(可以不需要设置)
smsApiClient.setConnectTimeout(100000);

//设置发送超时时长, 单位: millisecond(可以不需要设置)
smsApiClient.setSendTimeout(1000000);

//获得短信群发提交的数量限制
int ret = smsApiClient.getDestAddrsLimit();

//获取网关连接状态(Connect: 连接正常, Disconnect: 断连, NotConnect: 没有连接, Other: 其他)
ConnectStatus connectStatus= smsApiClient.getConnStatusIAGW();
if (!ConnectStatus.Connect.equals(connectStatus))
{
    System.out.println("网关未连接");
    return;
}

//计算短信条数和字数
SmsCount smsCount= smsApiClient.getSmsCount(content, MsgFmt.GB2312,
SmsType.Normal);

//获得扩展服务代码(插件短信扩展号码+流水号)长度
int xcodeLength= smsApiClient.getXCodeLength()

//发送短信
//=====构造发送短信对象开始, 下面代码演示发送短信对象几个比较主要的属性值,
其它的属性可以不设置, 如果要设置可以参考sendSms方法中 SmsSendRequest参数=====
SmsSendRequest smsSendRequest = new SmsSendRequest();
List<String> list = new ArrayList<String>();

//每次群发数量不能超过最大限制数
for (int i = 0; i < ret; i++)
{
    long address = mobile + i;
    list.add(String.valueOf(address));
}
```

```
}
smsSendRequest.destAddrs = list;
smsSendRequest.validTime = 10000; // 短信存活期, 单位秒
smsSendRequest.xCode = xCode; //短信扩展码
smsSendRequest.message = content; //短信内容
smsSendRequest.msgFormat =MsgFmt.GB2312; //短信编码类型
smsSendRequest.isNeedReport = true; //短信是否需要状态报告
smsSendRequest.priority = 1; //短信网关优先级, 短信优先级大于0 的整数 0为最高优先级,
// 数字越大级别越低
// Normal: 普通短信,Instant: 免提短信, Long: 长短信,Structured: 二进制短信,WapPush:
//WapPush 短信
smsSendRequest.type = SmsType.Normal;
smsSendRequest.appID = appID;
//=====构造发送短信对象结束=====
SmsSendResponse smsSendResponse = smsApiClient.sendSms(smsSendRequest);
System.out.println("提交成功, requestID:" + smsSendResponse.requestID);
}
catch (Exception e)
{
    System.out.println("API短信客户端调用失败:" + e.getMessage());
    return;
}
```

如提交成功会收到 requestID,如有网络连接、密码错误等问题,则会抛相应的异常,相关异常信息请参考 4.4.4 API 异常错误对应常量。

SmsApiClientHandler 接口类主要是对短信服务的一些回调对象进行处理,如接收状态报告、MO 短信等,下面是对该类中一些回调方法的代码举例。

短信服务的一些回调方法调用代码 example

```
/**
 * 接收状态报告
 * @param deliveryInfo 状态报告对象
 */
public void notifySmsDeliveryStatus(String submitID, Report[] report)
{
    System.out.println ("接收到状态报告"+ report.length+": SmsReport
    submitID: " + submitID + " sendResul= " + report[0].sendResult);
}

/**
 * 接收MO短信
 * @param message
 * @return
 */
public void notifySmsReception(Sms sms)
{
}
```

```
System.out.println ("收到一条MO, destId: " + sms.destID + "; 内容: " +  
message.content + "手机号码: " + sms.mobile);  
}
```

4.4.3. 彩信接口

4.4.3.1. 彩信 API 客户端主要操作类-MmsApiClient 类

彩信 API 客户端主要操作接口，通过 ApiClientFactory 类来创建期实例，一个 MmsApiClient 对象代表了与服务器连接，在正常与服务器通信时需要先调其 start 方法来初始化。

4.4.3.1.1. 设置是否自动重连到服务器-setAutoConnect 方法

功能：设置是否自动重连到服务器。

4.4.3.1.2. 设置自动重连服务器相隔时间-setReConnectInterval 方法

功能：设置自动重连服务器相隔时间(单位：秒)，默认为 30 秒。

4.4.3.1.3. 设置与服务连接超时时长-setConnectTimeout 方法

功能：设置与服务连接超时时长，单位：millisecond，需要在调用 start 方法前设置方可生效。

4.4.3.1.4. 设置发送超时时长-setSendTimeout 方法

功能：设置发送超时时长，单位：millisecond，需要在调用 start 方法前设置方可生效。

4.4.3.1.5. 启动彩信 API -start 方法

功能：启动 API，该方法会自动连接到 MAS 服务器并进行登录。所有的发送、接收等操作都需要在 start 成功后方可使用。

4.4.3.1.6. 彩信 API 退出-loginOut 方法

功能：彩信 API 退出。

4.4.3.1.7. 获得网关连接状态-getConnStatusIAGW 方法

功能：获得网关连接状态。

4.4.3.1.8. 获得 MAS 彩信提交的限制数量-getDestAddrsLimit 方法

功能：获得 MAS 当前彩信提交的限制数量。

4.4.3.1.9. 发送彩信-sendMms 方法

功能：发送彩信。其 MmsSendRequest 参数说明如下

字 段	类 型	描 述	备 注
appId	String	插件 ID	API 插件标识 ID
destAddrs	String	目的地址，即手机号码集合,多个手机号码之间用英文“,”号隔开	一次发送的最大长度，具体可通过 getDestAddrsLimit 方法获得
xCode	String	彩信子码+流水号	
subject	String	彩信主题	
content	String	彩信内容	mime 文件格式，采用 BASE64 加密。
priority	MessagePriority	彩信网关发送优先级	Default Low Normal High
level	int	MAS 发送彩信优先级。大于等于 0 的正整数。数字越小优先级越高，0 为最高优先级，依次减小。	
isNeedReport	boolean	是否需要状态报告	
validTime	int	彩信存活期，单位秒	开始时间以 MAS 收到 API 提交时间为准，validTime 默认为 0 时，存活期为 24 小时。
返回值	MmsSendResponse	本次发送响应	

其 MmsSendResponse 参数说明如下

字 段	类 型	描 述	备 注
requestID	String	提交到 MAS 服务平台标识	对应 submitID
code	String	错误编码	
desc	String	错误描述	

4.4.3.1.10. 获取该次发送还未提交到彩信网关的数量-getStat 方法

功能：提交到 MAS 服务平台标识(submitID)来获取该次发送还未提交到彩信网关的数量。

字 段	类 型	描 述	备 注
submitID	String	提交到 MAS 服务平台标	对应 MmsSendResponse 中的 requestID

D		识	
返回值	int	获得该 submitID 提交到 MAS 的彩信，但还未提交到网关的数量	大批量发送时用于查看当前的进度

4.4.3.1.11. 取消彩信发送-cancelMms 方法

功能：根据发送标识取消彩信发送，已提交到网关的不能取消。

字 段	类 型	描 述	备 注
submitID	String	提交到 MAS 服务平台标识	对 应 MmsSendResponse 中的 requestID
返回值	List<String>	取消成功的手机号码	

4.4.3.2. 彩信服务回调对象进行处理类-MmsApiClientHandler 类

由企业应用来实现，该接口类主要是对彩信服务的一些回调对象进行处理，如接收状态报告、MO 彩信等。

4.4.3.2.1. 发送响应通知-notifyMmsSendResponse 方法

功能：发送响应通知。其 mmsSendResponse 说明如下

字 段	类 型	描 述	备 注
requestID	String	发送请求标识	sendMms 方法返回的值
submitID	String	提交到 MAS 服务平台标识	可以对应后续收到的状态报告
errCode	ErrCode	错误代码，为 null 表示发送成功	

4.4.3.2.2. 状态报告通知-notifyMmsDeliveryStatus 方法

功能：状态报告通知。如果在发送时指定不要状态报告将不会调用到此方法

字 段	类 型	描 述	备 注
submitID	String	提交到 EMA 服务平台标识	对应 MmsSendResponse 中的 requestID
reportList	Report	状态报告对象	

其 Report 参数说明如下：

字 段	类 型	描 述	备 注
errDesc	String	错误描述	当提交失败时，参数有效
submitResult	int	提交结果	成功为 1，失败为 0
delStatus	DeliveryStatus	发送状态	DeliveredToNetwork, DeliveryUncertain, DeliveryImpossible,

			MessageWaiting, DeliveredToTerminal, DeliveryNotificationNotSupported
address	String	消息接收地址	
sendTime	String	提交到网关的时间	
recDelTime	Date	收到状态报告时间	

4.4.3.2.3. M0 彩信通知-notifyMmsReception 方法

功能：MO 彩信通知。

字 段	类 型	描 述	备 注
message	MmsMessageReference	彩信对象	
content	String	消息内容	

其 MmsMessageReference 参数说明如下

字 段	类 型	描 述	备 注
messageID	String	彩信 ID, 如接收彩信为纯文本则此参数无效	
messageServiceActivationNumber	String	目的地址	
senderAddress	String	发送方地址	
subject	String	彩信的标题	
message	String	若接收彩信为纯文本, 则此参数为消息体内容, 当此参数存在时, messageID 参数无效。	
dateTime	Date	消息接收时间	
xCode	String	扩展号码	
mmsMessage	MmsMessage	彩信附件对象	

4.4.3.3. 彩信多媒体消息内容格式定义说明

4.4.3.3.1. 彩信多媒体内容类型

mas 提供的各种彩信发送接口, 彩信内容 (content) 一项均是字符串类型。

4.4.3.3.2. 彩信多媒体内容加密

彩信多媒体消内容必须经是过 BASE64 加密的字符串

4.4.3.3.3. 彩信多媒体内容编码

彩信多媒体消内容文本附件内容必须采用 utf-8 编码

4.4.3.3.4. 彩信多媒体内容支持附件的类型

text/plain, text/xml, application/smil, audio/amr, audio/midi, text/i-melody,
text/e-melody, image/gif, image/jpeg, image/vnd.wap.wbm, image/png

4.4.3.3.5. 彩信多媒体消息内容示例

以下为加密前内容示例，通过 mas 发送彩信必须将此内容采用 BASE64 加密后发送：

```
--SubPart_7452684322002_77645
Content-Type:application/smil
Content-Transfer-Encoding:8bit
Content-ID:20080815103000023.smil
Content-Location:20080815103000023.smil smil
```

附件内容……

```
--SubPart_7452684322002_77645
Content-Type:image/gif
Content-Transfer-Encoding:8bit
Content-ID:20080815102800026.gif
Content-Location:20080815102800026.gif
```

gif 附件内容……

```
--SubPart_7452684322002_77645
Content-Type:text/plain;charset=UTF-8
Content-Transfer-Encoding:8bit
Content-ID:200808151030000231.txt
Content-Location:200808151030000231.txt
```

文本附件内容……

```
--SubPart_7452684322002_77645--
----NextPart_0_2817_24856—
```

示例内容说明，内容中可以存在多个附件，每个附件前必须

“--SubPart_7452684322002_77645”开始，最后一个附件后加--SubPart_7452684322002_77645--
----NextPart_0_2817_24856—

附件内容属性说明

Content-Type 附件类型

Content-Transfer-Encoding 附件编码位数，默认采用 8 位编码

Content-ID 附件名称

Content-Location 附件名称

4.4.3.3.6. 彩信多媒体消息具体加密前后内容示例

加密前：

--SubPart_7452684322002_77645

Content-Type:application/smil

Content-Transfer-Encoding:8bit

Content-ID:20080815114700048.smil

Content-Location:20080815114700048.smil

```
<smil><head><meta name="title" content=""/><layout><root-layout width="100%"
height="100%" /><region id="txt_top" width="100%" height="16%" left="0" top="0"
fit="scroll"/><region id="txt_button" width="100%" height="16%" left="0" top="84%"
fit="scroll"/><region id="img_top" width="100%" height="84%" left="0" top="0"
fit="fill"/><region id="img_button" width="100%" height="84%" left="0" top="16%"
fit="fill"/><region id="all" width="100%" height="100%" left="0" top="0" fit="fill"
/></layout></head><body><par dur="10000ms"><text src="200808151147000481.txt"
region="all" /></par></body></smil>
```

--SubPart_7452684322002_77645

Content-Type:text/plain;charset=UTF-8

Content-Transfer-Encoding:8bit

Content-ID:200808151147000481.txt

Content-Location:200808151147000481.txt

HELLO MMS!

--SubPart_7452684322002_77645--

----NextPart_0_2817_24856--

加密后:

LS1TdWJQYXJ0Xzc0NTI2ODQzMjIwMDJfNzc2NDUNCkNvbnRlbnQtVHlwZTphcHBsaWNh
dGlubi9zbWlsDQpDb250ZW50LVRYYW5zMmVYLUVuY29kaW5nOjhiaXQNCkNvbnRlbnQtS
UQ6MjAwODA4MTUxMTQ3MDAwNDguc21pbA0KQ29udGVudC1Mb2NhdGlvbjoyMDA4M
DgxNTEwNDcwMDA0OC5zbWlsDQoNCjxzZbWlsPjxoZWFKPjxtZXRhIG5hbWU9InRpdGxIiB
jb250ZW50PSIiLz48bGF5b3V0Pjxyb290LWxheW91dCB3aWR0aD0iMTAwJSIgaGVpZ2h0PSI
xMDAlIiAgLz48cmVnaW9uIGlkPSJ0eHRfdG9wLiB3aWR0aD0iMTAwJSIgaGVpZ2h0PSIxNiUi
IGxlZnQ9IjAiIHRvcD0iMCIgZml0PSJzY3JvbGwiLz48cmVnaW9uIGlkPSJ0eHRfYnV0dG9uLi
B3aWR0aD0iMTAwJSIgaGVpZ2h0PSIxNiUiIGxlZnQ9IjAiIHRvcD0iODQlIiBmaXQ9InNjc9s
bCIvPjxyZWdpb24gaWQ9ImltZ190b3AiIHdpZHRoPSIxMDAlIiBoZWlnaHQ9Ijg0JSIgbGVmd
D0iMCIgdG9wPSIwLiBmaXQ9ImZpbGwiLz48cmVnaW9uIGlkPSJpbWdfYnV0dG9uLiB3aWR0
aD0iMTAwJSIgaGVpZ2h0PSI4NCUiIGxlZnQ9IjAiIHRvcD0iMTYlIiBmaXQ9ImZpbGwiLz48c
mVnaW9uIGlkPSJhbGwiIHdpZHRoPSIxMDAlIiBoZWlnaHQ9IjEwMCUiIGxlZnQ9IjAiIHRvc
D0iMCIgZml0PSJmaWxsLiAvPjwvbGF5b3V0PjwvaGVhZD48Ym9keT48cGFyIGR1cj0iMTAw
MDBtcyI+PHRleHQgc3JjPSIyMDA4MDgxNTEwNDcwMDA0ODEudHh0IiByZWdpb249ImFs

```
bCIgIz48L3Bhcj48L2JvZHK+PC9zbWlsPg0KDQoNCi0tU3ViUGFydF83NDUyNjg0MzIyMDA
yXzc3NjQ1DQpDb250ZW50LVR5cGU6dGV4dC9wbGFpbjtaGFyc2V0PVVURi04DQpDb250
ZW50LVRyYW5zZmVyLUVuY29kaW5nOjhiaXQNCkNvbRlbnQtSUQ6MjAwODA4MTUxM
TQ3MDAwNDgxLnR4dA0KQ29udGVudC1Mb2NhdGlvbjoyMDA4MDgxNTEwNDcwMDA0O
DEudHh0DQoNCkhFTExPIE1NUyENCg0KDQotLVN1Y1BhcnRfNzQ1MjY4NDMyMjAwMl83
NzY0NS0tDQotLS0tTmV4dFBhcnRfMF8yODE3XzI0ODU2LS0NCg==
```

4.4.3.4. 彩信开发客户端代码举例

由图 4-1 所示,新建了 API 插件, API 插件标识为“API 插件”, API 插件密码“123123”,下面代码演示 API 客户端发送彩信及相关方法。

彩信开发客户端相关方法调用代码 example

```
MmsApiClient mmsApiClient = null;
String appID="API插件";//API插件标识ID
String appPwd="123123";//API插件密码
long mobile = 13800000000L;
String xcode="222";//彩信扩展码
String version="3.0";
int masApiPort =61617;
String masIP="192.168.0.238";

//MmsApiClientHandlerImpl是自己实现的
MmsApiClientHandler mmsHandler = new MmsApiClientHandlerImpl();
try
{
mmsApiClient = new MmsApiClientImpl(apiID, appPwd, version, masIP,
masApiPort, mmsHandler);

//设置是否自动重连到服务器(可以不需要设置)
mmsApiClient.setTimingConnect(true);

//设置自动重连服务器相隔时间(单位: 秒), 默认为30秒(可以不需要设置)
mmsApiClient.setConnectTime(60);

//设置与服务连接超时时长, 单位: millisecond(可以不需要设置)
mmsApiClient.setConnectTimeout(100000);

//设置发送超时时长, 单位: millisecond(可以不需要设置)
mmsApiClient.setSendTimeout(1000000);

int ret = mmsApiClient.getDestAddrsLimit();//获得彩信群发提交的数量限制

//获取网关连接状态(Connect: 连接正常, Disconnect: 断连, NotConnect: 没有连
```

```
接, Other: 其他)
ConnectStatus connectStatus= mmsApiClient.getConnStatusIAGW();
if (!ConnectStatus.Connect.equals(connectStatus))
{
    System.out.println("网关未连接");
    return;
}

//发送彩信
//=====构造发送彩信对象开始, 下面代码演示发送彩信对象几个比较主要的属性值,
其它的属性可以不设置, 如果要设置可以参考sendMms方法中 MmsSendRequest参数===
MmsSendRequest mmsSendRequest = new MmsSendRequest();
List<String> list = new ArrayList<String>();
StringBuilder addressStr = new StringBuilder();
//每次群发数量不能超过最大限制数
for (int i = 0; i < ret; i++)
{
    long address = mobile + i;
    list.add(String.valueOf(address));
}
mmsSendRequest.destAddrs = list;
mmsSendRequest.appID = appID;
// “彩信内容” 格式请参照本手册中的彩信多媒体消息内容格式定义说明
mmsSendRequest.content = "... .." //彩信内容
mmsSendRequest.isNeedReport = true; // 是否需要状态报告
mmsSendRequest.subject = "彩信发送标题"; //彩信主题
mmsSendRequest.validTime = 1000000; // 彩信存活期, 单位秒
mmsSendRequest.priority = MessagePriority.Default; // 彩信网关发送优先级
mmsSendRequest.xCode = xcode; //彩信扩展码
//=====构造发送彩信对象结束=====

MmsSendResponse mmsSendResponse =
mmsApiClient.sendMms(mmsSendRequest);
System.out.println("提交成功, requestID:" + mmsSendResponse.requestID);

}
catch (Exception e)
{
    System.out.println("API彩信客户端调用失败:" + e.getMessage());
    return;
}
```

如提交成功会收到 **requestID**, 如有网络连接、密码错误等问题, 则会抛相应的异常。相关异常信息请参考 4.4.4 API 异常错误对应常量。

MmsApiClientHandler 接口类主要是对彩信服务的一些回调对象进行处理,如接收状态报告、MO 彩信等,下面是对该类中一些回调方法的代码举例。

彩信服务的一些回调方法调用代码 example
<pre> /** * 接收状态报告 * @param report 状态报告列表 * @return */ public void notifyMmsDeliveryStatus(String requestID, Report[] reportList) { System.out.println ("接收到状态报告"+ reportList.length +"条: MmsReport requestID: " + requestID + " sendResul= " + reportList [0]. submitResult); } </pre>
<pre> /** * 接收MO信息 * * @param message 与接收消息相关的所有信息 * @param content 与消息一起接收到的数据。 * */ public void notifyMmsReception(MmsMessageReference message, String content) { System.out.println ("收到一条MO, messageID:"+ message.messageID +" xCode:"+ message.xCode +" senderAddress:"+ message.senderAddress +" dateTime:"+message.dateTime + "subject:"+message.subject +" content:"+content); } </pre>

4.4.4. API 异常错误对应常量

错误码	错误描述	可能导致错误的原因
ERR100	登陆出现异常错误	程序出现运行异常
ERR101	服务器连接失败	API 服务器没有启动或者启动失败
ERR102	APIID 不存在	传入的 APIID 在服务端没有找到
ERR103	API 密码错误	传入的 API 密码与服务端不一致
ERR104	API 处于暂停状态	服务端 API 插件没有运行
ERR105	API 短信业务能力没有开通	服务端 API 没有开通短信业务能力
ERR106	API 彩信业务能力没有开通	服务端 API 没有开通彩信业务能力

ERR107	设置与服务连接超时时长不能为负数	设置与服务连接超时时长只能是正整数
ERR108	发送超时时长不能为负数	设置发送超时时长只能是正整数
ERR109	API 退出出错	API 退出时可能出现运行时异常
ERR110	API 客户端没有登陆验证或登陆验证失败	在使用 API 其它方法之前没有调用 API 的 start 方法或者调用不成功
ERR111	无效版本号	传入的 API 版本在服务端没有找到
ERR112	服务端没有响应或者响应超时	网络超时导致
ERR113	设置与服务自动重连相隔时间不能为负数	设置与服务自动重连相隔时间只能是正整数

5. 接口描述

5.1. 短信接口描述

5.1.1. 接口说明

集团客户应用系统和应用插件通过此接口收发短消息。API 方式通信适配插件收到应用插件的发送请求后，通过 MAS 服务器基座中的 SMS 通信协议模块发送 SMS；当从 SMS 通信协议模块收到 SMS 后，通过此接口将短消息发送给集团客户应用系统和应用插件。

5.1.2. 流程说明

短消息接口主要包括短信发送、接收、和取消三个流程。

短信发送步骤说明：

- 企业应用通过 smsSend 发送短信操作向 MAS 的 API 适配模块提交要发送的短信，smsSend 方法会返回发送响应。
- MAS 服务器收到消息后，把提交结果返回给 API 客户端
- MAS 服务器将短信提交到网关，并将状态报告传给服务器的 API 适配模块
- API 适配模块将状态报告传给 API 客户端的回调对象（注，在发送时是否需要状态报告字段要填 true）

短信接收步骤说明：

- 行业网关收到手机上行的短信，将 MO 短信通知到 MAS 服务器
- MAS 服务器将 MO 短信交到 API 适配模块
- API 适配模块将 MO 交到 SmsApiClient
- SmsApiClient 将 MO 短信回调通知到 SmsApiClientHandler 的 notifySmsReception 接口

短信取消步骤说明：

- 企业应用通过 SmsApiClient 向 MAS 服务器发起取消请求，并到一个请求是否提交成功的结果
- MAS 服务器将已取消成功功能的手机号码列表通知给 API 客户端。

5.1.3. 主要类说明

类 名	描 述	备 注
ApiClientFactory	工厂方法类，主要用于创建 SmsApiClient 与 MmsApiClient	
SmsApiClient	短信客户端接口，一个 Client 代表与 MAS 服务端的一个连接	可以创建多个 Client 对象来发送短信
SmsApiClientImpl	短信客户端的实现类	
SmsApiClientHandler	短信 API 回调处理接口	由企业应用来实现，一个 Client 对象至少关联一个 Handler 实例
SmsSendRequest	短信发送请求参数类	
SmsSendResponse	短信发送响应类	
Sms	MO 短信表示对象	
Report	发送回执表示对象	

5.2. 彩信接口描述

5.2.1. 接口说明

应用系统或应用插件调用发送接口给目的用户发送一条彩信。通信适配插件采用非同步的通知机制将彩信提交状态通知给应用系统或应用插件。对于彩信接收，可以由通信适配插件主动通知应用系统或应用插件有彩信到达，应用系统或应用插件也可通过接口从通信适配插件接收彩信。

5.2.2. 流程说明

彩信发送流程步骤说明：

- 企业应用通过 mmsSend 发送短信操作向 MAS 的 API 适配模块提交要发送的短信，mmsSend 方法会返回一个发送响应。
- MAS 服务器收到消息后，把提交结果返回给 API 客户端
- MAS 服务器将短信提交到网关，并将状态报告传给服务器的 API 适配模块
- API 适配模块将状态报告传给 API 客户端的回调对象（注，在发送时是否需要状态报告字段要填 true）

彩信 MO 流程步骤说明：

- 行业网关收到手机上行的彩信,将 MO 彩信通知到 MAS 服务器
- MAS 服务器将 MO 彩信交到 API 适配模块
- API 适配模块将 MO 交到 MmsApiClient
- MmsApiClient 将 MO 彩信回调通知到 MmsApiClientHandler 的 notifyMmsReception 接口。

取消彩信发送流程步骤说明：

- 企业应用通过 MmsApiClient 向 MAS 发起取消请求，并到一个请求是否提交成功的结果
- MAS 服务器将得到已取消成功功能的手机号码列表通知给客户端

5.2.3. 主要类说明

类 名	描 述	备 注
ApiClientFactory	工厂方法类，主要用于创建 SmsApiClient 与 MmsApiClient	
MmsApiClient	彩信客户端接口，一个 Client 代表与 MAS 服务端的一个连接	可以创建多个 Client 对象来发送彩信
MmsApiClientImpl	彩信客户端的实现类	
MmsApiClientHandler	彩信 API 回调处理接口	由企业应用来实现，一个 Client 对象至少关联一个 Handler 实例
MmsSendRequest	彩信发送请求参数类	
MmsSendResponse	彩信发送响应类	
MmsMessageReference	MO 彩信表示对象	
Report	发送回执表示对象	

6. API 短信 DEMO 使用说明

短信API的DEMO主要提供了启动API、获得短信网关连接状态、获得短信群发流量限制、发送短信、接收短信发送状态、短信状态报告、接收MO短信等功能。

6.1.1. 使用说明

启动 APIDemo 类后，输入 start 命令启动 API，成功后其它的功能才可以使用。
除短信状态报告以及短信 MO 是主动通知，其它的功能都是命令行输入来进行操作实现。

6.1.2. 主要命令说明

start命令:

功能: 启动API, 所有的发送、接收等操作都需要在start成功后方可使用。

命令格式: start <服务器IP地址> <服务器API端口> <API标识ID> <API密码> [短信扩展码]

其中<>中的参数表示必输入的, []表示可选参数, 假如输入的命令不符合标准, 则会提示start命令的输入格式。

如果不清楚start命令的输入格式, 可以在命令行输入 help start 来进行查看。

connstate命令:

功能: 获得短信网关连接状态。

命令格式: connstate

如果不清楚connstate命令的输入格式, 可以在命令行输入 help connstate来进行查看。

limit命令:

功能: 获得短信群发流量限制。

命令格式: limit

如果不清楚limit命令的输入格式, 可以在命令行输入 help limit来进行查看。

send命令:

功能: 发送短信。

命令格式: send <手机号码> <短信内容> [发送次数]

其中<>中的参数表示必输入的, 假如输入的命令不符合标准, 则会提示send命令的输入格式。[]表示可选参数, 假如输入格式不正确, 则当作内容一起发送。

如果不清楚send命令的输入格式, 可以在命令行输入 help send来进行查看。

sendwappush 命令 :

功能: 发送wappush短信。

命令格式: sendwappush <手机号码> <主题> <推送地址> [发送次数]

其中<>中的参数表示必输入的, 假如输入的命令不符合标准, 则会提示sendwappush命令的输入格式。[]表示可选参数, 假如输入格式不正确, 则当作内容一起发送。

如果不清楚sendwappush命令的输入格式, 可以在命令行输入 help sendwappush来进行查看。

help命令:

功能: 发送短信。

命令格式: help

exit命令:

功能: 退出。

命令格式: exit

quit 命令:

功能: 退出。

命令格式: quit

7. 注意事项

- 要想成功发送短信还需要 MAS 服务平台开通相应的业务能力，并且 MAS 已正常连接到短信网关。
- Mas 服务器的防火墙需要打开 61616、61617 端口，才能保证 API 短信和彩信发送功能正常工作。
- 如果与 MAS 服务器通讯异常，请确认 MAS 服务上 61616、61617 端口监听是否常，可以通过 API 所在的客户端来 telnet 测试
- SmsApiClient、MmsApiClient 里有一些参数可以调节，如连接超时时间、发送超时时间等，这些参数都需要在调用 start 方法之前设置才能生效。

8. 参考文档

<< MAS2.0 安装手册.doc>>

<< MAS2.0 二次开发手册(总则).doc>>