

# 金笛短信中间件 Web 版

## 开发接口说明文档



**金笛短信**  
www.sendsms.cn

## 目录

<b>第一章 快速使用</b> .....	3
HTTP 接口快速使用 .....	3
数据库接口快速使用 .....	4
回调接口快速使用 .....	4
<b>第二章 数据库接口</b> .....	6
数据库表结构.....	6
smsserver_in 表结构.....	6
smsserver_calls 表结构.....	6
smsserver_out 表结构.....	7
数据库接口使用 .....	8
数据库连接性测试 .....	8
插入短信数据.....	8
<b>第三章 HTTP 接口</b> .....	9
使用 HTTP 接口发送短信 .....	9
发送接口说明 .....	9
发送接口演示 .....	10
发送状态查询 .....	11
发送接口调试案例 .....	11
使用 HTTP 接口读取短信 .....	15
读取接口说明 .....	15
<b>第四章 回调接口</b> .....	16
开启回调接口.....	16
回调接口配置.....	16
通过配置文件更改 .....	17
<b>第五章 邮件接口</b> .....	18
配置中继邮箱.....	18
修改邮件接口配置 .....	19
指定邮件转短信方式 .....	20
测试配置是否生效 .....	21
<b>第六章 常见问题</b> .....	22
配置文件的位置 .....	22
配置文件无法更改 .....	22
重启中间件服务的方法 .....	23
中间件无法发出短信 .....	24
版本更新后能发短信，但无法收到短信 .....	24

# 第一章 快速使用

本章节将简要描述接口使用方法，详细使用说明请查看各章节详细介绍。

## 1、HTTP 接口快速使用

位置：**【中间件管理界面】 - 【配置与日志】 - 【系统设置】**

默认配置如下图，HTTP 接口的端口为 8060



示例：

\*如果没有这个链接，您可以按照这个 URL 的形式构造一个 URL 来测

<https://192.168.0.235:8060/send?password=1&recipient=10086&text=ceshi&encoding=U>

示例中 URL 各个参数的作用如下图



参数中：`password` 为发送密码，默认为 1，`recipient` 为收件人手机号码，`text` 为短信内容，`encoding` 为 `text` 的字符编码，默认为 U(UTF-8)，使用时将 `recipient` 和 `text` 的参数值替换即可

注意事项：

1. HTTP 接口的 HTTPS 加密 (SSL 协议) 默认是开启的，需要在 URL 最前面添加 HTTPS 访问这个网址或者关闭 HTTPS 加密后使用 HTTP 访问。
2. 使用 HTTPS 时若报错或显示证书问题，请关闭发送程序的 HTTPS 证书校验，如果没有 HTTPS 相关需求，建议关闭 HTTPS 加密，减少调试过程中可能遇到的麻烦。
3. Text 的内容中如果有中文或空格、换行等字符，请对 Text 的内容使用 URLEncode 编码，有关 URLEncode 的使用方法请看第三章详细介绍。

如果使用接口添加了请求，但没有收到短信，请查看【常见问题】章节中“短信无法发出”项目来排查问题

## 2、数据库接口快速使用

位置：【中间件管理界面】 - 【配置与日志】 - 【数据源配置】

在此处可以确定短信中间件当前连接的数据库类型以及地址参数：



程序成功连接到数据库后，添加如下 SQL 语句即可将短信数据提交到中间件

```
insert into smsserver_out (recipient,text,encoding,create_date) values ('13800138000','短信中间件数据库测试','U', '2021-10-08 16:07:44');
```

recipient 为收件人手机号,text 为短信内容,encoding 为 text 的字符编码,默认为 U(UTF-8)

发送时间需要使用 datetime 格式，系统支持时可以用 SQL 自带的 now()函数来自动填充

datetime 格式 'YYYY-MM-dd HH:mm:ss' 即

'2021-10-08 16:07:44'

如果使用接口添加了请求，但没有收到短信，请查看下方【常见问题】章节“短信无法发出”项目排查问题

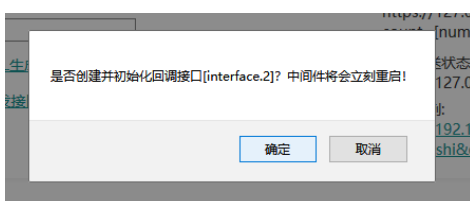
## 3、回调接口快速使用

位置：【中间件管理界面】 - 【配置与日志】 - 【系统设置】

\*此演示适用于高于 3.6.9-0915 版本的 WEB 中间件，若低于此版本，请安装新版本后操作



点击【启用回调】初始化回调接口并重启中间件：



重新登录中间件后，会在设置中显示回调接口调用的 URL

## 回调接口设置

回调URL

```
http://192.168.0.235:8090  
/test.jsp?gatewayId=%gatewayId%&text=%text%&  
originator=%originator%
```

回调失败重试间隔

3600

秒

回调失败天数间隔

14

日

测试工具

[点击发送回调请求](#)

保存

重置

可以看到生成了新的回调 URL

```
http://192.168.0.235:8090/test.jsp?gatewayId=%gatewayId%&text=%text%&originator=%originator%
```

此时，中间件在收到短信后，中间件会自动访问此 URL，即可用此特性被动接收短信。

回调 URL 参数中 `%gatewayId%`、`%text%`、`%originator%` 是短信参数，分别是设备号、短信内容、发件人手机号，中间件会自动替换其内容

例如，服务器为接收地址为 `192.168.0.115/recv` 时，即可构造回调 URL

```
http://192.168.0.115/recv?msgfrom=%originator%&msg=%text%
```

更改完请点击测试工具发送回调请求验证是否能正常发送，否则请检测网络连接

## 第二章 数据库接口

本章节介绍中间件使用的表结构以及如何使用数据库接口添加短信

### 数据库表结构

Web 中间件使用的数据库有七张表，默认皆以 `smsserver_` 作为前缀，各表说明如下

表名	说明	解释
SMSSERVER_ADDRERCORD	添加记录	最新版未使用
SMSSERVER_ALERT	错误记录	最新版未使用
SMSSERVER_CALLS	来电记录	用于记录短信猫未接来电
SMSSERVER_IN	短信接收表	接收短信
SMSSERVER_OUT	短信发送表	待发送的短信插入此表，发送完成后自动更改状态位
SMSSERVER_TMPMSG	收信临时表	收短信用临时表，升级后可清除，版本变化：368版添加1个字段
SMSSERVER_USER	用户账户表	存有中间件的登录账户信息

外部调用一般只会用到 `smsserver_in`、`smsserver_out`、`smsserver_calls` 三个表，接下来详细介绍这三个表

#### smsserver\_in 表结构

短信接收表，用于查询中间件收到的信息

字段名	类型	属性	描述
ID	int (10)	自增	主键
PROCESS	int (11)	Not Null	预留备用
ORIGINATOR	char (25)	Not Null	发件人号码
TYPE	char (1)	Not Null	短信类型: "I"-普通短信, "S"-短信状态报告
ENCODING	char (1)	Not Null	编码格式: "7" - 7 bit "8" - 8 bit "U" - Unicode/UCS2
MESSAGE_DATE	datetime	Not Null	短信发送日期时间(短信中附带的发送时间)
RECEIVE_DATE	datetime	Not Null	短信接收日期时间(本机接收时间)
TEXT	char (1000)	Not Null	短信正文
ORIGINAL_REF_NO	char (64)	Null	用于短信状态报告: 此号码对应于外发的短信序号
ORIGINAL_RECEIVE_DATE	datetime	Null	用于短信状态报告: 对方收到短信的日期时间
GATEWAY_ID	char (64)	Not Null	通道标识, 用于标识是哪个通道收到的短信

#### smsserver\_calls 表结构

来电记录表

字段名	类型	属性	描述
ID	int (10)	自增	主键
CALL_DATE	datetime	Not Null	来电时间
GATEWAY_ID	char (64)	Not Null	通道标识
CALLER_ID	char (16)	Not Null	对方号码

## smsserver\_out 表结构

短信发送表，发短信主要使用的表，默认使用 UTF-8 编码

字段名	类型	主要属性	其他属性	描述
ID	int (10)	主键	自增	数据库自动管理。如非必要，请勿赋值。
TYPE	char (1)	Not Null	默认值 "O"	短信类型: "O" - 普通短信 (英文大写O) "W" - WAP短信。
RECIPIENT	char (25)	Not Null	必填	收件人手机号码，前面不要加 "0" 或 "+"，发给座机请带区号。
TEXT	char (1000)	Not Null	必填	短信正文，收到短信乱码时请查看编码字段设置是否正确
WAP_URL	char (100)	Null		WAP 短信的 URL 地址
WAP_EXPIRY_DATE	datetime	Null		WAP 短信的有效期，默认7天
WAP_SIGNAL	char (1)	Null		"N" NONE, "L" LOW, "M" MEDIUM, "H" HIGH, "D" DELETE
CREATE_DATE	datetime	Not Null	必填	短信插入数据库的时间'YYYY-MM-DD HH:mm:ss'
ORIGINATOR	char (16)	Not Null	默认值 ""	发件人，默认值正常情况下为空字符串
ENCODING	char (1)	Not Null	必填 默认值 "7"	短信正文编码格式: "7" - 7 bit, "8" - 8 bit, "U" - Unicode/UCS2 中文短信请置为"U",此时70个汉字算一个短信长度
STATUS_REPORT	int (1)	Not Null	默认值 0	状态报告，设为 1需要短信状态报告，默认为0
FLASH_SMS	int (1)	Not Null	默认值 0	闪信，设为1 发送闪信，默认为0
SRC_PORT	int (6)	Not Null	默认值 -1	彩铃源端口
DST_PORT	int (6)	Not Null	默认值 -1	彩铃目的端口
SENT_DATE	datetime	Null		用于定时发送，若未指定，中间件会记录实际发出的时间
REF_NO	char (64)	Null		短信猫内部 ID
PRIORITY	int (5)	Not Null	默认值 0	优先级: 低-负数 普通-0 高-正数
STATUS	status (1)	Not Null	默认值 "U"	短信状态: "U": 未发送, "Q": 排队中, "S": 已发送, "F": 失败, "Z": 终止
ERRORS	int (2)	Not Null	默认值 0	当前重发次数，可以在数据源设置中配置
GATEWAY_ID	char (64)	Not Null	默认值 "**"	短信通道标识，可指定短信发送通道 ** 自动选择设备发送 "# 使用网关通道发送 "\$ 自动选择手机发送 "modem1" 设备编号 <sup>[1]</sup> modem1的设备发送 置空时，由短信服务器自动选择
USER_ID	char (20)	Null		用户标识，用于识别进行发信操作的账号 (于3.6.7版本新增，出现版本迁移后请检查此字段)

注[1]: 设备编号在【短信设备列表】中点击序号或端口号可查到，可能会不同于短信设备页面所显示序号，以【短信猫参数】页面显示设备编号为准

其他说明:

标记为必填的 recipient, text, encoding, create\_date 4 个字段为最基础的参数，其他可以视需求选填

## 数据库接口使用

在开始使用数据库前，请确认中间件已经成功在数据库中创建默认的 7 个表，并且可以正常连接到数据库

### 数据库连接性测试

在控制台管理界面选择【短信发送】，随意发送一条短信，看短信是否能正确在【待发送列表】或【短信记录】中显示：

如果可以正常显示新添加的短信，说明数据库连接性正常，可以继续进行接下来的测试。

如果无法添加到待发送列表，一般是数据库连接的问题，请参阅【常见问题】-数据库连接失败

### 插入短信数据

使用如下 SQL 语句，可以插入短信需要发送的短信，

发送短信

```
insert into smsserver_out
(recipient,text,encoding,create_date)
values
('13601019694','短信中间件数据库测试','U', '2021-10-08 13:00:00');
```

发送短信（使用 SQL 自带的 now()函数自动填充时间）

```
insert into smsserver_out
(recipient,text,encoding,create_date)
values
('13601019694','短信中间件数据库测试','U', now());
```

指定短信设备 modem2 发送短信

```
insert into smsserver_out
(recipient,text,encoding,create_date,gateway_id)
values
('13601019694','短信中间件数据库测试','U', '2021-10-08 13:00:00','modem2');
```

指定用手机 APP 发送短信

```
insert into smsserver_out
(recipient,text,encoding,create_date,gateway_id)
values
('13601019694','数据库发送测试','U', '2021-10-08 13:00:00','$');
```

指定用网关发送短信

```
insert into smsserver_out
(recipient,text,encoding,create_date,gateway_id)
values
('13601019694','jdsmsserver 数据库发送测试','U', '2021-10-08 13:00:00','#');
```



## 第三章 HTTP 接口

HTTP 接口可以用于发送短信、查看接收到的短信，完全通过 URL 方式调用，客户可以通过 Java\C++\C#\Python\Shell 等多种语言通过网络调用此接口，接口开发方便并且管理省心。

### 使用 HTTP 接口发送短信

#### 发送接口说明

标准的调用 URL 如下

`https://IP:Port/send?password=[发送密码]&text=[短信内容]&recipient=[手机号]&encoding=U`

中间件在初次安装时默认使用 SSL(即 https)，端口、协议（SSL）、发送密码等参数可以在设置页面更改

发送参数	含义	说明	示例
IP	必填	服务器IP	保证IP可达、8060端口开放即可 192.168.0.6
PORT	必填	中间件端口	以实际设置中端口号为准 8060
PASSWORD	必填	发送密码	默认值:1 123456
TEXT	必填	发送内容	中文内容需要URLEncode之后发送 *详见常见问题
RECIPIENT	必填	收件人号码	可群发，用逗号分隔每个号码 13800138001,13800138002
ENCODING	必填	字符集编码	可选:U-UTF8 G-GBK U
GATEWAY	选填 默认值*	发送通道	从哪一个通道发出 可选"# "\$" "modem*" *-系统自动分配 #-网关发送 \$-手机发送 modem*-短信猫发送(*代表不指定)
PRIORITY	选填	优先级	低: 负数 普通: 0 高: 正数 10
FROM	选填	发送者名称	部分运营商支持，只针对 GSM modem

URLEncode 中，%0D%0A 是回车换行，%20 是空格，可以在短信内容中直接添加%0D%0A 来换行

#### 示例 1: GET 方式发送

`https://192.168.0.233:8060/send?password=1&text=你好%20 黄先生%0D%0A 请注意今晚 5 点开会  
&recipient=13800138000&encoding=U`

可以直接在浏览器访问此 URL 来测试 http 接口功能是否正常

如果 https 显示错误，请到【系统设置】-【HttpServer 短信收发接口设置】中，关闭 HTTPS 加密（旧版为 SSL）

**HttpServer短信收发接口设置**

端口

协议类型  启用HTTPS加密

#### 示例 2: POST 方式发送

参数	内容
目标网址	<code>https://192.168.0.233:8060/send</code>
数据形式	<code>application/x-www-form-urlencoded</code>
数据内容	<code>password=1&amp;recipient=10086&amp;text=%E4%BD%99%E9%A2%9D%E6%9F%A5%E8%AF%A2&amp;encoding=U</code>

组合成 HTTP 请求后

```

POST /send HTTP/1.1
Host: 192.168.0.235:8060
Content-Transfer-Encoding: application/x-www-form-urlencoded
Content-Type: application/x-www-form-urlencoded
Content-Length: 79

password=1&recipient=10086&text=%E4%BD%99%E9%A2%9D%E6%9F%A5%E8%AF%A2&encoding=U
    
```

## 发送接口演示

按照表中示例选项组建一个包含所有选项的请求，

```
https://192.168.0.6:8060/send?password=1&text=%E4%BD%A0%E5%A5%BD&recipient=13800138001,13800138002&encoding=U&gateway=modem6&priority=10&from=Jindi
```

- 注：
- (1) **红底白字**为必填参数，**青底白字**为自选参数
  - (2) 请参考接口设置决定是否需要使用 https
  - (3) 接口默认密码为 1

正确调用时的返回值(此时还未正式发送短信，但请求已收入到数据库中)：

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <send>
    <error>0</error>
    <message_status>UNSENT</message_status>
    <recipient>13800138001</recipient>
    <sent_date>2021-09-10T09:56:18+00:00</sent_date>
    <submit_date>2021-09-10T09:56:18+00:00</submit_date>
    <ref_no></ref_no>
    <msg_id>6</msg_id>
    <gateway>modem6</gateway>
  </send>
  <send>
    <error>0</error>
    <message_status>UNSENT</message_status>
    <recipient>13800138002</recipient>
    <sent_date>2021-09-10T09:56:18+00:00</sent_date>
    <submit_date>2021-09-10T09:56:18+00:00</submit_date>
    <ref_no></ref_no>
    <msg_id>7</msg_id>
    <gateway>modem6</gateway>
  </send>
</result>
```

### 特殊说明

1. 此 xml 文档默认使用 UTF-8 编码
2. 无论请求结果是什么，所有结果都将在 <result></result> 标签中间显示
3. 如一个请求中有多个收件人，每个手机号的短信都会生成一个 <send> 标签，请确认返回值中标签数量是否足够
4. <message\_status> 中 **显示 UNSENT** 表示请求**已成功加入发送队列**，是指目前还未发送，并不说明发送失败
5. <msg\_id>返回的是短信插入到数据库后获得的索引号，可用于短信发送状态查询

属性	含义	备注
ERROR	错误代码	0表示发送成功 错误代码一般为-9000~-9999
MESSAGE_STATUS	短信当前的发送状态	UNSENT – 未发送, SENT – 已发送, FAILED – 发送失败
RECIPIENT	短信收件人	用于识别短信归属，及多条短信区分收件人
SENT_DATE	定时发送时间	如果发送时间在过去，中间件会直接发送此短信； 如果时间还未到，此短信将在预计的时间到达后发出
SUBMIT_DATE	此请求的提交时间	此请求记录到数据库的时间,与create_date一致 请留意检查服务器时间与数据库时间是否正确
REF_NO	短信猫内部参数	短信猫内部对此短信的编号
MSG_ID	此短信唯一识别码	与数据库中ID一致，使用方法见[查看发送状态]
GATEWAY	实际发送通道	*表示系统自动选择 详细说明请看上方 <a href="#">smsserver_out</a> 表中gateway_id字段中说明

## Error 值对应的错误:

```

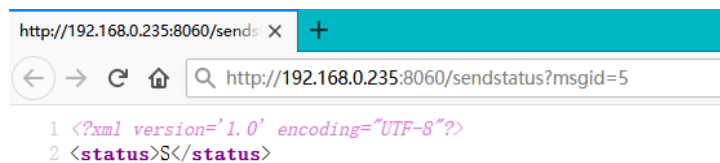
0      => ' ' ,/* 请求无错误, 已将短信请求提交到中间件中*/
-9000 => '[SMS] sender illegal param',/* 参数非法*/
-9001 => '[SMS] sender lack of param',/* 缺少参数*/
-9002 => '[SMS] send error' ,/* 发送错误*/
-9003 => '[SMS] wrong password' ,/* 密码错误*/
-9999 => '[SMS] sending aborted' ,/* 发送终止*/
  
```

## 发送状态查询

短信发送状态查询使用的 URL:

```
http://127.0.0.1:8060/sendstatus?msgid=[msgid]
```

将 msgid 的值替换成 <msg\_id>7</msg\_id> 中的 7, 即可查询示例中提交的短信的发送状态



短信状态: "U": 未发送, "Q": 排队中, "S": 已发送, "F": 失败, "Z": 终止

## 发送接口调试案例

以下将演示如何通过各种调试途径给 10086 发送内容为 ceshi 的短信

默认中间件参数: 服务器 IP-192.168.0.235, http 接口的端口-8060

### (1) 使用浏览器调试

填写四个必要参数 password、recipient、text、encoding, 构造新的 URL 如下

```
http://192.168.0.235:8060/send?password=1&recipient=10086&text=ceshi&encoding=U
```

访问该网址, 即可给 10086 发送内容为 ceshi 的短信, 返回结果如下



返回 message\_status 状态值为 UNSENT 和 SENT、并且 error 为 0 时, 表明中间件已收到请求, 短信已成功加入到待发送队列。

```

<?xml version="1.0" encoding="UTF-8"?>
<result>
  <error>-9003</error>
  <error_description>Invalid password.</error_description>
</result>
  
```

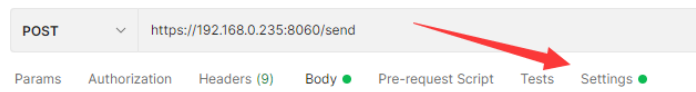
error 的值为非零时说明出了错误  
具体错误请查看 Error 值含义表

如果返回其他值, 请查看 error 是否  
是否为 0, 如出现其他值, 请检查

上方 Error 值含义表。

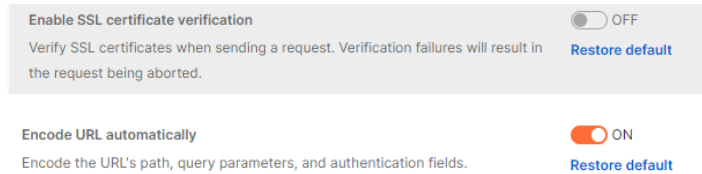
## (2) 使用 Postman 调试 (POST 方式发送)

开始之前，需要对 Postman 的以下设置进行更改



关闭 Enable SSL certificate verification (开启 SSL 证书校验)

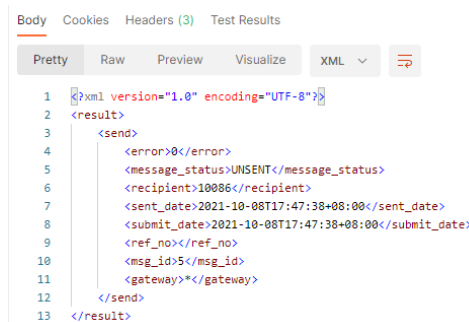
打开 Encode URL automatically (自动对 URL 进行编码，对中文内容生效)



本案例演示使用 POST 方式，以 HTTPS 加密的形式发送短信，在 body 中以 X-www-form-urlencoded 形式填写四个必要参数 password、recipient、text、encoding，如下图



返回值如下



返回 UNSENT 和 SENT 说明发送成功，发送失败请查看下方【常见问题】查找解决办法

## (3) 使用 cURL 调试

Linux:

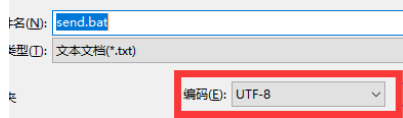
```
curl -X GET -data-urlencode "http://192.168.0.235:8060/send?password=1&text=您好&recipient=手机号码&encoding=U"
```

Windows:

将下列代码保存为 send.bat，并以 UTF-8 的编码保存

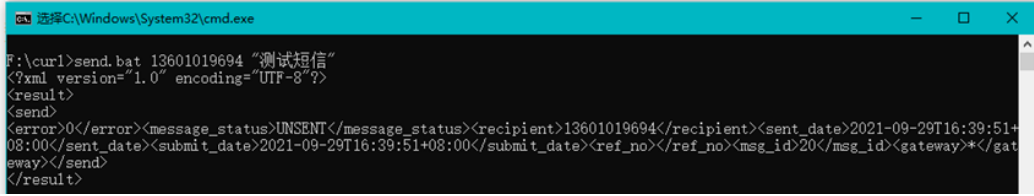
```

@echo off
set MsgTo=%1%
set MsgText=%2%
c:\curl\curl http://192.168.0.235:8060/send -X POST -data-urlencode -d password=1 -d text=%MsgText% -d recipient=%MsgTo% -d encoding=U
  
```



再将 curl.exe 存到 c:\curl 目录(工具下载地址 [金笛短信中间件 web 版 http 接口 CURL 调用说明](http://www.sendsms.cn))即可正常调用

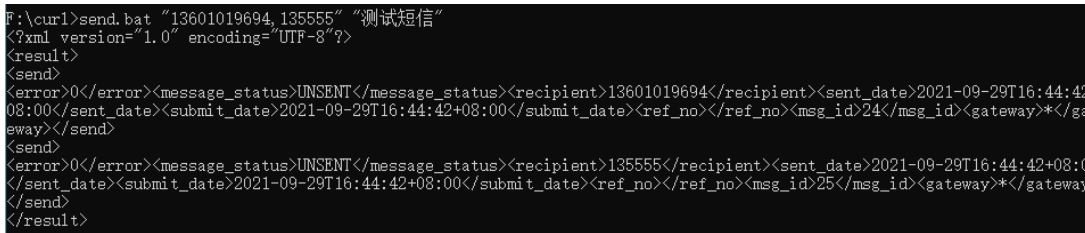
调用方法：在命令行输入 [send.bat 手机号 "短信内容" #引号请使用英文引号，不包含方框] 如图



可以看到短信内容已正确添加到带发送记录中



如果要给多个收件人发送,请使用英文逗号分隔手机号, 并且用引号将多个收件人 send.bat "手机号 1, 手机号 2" "短信内容"



Linux 也可以自行构造类似的 Shell 通过 cURL 发送短信, 编写过程中请注意编码, 也可以按照下一条直接调用 python 脚本完成操作

#### (4)使用 python 调试

将以下 python 保存成 send.py,使用 python 命令行调用即可

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
import requests,sys

def sendsms(recipient,text):
    params = {
        'text' : text,
        'recipient' : recipient,
        'password' : 1,
        'encoding' : 'U',
    }
    print(params)
    post_url='http://127.0.0.1:8080/send'
    resp = requests.get(url=post_url,params=params,timeout=5)
    resp.encoding = 'utf-8'
    print(resp.text)
```

```
if __name__ == '__main__':  
    phone = sys.argv[1]  
    message = sys.argv[2]  
    sendsms(phone,message)
```

\*请先安装 requests 库再运行(yum install python-pip, 再 pip install requests 即可)

返回值如下:

```
[root@hp-testbench pyTest]# ./send.py 10086 "nihao"  
{'text': 'nihao', 'password': 1, 'recipient': '10086', 'encoding': 'U'}  
<?xml version="1.0" encoding="UTF-8"?>  
<result>  
<send>  
<error>0</error><message_status>UNSENT</message_status><recipient>10086</recipient><sent_date>2021-10-08T18:13:29+08:00<  
/sent_date><submit_date>2021-10-08T18:13:29+08:00</submit_date><ref_no></ref_no><msg_id>6</msg_id><gateway>*</gateway></  
send>  
</result>
```

可以自行更改代码来适配各种需求

### (5) 使用 URL 生成工具调试

在 3.6.9-0715 以上版本中有自带的 URL 生成工具, 可以使用此工具生成的 URL 直接用浏览器调试短信发送功能

**URL生成工具**  
本工具可用于测试、演示HTTP接口

以下参数全部为必填项目, 请正确填写之后点击“生成URL”, 即可生成可用的HTTP接口链接。详细文档参考[开发接口详细说明](#)

发送目标(手机号)  
收信手机号 **此处填写手机号**

短信内容  
尊敬的用户您好, 您的验证码为\*\*\*\*\*, 验证码有效期为10分钟。

服务器IP  
192.168.0.235

端口号(默认为8060)  
8060

HTTPS  
 使用UTF-8(中文短信)  
 使用URLEncode(编码短信中特殊字符)

发送密码(默认为1)  
1

生成URL

填写手机号后点击【生成 URL】即可自动生成可用的 URL, 此 URL 可用浏览器访问, 也可改编成固定的发送地址

### (6) 第三方接口案例

以下为某平台设置界面截图

短信服务器设置

短信网关 SMPP

GET http://2000000000:8060/send

HTTP参数  
password=1&text=\$message&recipient=\$recipient&encoding=U

请求头

无身份验证  基本身份验证

将消息编码为Unicode

\*说明: 请确保您阅读并遵守SMS Gateway提供需指定的条款和条件。

GET 方法和 POST 方法皆可, 对请求头、身份验证没有要求。

其中 POST 方法需要用 x-www-form-urlencoded 格式, GET 方法需要用 URLEncode 对 TEXT 字段进行编码

## 使用 HTTP 接口读取短信

### 读取接口说明

金笛短信中间件 Web 版可以通过 URL 接口来查询收到的短信

```
http://127.0.0.1:8060/read?password=[password]&gateway=modem[id]&count=[number]
```

发送参数	含义	说明	示例
PASSWORD	必填	发送密码	默认值:1 admin
GATEWAY	选填	发送通道	详见发送表,可指定某个设备 modem6
COUNT	选填	数量	单词查询的条数 30

示例:

```
https://192.168.0.235:8060/read?password=1&gateway=modem1&count=20
```

此地址可以查询通道号为 modem1 的短信猫接收到的短信, 其中 modem1 不是指 com1, 是指中间件分配的通道号, 可以点击设备序号查看

如果 https 显示错误, 请[关闭 https 加密](#)

**HttpServer短信收发接口设置**

端口	<input style="width: 80%;" type="text" value="8060"/>
协议类型	<input type="checkbox"/> 启用HTTPS加密

调用返回结果

```
<?xml version='1.0' encoding="UTF-8"?>
<messages>
  <error>0</error>
  <message>
    <message_type>Inbound</message_type>
    <message_date>2021-03-22T14:45:11 08:00</message_date>
    <originator>861500000000</originator>
    <text>
      <![CDATA[
        %E6%9D%A5%E4%BF%A1%E5%B7%B2%E6%94%B6%E5%88%B0%EF%BC%8C%E8%B0%A2%E8%B0%A2
      ]]>
    </text>
    <encoding>U</encoding>
    <gateway>modem1</gateway>
  </message>
</messages>
```

使用 URLDecode 解码之后, 可以看到返回的短信内容“来信已收到, 谢谢”

Unicode编码
UTF-8编码
URL编码/解码
Unix时间戳
Ascii/Native编码互转
Hex编码/解码
Html编码/解码

来信已收到, 谢谢

utf-8

UrlEncode编码

UrlDecode解码

清空结果



## 第四章 回调接口

回调接口是中间件主动访问一个网址，告知该系统最新收到的短信内容及发送方号码，可用项目中被动接收中间件新收到的短信。

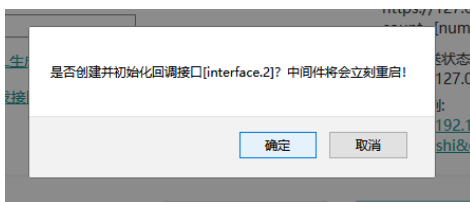
### 开启回调接口

在【中间件管理界面】-【配置与日志】-【系统设置】中找到【回调接口设置】

*\*此演示适用于高于3.6.9-0915版本的WEB中间件，若低于此版本，请安装新版本后操作*



初始界面如上，点击【启用回调】即初始化回调接口、并自动重启中间件



重新登录中间件后，会在设置中显示回调接口调用的 URL



可以看到回调 URL 中生成了一个默认的地址

`http://192.168.0.235:8090/test.jsp?gatewayId=%gatewayId%&text=%text%&originator=%originator%`

当收到短信，中间件会自动访问此 URL，此地址的页面可以将回调接口生成的数据格式化输出到日志文件中，可以在当天日志中查看调用情况。

### 回调接口配置

发送参数	含义	说明	示例
%gatewayId%	选填	来信通道号	默认值:1 modem1
%text%	选填	来信内容	最多1000字，内容已被URLencode编码，需要用URLDecode解码后查看 %E8%B0%A2%E8%B0%A2 解码得-“谢谢”
%originator%	选填	来信发件人	内容为数字，长度50 10086



回调 URL 中的%gatewayId%、%text%、%originator%会被中间件自动替换成每条短信的实际参数，请注意大小写以%gatewayId%、%text%、%originator%为准

例如，服务器为接收地址为 192.168.0.115/recv 时，回调 URL 可以写成

```
http://192.168.0.115/recv?msgfrom=%originator%&msg=%text%
```

那么中间件将在收到短信后访问这个 URL，并且将%originator%替换成电话号，%text%替换成经 URLEncode 编码的字符，并访问这个 URL，远端即可接收到回调指令

只需要保证 192.168.0.115/recv 的地址可以接收 GET 请求，并且执行记录或其他需要完成的任务，即可完成回调接口的对接

\*建议每次更改完后，点击【测试工具】发送回调请求验证是否能正常发送，否则请检测网络连接

## 通过配置文件更改

管理界面操作失败或者需要更多自选配置，可以选择此方法操作

找到配置文件中 #interface.2=http1,HttpPush，内容如图

```
db1.url.rac=jdbc:oracle:thin:@(DESCRIPTION=(LOAD_BALANCE=on)(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=192
#interface.2=http1,HttpPush
#http1.get_url=http://127.0.0.1:8090/test.jsp?gatewayId=%gatewayId%&text=%text%&originator=%originator%
#http1.method=GET
```

“#”代表这一行配置已被注释，中间件不执行此行配置，更改完配置后删除“#”保存即生效

将【http1.get\_url=】的值更改为项目使用的接收地址，并在新 url 中，将%gatewayId%、%text%、%originator%中需要的部分填入新的回调地址中，并将注释去掉后保存，重启中间件即可完成配置，界面将会自动变化

## 第五章 邮件接口

Web 中间件的邮件接口是将邮件通过短信方式发送给对方，触发规则是：邮箱中只要有邮件，程序就会通过短信的方式发送邮件。

### 注意事项：

1. 邮件接口正常开启的前提条件：需要开启目标邮箱的 pop3 和 smtp 服务，允许客户端读取短信，而且邮件内容不能带附件。
2. 短信收件人识别规则：  
[标题]或[内容的第一行]会被识别为手机号，[第二行及之后内容]会被识别为短信内容。也可以在中间件设置中修改。
3. 群发时，手机号可以用“；”分号进行分割，由中间件负责把这封邮件装成短信发给邮件中的手机号。
4. 邮件通过短信方式发出后，收件箱中的邮件会自动删除，以免重复发送。

### 配置中继邮箱(以 QQ 邮箱为例)

登录中继邮箱，找到邮箱设置



### 邮箱设置



在设置中，找到【帐户】选项

在下方开启【POP3】和【SMTP】服务

### POP3/IMAP/SMTP/Exchange/CardDAV/CalDAV服务

开启服务:	POP3/SMTP服务 (如何使用 Foxmail 等软件收发邮件?)	已关闭   开启
	IMAP/SMTP服务 (什么是 IMAP, 它又是如何设置?)	已关闭   开启
	Exchange服务 (什么是Exchange, 它又是如何设置?)	已关闭   开启
	CardDAV/CalDAV服务 (什么是CardDAV/CalDAV, 它又是如何设置?)	已关闭   开启
	(POP3/IMAP/SMTP/CardDAV/CalDAV服务均支持SSL连接。如何设置?)	

QQ 邮箱使用的的默认端口【使用 SSL 端口是指使用加密接口，如果无法成功配置，请关闭此选项】

邮箱类型：POP3

帐号：

收件服务器：  SSL 端口：

发件服务器：  SSL 端口：

如果服务器支持，就试用STARTTLS加密传输

记住此选项中配置，在配置文件中会用到

## 修改邮件接口配置

配置文件位置可以参考常见问题中的【配置文件的位置】

在配置文件中，找到#interface.2=email1,Email 这一行如图

```

46 #interface.2=email1,Email
47 #email1.mailbox_protocol=pop3
48 #email1.mailbox_host=pop.qq.com
49 #email1.mailbox_port=110
50 #email1.mailbox_user=1843933818@qq.com
51 #email1.mailbox_password=111111
52 #email1.mailbox_name=INBOX
53 #email1.to=1843933818@qq.com
54 #email1.from=1843933818@qq.com
55 #email1.smtp_host=smtp.qq.com
56 #email1.smtp_port=25
57 #email1.smtp_user=1843933818@qq.com
58 #email1.smtp_password=111111
59 #email1.message_subject=SMS from %originator%
60 #email1.message_body=emailbody.html.tpl
  
```

先将第一行最前面的#删除，开始详细配置其他信息

使用 QQ 邮箱时

下方黄色区域更改为【收件用邮箱】，绿色区域为您的密码，红色区域为来源邮箱(即【告警系统使用的邮箱账号】，可以与【收件用邮箱】相同)

若使用其他邮箱

需要找到这个邮件服务器的开放接口设置，按照说明更改浅蓝色区域，其他配置按照 QQ 邮箱部分的改

```
#email1.mailbox_protocol=pop3
#email1.mailbox_host=pop.qq.com
#email1.mailbox_port=110
#email1.mailbox_user=ShouJianRen@qq.com
#email1.mailbox_password=111111

#email1.mailbox_name=INBOX      #如果您的【收件箱】为其他名称，请对此项进行修改

#email1.to=ShouJianRen@qq.com
#email1.from=FaJianRen@qq.com
#email1.smtp_host=smtp.qq.com
#email1.smtp_port=25
#email1.smtp_user=ShouJianRen@qq.com
#email1.smtp_password=111111
#email1.message_subject=SMS from %originator%
#email1.message_body=emailbody.html.tpl
```

设置完成后，请将所有行首的#删掉，改完之后如下图

```
46 interface.2=email1,Email
47 email1.mailbox_protocol=pop3
48 email1.mailbox_host=pop.qq.com
49 email1.mailbox_port=110
50 email1.mailbox_user=ShouJianRen@qq.com
51 email1.mailbox_password=111111
52 email1.mailbox_name=INBOX
53 email1.to=ShouJianRen@qq.com
54 email1.from=FaJianRen@qq.com
55 email1.smtp_host=smtp.qq.com
56 email1.smtp_port=25
57 email1.smtp_user=ShouJianRen@qq.com
58 email1.smtp_password=111111
59 email1.message_subject=SMS from %originator%
60 email1.message_body=emailbody.html.tpl
```

至此，配置文件已成功更改，保存此文件后退出

如果下一步之后中间件重启不成功，请换回之前备份的配置文件

## 指定邮件转短信方式

登录中间件，在【配置与日志】中找到【系统设置】



在【邮件转短信设置】中选择收件人手机号获取方式，以及短信内容获取方式

### 邮件转短信设置

获取手机号码

从主题中提取

从内容中提取

指定号码:

获取发送内容

从邮件标题中提取

从邮件正文中提取

(指定号码时有效)

保存

重置

### 具体配置方法

#### 1. 收件人固定的情况

直接在中间件中配置“指定号码”，多个收件人用英文分号“;”隔开，如下图

从主题中提取  从内容中提取  指定号码:

保存此配置后重启中间件即可生效;

#### 2. 收件人不固定的情况

如果邮件内容可以定制，建议直接在主题中获取手机号

如果邮件形式固定，请按照实际情况配置手机号码和短信内容的获取方式

修改完之后，重启中间件

中间件重启方法见下方【常见问题】 - 【重启中间件服务的方法】

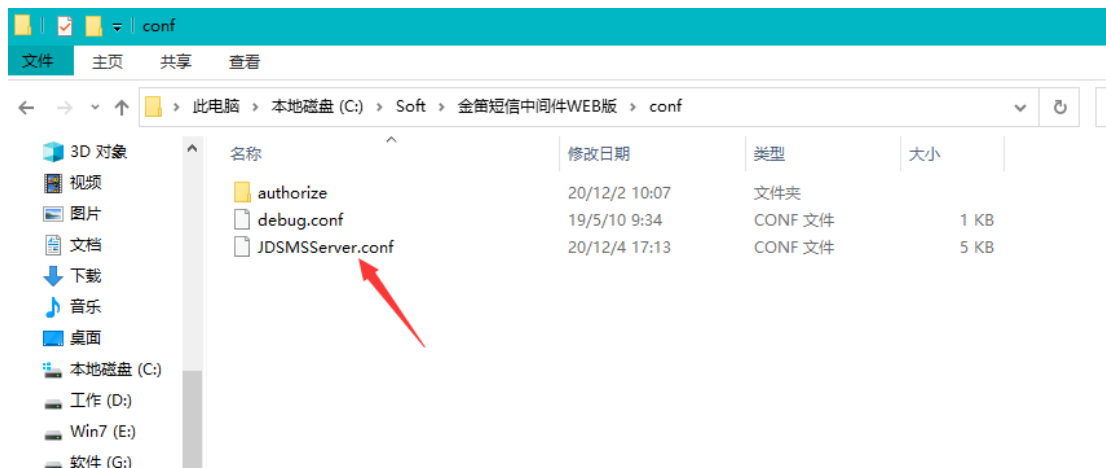
### 测试配置是否生效

请尝试使用【告警系统使用的邮箱账号】给【收件用邮箱】发送一条邮件，查看中间件是否能将该邮件收录到【待发送列表】

## 第六章 常见问题

### 配置文件的位置

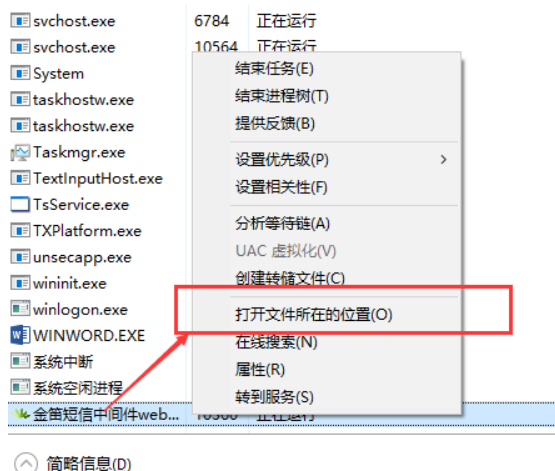
在中间件安装目录中，有 conf 文件夹，中间件的配置文件 JDSMSServer.conf 就在其中  
注意，在更改前请先将这个文件复制一份到其他文件夹，以作备份



如果未找到此文件，请先安装中间件，或者通过中间件在系统服务的进程定位到此目录



【转到详细信息】将会转到进程列表，可以在这里找到中间件的安装位置




点击【打开文件所在的位置】即可找到安装目录

### 配置文件无法更改

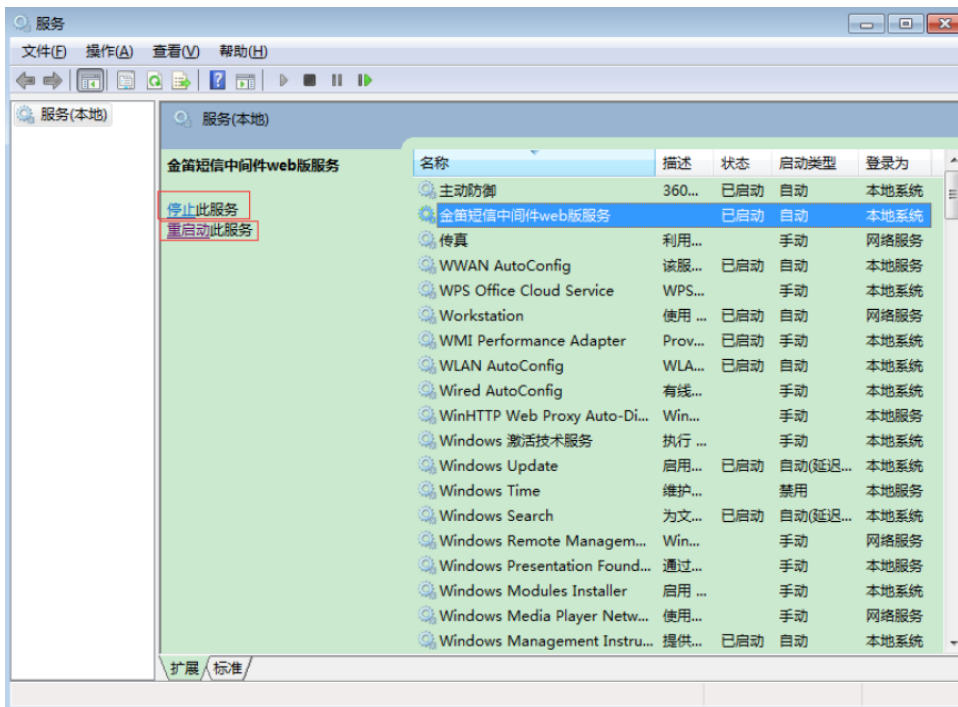
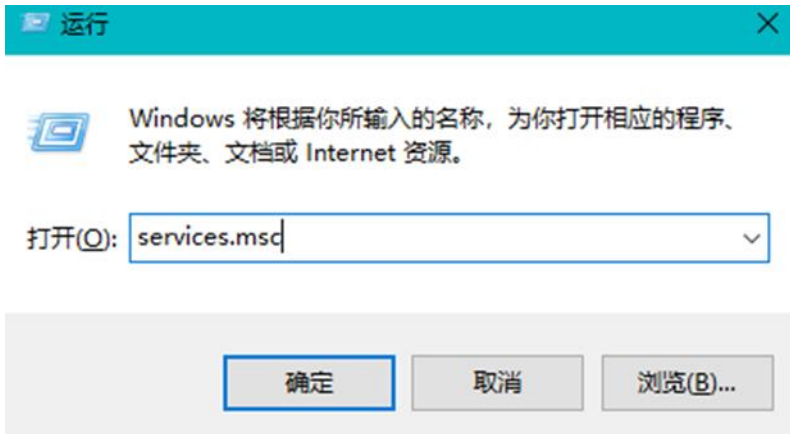
配置文件无法更改一般是中间件服务占用配置文件导致，将中间件的服务停止后更改即可。

## 重启中间件服务的方法

### 1、中间件自带的重启功能：

在左下角找到  按钮，进入关于页面，即可找到重启按钮，点击即可重启中间件，重启完成会自动跳转到登陆界面，重新登陆即可

### 2、Windows:找到系统服务配置(可以通过任务管理器-服务找到，或者在“运行”中输入 services.msc)



点左上方重新启动即可重启服务。

### 3、Linux: 找到中间件安装目录，在此中输入

```
./JDSmsService-web restart
```

演示：



```
[root@hp-testbench JDSmsService-web369]# ls
accounts  conf          getSystemKind.sh  jdk          lib          smppServiceCmd.sh  uninstall
bin       console      index.html        JDSmsPy30k  logs         smppService.sh    restart-jdsms.sh
cert     derby.log   infos            JDSmsService-usb  mysql        restart-jdsms.sh
CHANGELOG Doc          install          language     Readme.txt   temp
```

```
[root@hp-testbench JDSmsService-web369]# ./JDSmsService-web restart
```

即可重启中间件

## 中间件无法发出短信

检查项目：

1. 短信猫是否已授权(未授权的情况下，只能发送 88 条短信)
2. 短信猫状态是否已开启(在发送通道管理-短信设备中查看)
3. 数据库是否能正常连接、正常写入数据(在界面中发送一条短信，查看是否能正确添加到待发送列表)
4. 查看日志中是否显示错误，可以致电我公司 400-050-1096 转售后工程师

## 版本更新后能发短信，但无法收到短信

是新版本的数据表结构发生变化导致的问题，请在 smsserver\_tmpmsg 表中添加 ccid 字段后，重启设备即可修复

数据表	需添加字段	类型	长度	非空 NOT NULL
smsserver_tmpmsg	ccid	VARCHAR或CHAR	50	是

符合以下数据结构即可

Name	Type	Length	Decimals	Not null	
id	int	10	0	<input checked="" type="checkbox"/>	1
pdustring	varchar	500	0	<input checked="" type="checkbox"/>	
memindex	int	10	0	<input checked="" type="checkbox"/>	
memlocation	varchar	2	0	<input checked="" type="checkbox"/>	
ml	int	10	0	<input checked="" type="checkbox"/>	
ccid	varchar	50	0	<input checked="" type="checkbox"/>	