

---

# HTTP API 使用指南



---

# 版本更新日志

---

## 版本 1.1 2019-10-16

1. 加入新的 API: 获取设备状态
2. 增加示例

## 版本 1.0 2016-12-8

1. 在短信请求中加入短信回执参数;
2. 加入推送功能;
3. 取设备信息的 API 中加入了 “slot, callstate, signal, gprs” 参数;
4. 加入了设备端口的 API。

## 版本 0.5 2015-7-22

1. 加入 CDR 的 API;
2. 加入 STK API。

## 版本 0.4 2015-4-24

1. 修改发送短信请求的格式, 每个号码允许带一个 user\_id;
2. 加入查询当前等发送队列中的短信数目;
3. 查询短信发送结果中加入 user\_id 参数。

## 版本 0.3 2015-3-20

1. 短信发送状态加入 “已送达” 状态。

## 版本 0.2 2015-1-22

1. 短信发送加入 user\_id, 解决前一个版本的 BUG。

---

**版本 0.1 2014-12-25**

首个版本



---

# 目录

---

<b>版本更新日志</b> .....	<b>II</b>
版本 1.1 2019-10-16 .....	II
版本 1.0 2016-12-8 .....	II
版本 0.5 2015-7-22 .....	II
版本 0.4 2015-4-24 .....	II
版本 0.3 2015-3-20 .....	II
版本 0.2 2015-1-22 .....	II
版本 0.1 2014-12-25 .....	III
<b>1 API 介绍</b> .....	<b>1</b>
1.1 应用场景.....	1
1.2 主要应用组网图.....	1
1.3 主要特性.....	1
1.4 使用须知.....	1
<b>2 短信发送</b> .....	<b>3</b>
2.1 请求.....	3
2.2 请求参数.....	3
2.3 响应参数.....	4
2.4 短信发送实例.....	4
<b>3 查询短信发送结果</b> .....	<b>5</b>
3.1 请求.....	5
3.2 请求参数.....	5
3.3 响应参数.....	5
3.4 查询短信结果实例.....	6
<b>4 查询短信送达状态</b> .....	<b>7</b>
4.1 请求.....	7
4.2 请求参数.....	7
4.3 响应参数.....	7

---

4.4 查询短信送达状态实例 .....	8
<b>5 待发送短信数量.....</b>	<b>9</b>
5.1 请求.....	9
5.2 请求参数.....	9
5.3 响应参数.....	9
5.4 查询待发送短信数量实例 .....	9
<b>6 短信接收.....</b>	<b>10</b>
6.1 请求.....	10
6.2 请求参数.....	10
6.3 响应参数.....	10
6.4 短信接收实例.....	11
6.5 建议.....	11
<b>7 发送 USSD.....</b>	<b>12</b>
7.1 请求.....	12
7.2 请求参数.....	12
7.3 响应参数.....	12
7.4 发送 USSD 实例.....	13
<b>8 接收 USSD.....</b>	<b>14</b>
8.1 请求.....	14
8.2 请求参数.....	14
8.3 响应参数.....	14
8.4 接收 USSD 实例.....	14
<b>9 中止短信发送任务.....</b>	<b>16</b>
9.1 请求.....	16
9.2 请求参数.....	16
9.3 响应参数.....	16
9.4 中止短信任务实例.....	16
<b>10 端口信息获取.....</b>	<b>17</b>
10.1 请求.....	17
10.2 请求参数.....	17
10.3 响应参数.....	17
10.4 端口信息获取实例.....	18
<b>11 端口设置.....</b>	<b>20</b>

---

11.1 请求.....	20
11.2 请求参数.....	20
11.3 响应参数.....	20
11.4 端口设置实例.....	21
<b>12 获取 CDR.....</b>	<b>22</b>
12.1 请求.....	22
12.2 请求参数.....	22
12.3 响应参数.....	22
12.4 获取 CDR 实例.....	23
<b>13 获取 STK 视图.....</b>	<b>24</b>
13.1 请求.....	24
13.2 请求参数.....	24
13.3 响应参数.....	24
13.4 实例.....	25
<b>14 STK 操作.....</b>	<b>26</b>
14.1 请求.....	26
14.2 请求参数.....	26
14.3 STK 操作实例.....	26
<b>15 获取 STK 的 Frame ID .....</b>	<b>27</b>
15.1 请求.....	27
15.2 请求参数.....	27
15.3 响应参数.....	27
15.4 获取 STK 的 Frame ID 实例.....	27
<b>16 获取设备状态.....</b>	<b>28</b>
16.1 请求.....	28
16.2 请求参数.....	28
16.3 响应参数.....	28
16.4 实例.....	28
<b>17 推送.....</b>	<b>30</b>
<b>18 呼叫转移.....</b>	<b>32</b>
18.1 设置呼叫转移请求.....	32
18.2 设置请求参数.....	32
18.3 读取呼叫转移请求第一步.....	32

---

18.4 第一步读取请求参数.....	32
18.5 第二步读取呼叫转移请求.....	33
18.6 第二步读取请求参数.....	33
18.7 实例.....	33
<b>19 NAT 解决方案.....</b>	<b>34</b>
19.1 应用场景.....	34
19.2 配置部分.....	34
19.3 API 改动.....	34
<b>20 常见问题.....</b>	<b>36</b>
20.1 如何指定一个端口发送?.....	36
20.2 如何匹配短信发送请求和短信发送结果?.....	36
20.3 一条短信需要发多久?.....	36
20.4 怎样获得短信回执?.....	36
20.5 如何禁掉短信回执?.....	36
20.6 如何在程序中发送 HTTP 和 HTTPS 请求.....	37



---

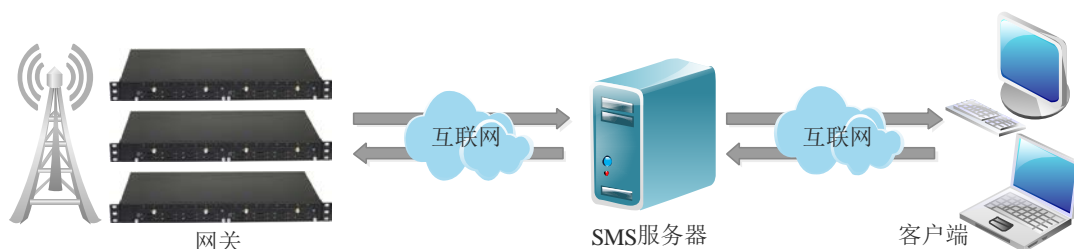
# 1 API 介绍

---

## 1.1 应用场景

一个 SMS server 通过 API 与成批网关通讯，用户可以连上这个 server 上进行短信收发。

## 1.2 主要应用组网图



## 1.3 主要特性

- 支持短信收发和 USSD 收发
- 支持批量短信发送
- 支持短信回执
- 支持获取基本的端口信息

## 1.4 使用须知

1) 本 API 基于 HTTP 和 JSON，请查询有关的编程手册，了解如何发送 HTTP 请求以及如何对 JSON 数据进行编解码。本文档使用 cURL 来进行示例，请您使用熟悉的编程语言转换成相应的代码。

2) 请在进一步测试前打开 API，在网关“无线配置→基本设置→API”页面，选择新版 API。网关必须使用 1102 及以上版本。推荐使用最新版本。



### 3) 关于短信的一些概念

- a) 网关支持两种编码, 即 **GSM 7bit** 和 **UCS2**。GSM 7bit 适用于英文及一些欧洲的字符。而 UCS2 适用于任何字符, 包含中文, 韩文, 日文, 甚至 **emoji** 表情;
- b) 使用 GSM 7bit 编码的情况下, 一条短信最多可发送 160 个字符, 而使用 UCS2 编码时, 一条短信只能发送 70 个字符;
- c) 网关支持**长短信**, 即需要发送一条很长的短信的时候, 网关会自动地把该长短信转成多条小短信发送, 而用户在手机侧看到的依然是一条很长的短信。

---

# 2 短信发送

---

## 2.1 请求

POST [https://gateway\\_ip/api/send\\_sms](https://gateway_ip/api/send_sms)

## 2.2 请求参数

参数名	类型	描述
必选参数		
text	字符串	短信的内容 支持 #param# 这种语法, 用于与后面的 text_param 匹配
param	对象数组	每个元素为一个对象,每个对象包括以下内容: number: 不超过 24 个字节的字符串 text_param: 字符串列表, 用于替换 text 中的 #param#的参数, 可选 user_id: 一个大于或等于 0 的整数。用于与发送结果匹配, 可选 (建议在短信发送请求中带上 user_id 这个参数, 方便其和短信发送结果匹配。短信每发往一个号码, 都带上一个 user_id, 对应的发送结果也会带上这个 user_id, 用于匹配。在您的程序中, user_id 可不断递增, 使每条发送请求中的 user_id 都是唯一的)
可选参数		
port	整数型数组	发送短信的端口 每个元素是一个数字, 范围为 0 到 31
encoding	字符串	可以为"unicode"或者"gsm-7bit", 默认为"unicode"
request_status_report	布尔型	是否需要短信回执, 可以为 true 或 false, 默认为 true

---

## 2.3 响应参数

参数名	类型	描述
error_code	整数	返回请求的处理结果,可能的结果为: 202: 表示请求已经接受, 稍后处理 400: 请求的格式有错误 413: 号码数量超过 128 或者 text 超过出 1500 字节 500: 其他错误 550: 当前无任何共用端口
sn	字符串	设备当前的 SN
sms_in_queue	整数	返回当前正在等待处理的短信的条数
task_id	整数	该任务 ID 后续可用来中止对应的发送任务

## 2.4 短信发送实例

简单请求:

```
curl -k --anyauth -u admin:admin -d  
'{"text": "ye", "param": [{"number": "10086"}]}' -H "Content-Type:  
application/json" https://gateway_ip/api/send_sms
```

使用#param#的请求:

```
curl -k --anyauth -u admin:admin -d  
'{"text": "#param#", "port": [2, 3], "param": [{"number": "10086", "te  
xt_param": ["bj"], "user_id": 1}, {"number": "10086",  
"text_param": ["ye"], "user_id": 2}]}' -H "Content-Type:  
application/json" https://gateway_ip/api/send_sms
```

**注意:**

1. 本文档中的用户名和密码为设备默认用户名和密码, 请按实际的用户名和密码替换
2. 本文档中的 gateway\_ip 请替换为网关实际的 IP

响应:

```
{"error_code": 202,  
"sn": "xxxx-xxxx-xxxx-xxxx", "sms_in_queue": 2, "task_id": 2}
```

---

# 3 查询短信发送结果

---

## 3.1 请求

POST [https://gateway\\_ip/api/query\\_sms\\_result](https://gateway_ip/api/query_sms_result)

## 3.2 请求参数

参数名	类型	描述
可选参数		
number	字符串数组	每个元素为一个长度不超过 24 个字节的字符串, 号码个数不超过 32 个
port	整数型数组	发送短信的端口 每个元素是一个数字, 范围为 0 到 31
time_after	字符串	是一个"YYYY-MM-DD HH:MM:SS"格式的字符串, 用于查询在这个时间以后发送的短信发送结果
time_before	字符串	是一个"YYYY-MM-DD HH:MM:SS"格式的字符串, 用于查询在这个时间以前发送的短信发送结果
user_id	整数型数组	每个 user_id 用于与发送请求中的 user_id 匹配 推荐使用此参数.(此 user_id 为发送短信时设置的 ID 索引值, 查询短信时, 携带 user_id 能够准确查询到对应的短信)

## 3.3 响应参数

参数名	类型	描述
error_code	整数	返回对于请求的处理结果,可能的结果包括: 200: 表示请求已经接受 400: 请求的格式有错误 413: 号码数量超过 32 个 500: 其他错误

sn	字符串	设备的 SN
result	对象数组	发送结果的数组，每个元素包含以下信息： port: 端口号 number: 目的号码 user_id: 这条短信的 user_id，只有当发送短信的请求带了 user_id，发送结果中才会包含该 user_id time: 发送时间 status: 发送状态，包括 FAILED、SENDING、SENT_OK、DELIVERED count: 短信被分成了几段 succ_count: 成功发送了几段 ref_id: 这条短信的第一段的参考 ID，用于与短信送达状态匹配。 <b>注意：该 ref_id 只是一个参考 ID，它的取值范围为 0 到 255，因此最好与号码，端口甚至发送时间捆绑起来与送达状态进行匹配</b> imsi: 卡的 IMSI

### 3.4 查询短信结果实例

请求:

```
curl -k --anyauth -u admin:admin -d '{"user_id":[1,2]}' -H
"Content-Type: application/json"
https://gateway_ip/api/query_sms_result
```

响应:

```
{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx","result":[{"port":
:0,"user_id":1,"number":"12351","time":"2014-12-21
12:06:01","status":"SENT_OK","count":3,"succ_count":3,
"ref_id":12,"imsi":"460004642148063"}]}
```

建议使用 user\_id 来查询，也可以使用推送来获取。[推送](#)的格式如下:

```
{"sn":"xxxx-xxxx-xxxx-xxxx","sms_result":[{"port":1,"number":
"10086","time":"2016-07-12
01:46:02","status":"DELIVERED","count":1,"succ_count":1,"ref_i
d":215,"imsi":"460004642148063"}]}
```

---

# 4 查询短信送达状态

---

## 4.1 请求

POST [https://gateway\\_ip/api/query\\_sms\\_deliver\\_status](https://gateway_ip/api/query_sms_deliver_status)

## 4.2 请求参数

参数名	类型	描述
可选参数		
number	字符串数组	每个元素为一个长度不超过 24 个字节的字符串, 号码个数不超过 32 个
port	整型数组	发送短信的端口 每个元素是一个数字, 范围为 0 到 31
time_after	字符串	是一个"YYYY-MM-DD HH:MM:SS"格式的字符串, 用于查询这个时间以后发送的短信发送结果
time_before	字符串	是一个"YYYY-MM-DD HH:MM:SS"格式的字符串, 用于查询这个时间以前发送的短信发送结果

## 4.3 响应参数

参数名	类型	描述
error_code	整数	返回请求的处理结果, 可能的结果包括: 200: 表示请求已经接受 400: 请求的格式有错误 413: 号码数量超过 32 500: 其他错误
sn	字符串	设备当前的 SN
result	数组	送达状态的数组, 每个元素包含以下信息: port: 端口号

		<p><b>number:</b> 发送号码</p> <p><b>time:</b> 发送时间</p> <p><b>ref_id:</b> 用于与短信发送结果匹配</p> <p><b>status_code:</b> 送达状态码, 这是一个 0 至 255 的值, 如  0: 表示被对端收到,  32 ~ 63: temporary error  64~255: permanent error  具体的值的含义请查询“SMS Status Report”</p> <p><b>imsi:</b> 卡的 IMSI</p>
--	--	---

## 4.4 查询短信送达状态实例

请求:

```
curl -k --anyauth -u admin:admin -d '{"number":["12341234"],
"port":[1,2,3], "time_after":"2014-12-12 19:29:19",
"time_before":"2014-12-12 19:29:19"}' -H "Content-Type:
application/json"
https://gateway_ip/api/query_sms_deliver_status
```

响应:

```
{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx","result":[{"port":
:0,"number":"12341234","time":"2014-12-21 12:06:01","ref_id":12,
"status_code":0,"imsi":"460004642148063"}]}
```

送达状态也支持[推送](#), 格式如下:

```
{"sn":"xxxx-xxxx-xxxx-xxxx","sms_deliver_status":[{"port":1,"n
umber":"10086","time":"2016-07-12
15:46:53","ref_id":215,"status_code":0,"imsi":"460004642148063
"}]}
```



---

# 5 待发送短信数量

---

## 5.1 请求

GET `https://gateway_ip/api/query_sms_in_queue`

## 5.2 请求参数

无

## 5.3 响应参数

参数名	类型	描述
error_code	整数	可能的返回值包括： 200: 合法请求，按请求返回结果 500: 其他错误
sn	字符串	设备当前的 SN
in_queue	整数	当前设备上待发送短信的数目

## 5.4 查询待发送短信数量实例

请求:

`https://gateway_ip/api/query_sms_in_queue`

**注:** 这个请求可以直接在浏览器中输入

响应:

```
{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx","in_queue":0}
```

---

# 6 短信接收

---

## 6.1 请求

GET https://gateway\_ip/api/query\_incoming\_sms

## 6.2 请求参数

参数名	类型	描述
可选参数		
incoming_sms_id	整数	大于 0 的整数，网关会返回 ID 大于这个值的短信，默认值为 0
flag	字符串	可选值为： “unread”：未读短信， <b>读取后变为已读</b> “read”：已读短信 “all”：未读和已读 默认值为“unread”
port	数组	接收短信的端口 每个元素是一个数字，范围为 0 到 31

## 6.3 响应参数

参数名	类型	描述
error_code	整数	可能的返回值包括： 200：法请求，按请求返回结果 500：其他错误
sn	字符串	设备当前的 SN
sms	数组	每个元素为一条短信，包含以下内容： incoming_sms_id：在网关数据库中的短信 ID，整数

		port: 短信所属的端口, 端口号码为整数 number: 短信发送方的号码, 字符串 smsc: 短信中心号码, 字符串 timestamp: 接收时间, 字符串 text: 短信内容, 字符串
read	整数	当前设备上已读短信的数目
unread	整数	当前设备上未读短信的数目

## 6.4 短信接收实例

请求:

```
https://gateway_ip/api/query_incoming_sms?flag=all
```

响应:

```
{ "error_code": 200, "sn": "xxxx-xxxx-xxxx-xxxx", "sms": [ { "incoming_sms_id": 1, "port": 2, "number": "123456789", "smc": "+8613800123456", "timestamp": "2014-12-09 17:11:18", "text": "This is a test" }, ], "read": 1, "unread": 0 }
```

短信支持[推送](#), 格式如下

```
{ "sn": "xxxx-xxxx-xxxx-xxxx", "sms": [ { "incoming_sms_id": 1, "port": 1, "number": "6717", "smc": "+8613800757511", "timestamp": "2016-07-12 15:46:18", "text": "test" } ] }
```

## 6.5 建议

1. 使用推送来获取短信
2. 如果采用查询方式来获取短信, 需要带入 incoming\_sms\_id, 并且每次在随后的请求中不断增大。例如, 第一次可以不带, 第二次开始, 就带上第一次返回的结果里最大的 id。

---

# 7 发送 USSD

---

## 7.1 请求

POST [https://gateway\\_ip/api/send\\_ussd](https://gateway_ip/api/send_ussd)

## 7.2 请求参数

参数名	类型	描述
必选参数		
text	字符串	发送的 USSD 内容 必须不超过 60 字节 当 command 为 “send” 时必须包含 USSD 内容 当 command 为 “cancel” 时，该参数可以为空
port	数组	发送 USSD 的端口 每个端口号码是一个数字，范围为 0 到 31
可选参数		
command	字符串	可选值为 “send” 或 “cancel”，默认为 “send”

## 7.3 响应参数

参数名	类型	描述
error_code	整数	可能的返回值包括： 202：请求已处理 400：非法请求,例如命令为 “send”，但是 “text” 不存在 500：其他错误
sn	字符串	设备当前的 SN
result	数组	发送结果的数组，包含以下内容： port：发送 USSD 的端口，端口号码为整数 status：状态，可能出现以下值 200：表示成功发送

---

		486: 表示当前端口忙（例如正在发送短信） 503: 表示当前端口未注册上
--	--	---

## 7.4 发送 USSD 实例

请求:

```
curl -k --anyauth -u admin:admin -d  
'{"port":[1,2,3],"command":"send","text":"*125#}' -H  
"Content-Type: application/json"  
https://gateway_ip/api/send_ussd
```

响应:

```
{"error_code":202,"sn":"xxxx-xxxx-xxxx-xxxx","result":[{"port":  
:0, "status":503}, {"port":1, "status":503}, {"port":2,  
"status":200}]}
```

---

# 8 接收 USSD

---

## 8.1 请求

GET `https://gateway_ip/api/query_ussd_reply`

## 8.2 请求参数

参数名	类型	描述
必选参数		
port	数组	需要查询的端口，例如查询的端口是 3，那个返回的是端口 3 接收到的 USSD

## 8.3 响应参数

参数名	类型	描述
error_code	整数	可能的返回值包括： 202：请求已处理 400：非法请求 500：设备出现异常
sn	字符串	设备当前的 SN
reply	数组	每个元素是一条 USSD 内容，包含以下内容： Port: USSD 所属的端口，端口号码为整数 Text: 回复的内容

## 8.4 接收 USSD 实例

请求：

```
https://gateway_ip/api/query_ussd_reply?port=1,2,3
```

响应：

```
{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx","reply":[{"port":1,"text":""},{ "port":2,"text": "Test..."}, {"port":3,"text": ""} ]}
```

USSD 支持[推送](#)，格式如下：

---

```
{"sn":"da00-0030-1901-2817","ussd":[{"port":1,"text":"Thank  
you!"}]}
```

---

# 9 中止短信发送任务

---

## 9.1 请求

GET `https://gateway_ip/api/stop_sms`

## 9.2 请求参数

参数名	类型	描述
必选参数		
task_id	整数	需要中止的任务的 ID，对应 send_sms 响应中的 task_id

## 9.3 响应参数

参数名	类型	描述
error_code	整数	可能的返回值包括： 200: 任务已删除 486: 未找到该任务 500: 其他错误
sn	字符串	设备当前的 SN

## 9.4 中止短信任务实例

请求:

```
https://gateway_ip/api/stop_sms?task_id=1
```

响应:

```
{"error_code": 200, "sn": "xxxx-xxxx-xxxx-xxxx"}
```



---

# 10 端口信息获取

---

## 10.1 请求

GET https://gateway\_ip/api/get\_port\_info

## 10.2 请求参数

参数名	类型	描述
必选参数		
info_type	字符串数组	可以是以下字符串的组合： type、imei、imsi、iccid、number、reg、slot、callstate、signal、gprs、remain_credit、remain_monthly_credit、remain_daily_credit、remain_daily_calltime、remain_hourly_calltime、remain_daily_connect
可选参数		
port	数组	需要查询的端口，例如需要查询的是端口 1，那么返回的是端口 1 的信息 默认为所有端口

## 10.3 响应参数

参数名	类型	描述
error_code	整数	可能的返回值包括： 200: 正常查询到结果 400: 非法请求 500: 其他错误
sn	字符串	设备当前的 SN
info	数组	每个元素是各个端口的信息，包含以下内容： port: 查询的端口号 type: 可以是 GSM、CDMA、WCDMA 或 LTE，未识别类型时为 UNKNOWN imei: 该端口的 IMEI

		<p>imsi: 该端口上的 SIM 卡的 IMSI</p> <p>iccid: 该端口上的 SIM 卡的 ICCID</p> <p>number: 该端口上的 SIM 卡的手机号码</p> <p>reg: 该端口的注册状态, 可能值为:</p> <p>POWER_OFF</p> <p>NO_SIM</p> <p>PIN_REQUIRE</p> <p>PUK_REQUIRE</p> <p>UNREGISTER</p> <p>SEARCHING_NETWORK</p> <p>REGISTER_OK</p> <p>UNKNOWN</p> <p>slot: 当前选中的卡槽 (当设备为一模多卡设备, 需要选中卡槽, 范围为 0 到 3, 当不可用时返回 255)</p> <p>callstate: 当前呼叫状态, 可能的值包括: Idle、Processing、Ringing、Active、Alerting、Call Waiting、Call Holding 和 Unknow</p> <p>signal: 信号强度, 范围为 0 到 31</p> <p>gprs: GPRS 状态, attached 或者 detached</p> <p>remain_credit: 该端口总的呼叫余额</p> <p>remain_monthly_credit: 该端口月呼叫余额</p> <p>remain_daily_credit: 该端口日呼叫余额</p> <p>remain_daily_calltime: 该端口的日剩余呼叫次数</p> <p>remain_hourly_calltime: 该端口的小时剩余呼叫次数</p> <p>remain_daily_connect: 该端口的日剩余呼通次数</p>
--	--	--

## 10.4 端口信息获取实例

请求:

```
https://gateway_ip/api/get_port_info?port=1,2,3&info_type=imei,imsi,iccid,smcsc,type,number,reg,slot,callstate,signal,gprs,remain_credit,remain_monthly_credit,remain_daily_credit:,remain_daily_calltime,remain_hourly_calltime,remain_daily_connect
```

响应:

```
{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx","info":[{"port":1,"type":"WCDMA","imei":"863070017005173","imsi":"","iccid":"","smcsc":"","number":"","reg":"NO_SIM","callstate":"Idle","signal":0,"gprs":"detached","remain_credit":"1000.00","remain_monthl
```

---

```
y_credit":"500.00","remain_daily_credit":"300.00","remain_dail
y_call_time":"100","remain_hourly_call_time":"10","remain_dail
y_connected":"100"}, {"port":2,
"type":"GSM","imei":"358967042917201","imsi":"460016529802215"
,"iccid":"89860114840400428150","smsc":"+8613010868500","numbe
r":"","reg":"REGISTER_OK","callstate":"Idle","signal":0,"gprs"
:"detached","remain_credit":"1000.00","remain_monthly_credit":
"500.00","remain_daily_credit":"300.00","remain_daily_call_tim
e":"100","remain_hourly_call_time":"10","remain_daily_connecte
d":"100"}, {"port":3,
"type":"GSM","imei":"358967042917201","imsi":"","iccid":"","sm
sc":"","number":"","reg":"NO_SIM","callstate":"Idle","signal":
0,"gprs":"detached","remain_credit":"1000.00","remain_monthly_
credit":"500.00","remain_daily_credit":"300.00","remain_daily_
call_time":"100","remain_hourly_call_time":"10","remain_daily_
connected":"100"}]
```

---

# 11 端口设置

---

## 11.1 请求

GET https://gateway\_ip/api/set\_port\_info

## 11.2 请求参数

参数名	类型	描述
必选参数		
action	字符串	可以是以下值 slot: 选择卡槽(只适用于一多卡设备) reset: 重启模块 power: 用于启动或关闭模块 imei: 用于修改 IMEI, param 必须是 IMEI number: 用于修改手机号, param 必须是号码 lock: 锁 SIM 卡, param 必须是卡的 PIN 码 unlock:解锁 SIM 卡,param 必须是卡的 PIN 码 block:禁止端口呼叫 unblock:解除禁止 CallForward:设置呼叫转移, 详情见 <a href="#">呼叫转移</a> CheckCallForward: 发送查询呼叫转移的命令
param	字符串	Slot: 卡槽号 (用于一多卡设备, 范围为 0-3) Action 为 Power 时: 可以为“on”或“off”
port	整数	端口号, 范围为 0 至 31

## 11.3 响应参数

参数名	类型	描述
error_code	整数	可能的返回值包括:

---

		200: 常查询到结果 400: 非法请求 500: 其他错误
sn	字符串	设备当前的 SN

## 11.4 端口设置实例

请求:

```
https://gateway_ip/api/set_port_info?port=1&action=power&param=off
```

响应:

```
{"error_code": 200, "sn": "xxxx-xxxx-xxxx-xxxx"}
```

---

# 12 获取 CDR

---

## 12.1 请求

POST `https://gateway_ip/api/get_cdr`

## 12.2 请求参数

参数名	类型	描述
必选参数：无		
可选参数		
port	数组	需要查询的端口，默认为所有端口
time_after	字符串	是一个“YYYY-MM-DD HH:MM:SS”格式的字符串，查询这个时间以后发送的 CDR
time_before	字符串	是一个“YYYY-MM-DD HH:MM:SS”格式的字符串，查询这个时间以前发送的 CDR

## 12.3 响应参数

参数名	类型	描述
error_code	整数	可能的返回值包括： 200：正常查询到结果 400：非法请求 500：其他错误
sn	字符串	设备当前的 SN
cdr	数组	每个元素是一条完整的 CDR，包含以下信息： port：端口号 start_date：呼叫开始时间 answer_date：应答时间 duration：通话时长 source_number：源号码

		<p>destination_number: 被叫号码</p> <p>direction: 呼叫方向, 包含 gsm→ip、 ip→gsm 或 callback</p> <p>ip: ip→gsm 呼叫的源 IP 地址</p> <p>codec: 呼叫使用的编解码, 包含 G.711U、G.723.1、G.711A 和 G.729AB</p> <p>Hangup: 挂断方, 包含 called, calling, gateway</p> <p>gsm_code: GSM 侧挂断时的原因码。</p> <p>Bcch: 呼叫时使用的 BCCH</p>
--	--	--

## 12.4 获取 CDR 实例

请求:

```
curl -k --anyauth -u admin:admin -d '{"port":[2,3]}' -H
"Content-Type: application/json" http://gateway_ip/api/get_cdr
```

响应:

```
{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx","cdr":[{"port":2,
"start_date":"2015-07-21 16:35:20","answer_date":"2015-07-21
16:35:31","duration":3,"source_number":"1010","destination_num
ber":"6717","direction":"ip->gsm","ip":"172.16.100.136","codec
":"G.711U","hangup":"called","gsm_code":16,"bcch":""}]}
```

---

# 13 获取 STK 视图

---

## 13.1 请求

GET https://gateway\_ip/GetSTKView

## 13.2 请求参数

参数	类型	描述
必选参数		
port	数字	

## 13.3 响应参数

参数	类型	描述
title	字符串	
text	字符串	
input_type	整数	0: 只显示 1: 显示桌面 2: 选择 item 3: 要求 yes 或 no 的输入 4: 需要单个数字的输入 5: 需要单个字符的输入 8: 需要纯数字的输入 9: 需要纯数字的输入并且不能回显 10: 需要字符输入 11: 需要字符输入并且不能回显 12: 锁定状态, 不能输入
frame_id	整数	当前 STK 视图的 id
item	对象	当 input_type 为 1 或 2 时, 该参数才存在
Item.item_id	数字	item 的 id
Item.item_字符串	字符串	item 的名称



---

## 13.4 实例

请求:

```
https://gateway_ip/GetSTKView?port=0
```

响应:

```
{"title": "神州行天地", "item": [{"item_id": 1, "item_字符串": "轻松问候"}, {"item_id": 2, "item_字符串": "短信群发"}, {"item_id": 3, "item_字符串": "最新推荐"}, {"item_id": 4, "item_字符串": "业务精选"}, {"item_id": 5, "item_字符串": "无线音乐俱乐部"}, {"item_id": 6, "item_字符串": "139 互联网"}, {"item_id": 7, "item_字符串": "财信通"}, {"item_id": 8, "item_字符串": "移动优惠专区"}, {"item_id": 9, "item_字符串": "移动电子商务"}, {"item_id": 10, "item_字符串": "我的 SIM 营业厅"}, {"item_id": 15, "item_字符串": "SIM 卡信息"}], "input_type": "2", "frame_id": 750}
```

---

# 14 STK 操作

---

## 14.1 请求

POST `https://gateway_ip/STKGo`

## 14.2 请求参数

参数	类型	描述
必选参数		
port	整数	
可选参数		
item	整数	
param	字符串	
action	字符串	“ok”、“cancel”或“home”

## 14.3 STK 操作实例

请求:

```
curl -k --anyauth -u admin:admin -d '{"port":7,"item":1}' -H "Content-Type: application/json" -H "Content-Type: application/json" http://gateway_ip/STKGo
```

```
curl -k --anyauth -u admin:admin -d '{"port":7,"action":"candle"}' -H "Content-Type: application/json" -H "Content-Type: application/json" http://gateway_ip/STKGo
```

---

# 15 获取 STK 的 Frame ID

---

## 15.1 请求

GET `https://gateway_ip/GetSTKCurrFrameIndex`

## 15.2 请求参数

参数	类型	描述
必选参数		
port	整数	

## 15.3 响应参数

参数	类型	描述
frame_id	整数	

## 15.4 获取 STK 的 Frame ID 实例

请求:

```
https://gateway_ip/GetSTKCurrFrameIndex?port=0
```

响应:

```
{"frame_id":32}
```

---

# 16 获取设备状态

---

## 16.1 请求

POST `https://gateway_ip/api/get_status`

## 16.2 请求参数

参数	类型	描述
必选参数		
performance		

## 16.3 响应参数

参数名	类型	描述
cpu_used	字符串	CPU 占用率
flash_total	字符串	flash 总的大小
flash_used	字符串	已经使用的 flash 大小
memory_total	字符串	内存总大小
memory_cached	字符串	缓存大小
memory_buffers	字符串	缓冲区大小
memory_free	字符串	空闲的内存大小
memory_used	字符串	已经使用的内存大小

## 16.4 实例

请求

```
curl -k --anyauth -u admin:admin -d '{"performance"}' -H "Content-Type: application/json" https://gateway_ip/api/get_status
```

---

响应:

```
{"performance":{"cpu_used":"39","flash_total":"27648","flash_used":"17428","memory_total":"109448","memory_cached":"34192","memory_buffers":"0","memory_free":"58928","memory_used":"50520"}}
```

# 17 推送

在最新的版本中，网关支持推送一些事件到您的程序，这些事件包括短信、短信发送结果、短信送达状态、USSD 等。

The screenshot shows the 'Basic Configuration' (基本配置) window with the following settings:

- SIM模式: 本地
- API:  禁用  启用老版API  启用新版API  AT  SMPP
- 启用推送:
- URL: [Empty text box]
- 推送短信:
- 推送USSD:
- 推送短信发送结果:
- 推送短信送达状态:
- 推送未接来电:
- 推送SIM卡注册状态:
- 推送CDR:
- 推送开机设备信息:
- 推送异常事件:
- 高级设置:
- USSD缺省编码: UCS2
- GSM语音编码: 自动
- VoLTE: 启用

推送的短信的内容与[查询短信](#)的内容很接近:

```
{"sn": "xxxx-xxxx-xxxx-xxxx", "sms": [{"incoming_sms_id": 1, "port": 1, "number": "6717", "smsc": "+8613800757511", "timestamp": "2016-07-12 15:46:18", "text": "test"}]}
```

推送短信的发送结果与[查询短信发送结果](#)也很接近:

```
{"sn": "xxxx-xxxx-xxxx-xxxx", "sms_result": [{"port": 1, "number": "10086", "time": "2016-07-12 01:46:02", "status": "DELIVERED", "count": 1, "succ_count": 1, "ref_id": 215, "imsi": "460004642148063"}]}
```

短信送达状态:

```
{"sn": "xxxx-xxxx-xxxx-xxxx", "sms_deliver_status": [{"port": 1, "number": "10086", "time": "2016-07-12 15:46:53", "ref_id": 215, "status_code": 0, "imsi": "460004642148063"}]}
```

USSD:

---

```
{"sn":"xxxx-xxxx-xxxx-xxxx","ussd":[{"port":1,"text":"Thank you!"}]}
```

SIM 注册状态:

```
{"sn":"xxxx-xxxx-xxxx-xxxx","register":[{"port":8,"iccid":"89860040191844710023","imsi":"460004642148063","number":"13714637674","status":"up","sequence":2,"slot":2}]}
```

```
{"sn":"xxxx-xxxx-xxxx-xxxx","register":[{"port":8,"iccid":"<NULL>","imsi":"<NULL>","number":"13714637674","status":"down","sequence":1,"slot":2}]}
```

注: 一模多卡推送注册状态时会带上槽位号

CDR:

```
{"sn":"xxxx-xxxx-xxxx-xxxx","cdr":[{"port":2,"start_date":"2015-07-21 16:35:20","answer_date":"2015-07-21 16:35:31","duration":3,"source_number":"1010","destination_number":"6717","direction":"ip->gsm","ip":"172.16.100.136","codec":"G.711U","hangup":"called","gsm_code":16,"bcch":""}]}
```

注: CDR 只推送接通的呼叫。如果需要获取未接通的呼叫, 可以用 `get_cdr` 来获取

开机设备信息:

```
{"sn":"xxxx-xxxx-xxxx-xxxx","device":{"port_number":32,"IP":"172.18.55.142","MAC":"F8-A0-3D-48-E5-19","status":"power_off"}}
```

```
{"sn":"xxxx-xxxx-xxxx-xxxx","device":{"port_number":32,"IP":"172.18.55.142","MAC":"F8-A0-3D-48-E5-19","status":"power_on"}}
```

注: `power_off` 表示关机, `power_on` 表示开机

异常信息:

```
{"sn":"xxxx-xxxx-xxxx-xxxx","exception_info":{"port":0,"type":"call_fail","action":"reset"}}
```

注: 需要开启异常事件处理, 才会有信息推送

---

# 18 呼叫转移

---

## 18.1 设置呼叫转移请求

GET https://gateway\_ip/api/set\_port\_info

## 18.2 设置请求参数

参数	类型	描述
必选参数		
port	整数	设备端口号
action	字符串	为 CallForward
param	字符串	为 Uncondational, NoReply, Busy, Not_Reachable, CancelAll 的其中一个
number	字符串	电话号码

## 18.3 读取呼叫转移请求第一步

GET https://gateway\_ip/api/set\_port\_info

## 18.4 第一步读取请求参数

参数	描述
必选参数	
port	设备端口号
action	为 CheckCallForward



---

## 18.5 第二步读取呼叫转移请求

GET https://gateway\_ip/api/get\_port\_info

## 18.6 第二步读取请求参数

参数	描述
必选参数	
port	设备端口号
info_type	为 CallForward

## 18.7 实例

### 1. 设置呼叫转移

```
https://gateway_ip/api/set_port_info?port=8&action=CallForward&param=Unconditional&number=15013828917
```

```
{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx"}
```

### 2. 发送查询命令

```
https://gateway_ip/api/set_port_info?port=8&action=CheckCallForward
```

```
{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx"}
```

### 3. 获取查询结果

```
https://gateway_ip/api/get_port_info?port=8&info_type=CallForward
```

```
{"error_code":200,"sn":"xxxx-xxxx-xxxx-xxxx","info":[{"port":8,"CallForwarding":{"Unconditional":"15013828917"}}]}
```

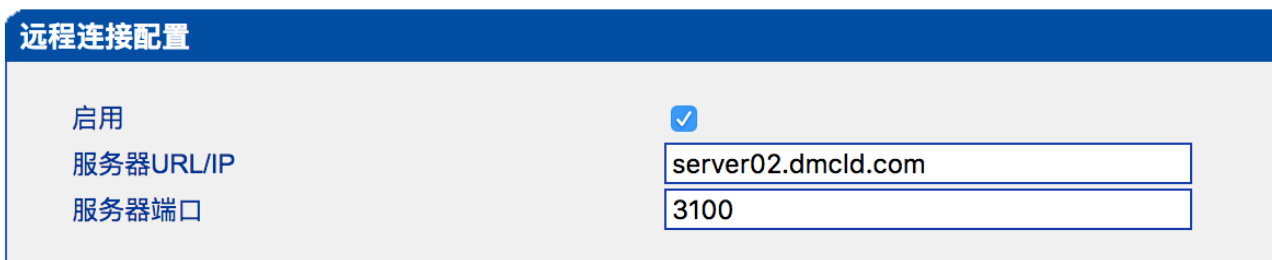
# 19 NAT 解决方案

## 19.1 应用场景

可通过对接 DRP 服务器实现同私网的网关通信，解决由于 NAT 原因导致 API 客户端无法直接发送请求到网关的问题；

## 19.2 配置部分

(1) 网关需启用“工具->远程连接配置”



远程连接配置	
启用	<input checked="" type="checkbox"/>
服务器URL/IP	server02.dmcloud.com
服务器端口	3100

(2) DRP 服务器上号添加对应设备



增加设备	
设备序号	<input type="text"/>
名称	<input type="text"/>
备注	<input type="text"/>

## 19.3 API 改动

根据 DRP 服务器配置的 IP 和端口（配置文件中），API 客户端将请求发送至该地址；

(1) 对接 DRP 服务器，发送账户密码进行认证

---

请求:

[https://drpserver\\_ip:port/doLogin?username=admin&password=admin](https://drpserver_ip:port/doLogin?username=admin&password=admin)

DRP 服务器认证通过后会回复 200 OK, 表示服务器接受 API 客户端后续请求。

- (2) 认证通过后, 发送设备序列号给 DRP 服务器:

请求:

[https://drpserver\\_ip:port/remoteWeb?product\\_sns=0123-4567-890A-BCDE](https://drpserver_ip:port/remoteWeb?product_sns=0123-4567-890A-BCDE)

DRP 服务器匹配到对应设备后回复 200 OK, 表示该 API 客户端后续请求会被转发给序列号指定的设备。

- (3) 后续根据具体功能发送对应请求, 参见 API 具体用法 (该模式 api 客户端的请求都发给 DRP 服务器)

---

# 20 常见问题

---

## 20.1 如何指定一个端口发送?

当需要指定特定端口发送短信时，请在发送短信请求中带上 `port` 参数，如不需要，则不用带上，网关会自动选择可用端口进行发送。

## 20.2 如何匹配短信发送请求和短信发送结果?

建议在短信发送请求中带上 `user_id` 这个参数，方便其和短信发送结果匹配。短信每发往一个号码，都带上一个 `user_id`，对应的发送结果也会带上这个 `user_id`，用于匹配。在您的程序中，`user_id` 可不断递增，使每条发送请求中的 `user_id` 都是唯一的。

## 20.3 一条短信需要发多久?

GSM 网络下一条短信的发送时间一般约为 5 至 8 秒，LTE 网络下为 2 至 4 秒。发送时间与网络有关，在网络状况很差的情况下，发送时间有可能长达 1 分钟。

## 20.4 怎样获得短信回执?

一般情况下，您的程序不需要去获取短信回执，因为短信是否送达会在短信发送结果中显示。

如果您希望自己来处理短信的送达，可以通过 `ref_id` 来匹配。对于一条长短信，如果总共分成了 3 段进行发送，假如首段的短信发送结果中的 `ref_id` 是 0，剩余的两段的 `ref_id` 是 1 和 2，那么就有至少 3 个短信回执，且 `ref_id` 分别是 0、1 和 2。由于 `ref_id` 的范围是 0~255，仅靠 `ref_id` 也不能准确匹配，因此建议 `ref_id` 结合号码来匹配短信回执。

## 20.5 如何禁掉短信回执?

在短信发送请求中，设置 `request_status_report` 参数为 `false`，即可不要求短信回执。

---

## 20.6 如何在程序中发送 HTTP 和 HTTPS 请求

C/C++以及 PHP 都可以使用 libcurl 来发送请求, 请求中需要带的参数与 curl 命令存在一一对应的关系。

命令行参数	libcurl 参数
-k	CURLOPT_SSL_VERIFYPEER, CURLOPT_SSL_VERIFYHOST,
-u	CURLOPT_USERPWD
--anyauth	CURLOPT_HTTPAUTH
-H	CURLOPT_HTTPHEADER
-d	CURLOPT_POSTFIELDSCURLOPT_POSTFIELDS

以 PHP 为例, curl 的设置如下

```
$curl = curl_init($url);  
curl_setopt($curl, CURLOPT_HTTPAUTH, CURLAUTH_ANY );  
curl_setopt($curl, CURLOPT_HEADER, true);  
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);  
curl_setopt($curl, CURLOPT_HTTPHEADER, array("Content-type:  
application/json;"));  
curl_setopt($curl, CURLOPT_USERPWD, "admin:admin");  
curl_setopt($curl, CURLOPT_POST, true);  
curl_setopt($curl, CURLOPT_POSTFIELDS, $content);  
curl_setopt($curl, CURLOPT_FOLLOWLOCATION, 1);  
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 0);  
curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, 0);
```

C#,JAVA,PYTHON 等其他语言有各自的库来发送, 请查询相关的资料