

# Yunpian API Documents

## 1. Overview

The Yunpian REST API allows you to send sms, query meta-data about your account, phone numbers, text messages, and recordings.

Since the API is based on REST principles, it's very easy to write and test applications. You can use your browser to access URLs, and you can use pretty much any HTTP client in any programming language to interact with the API.

## 2. Base URI

All URLs referenced in the documentation have the following base:

`http://yunpian.com/{version}`

OR

`https://yunpian.com/{version}`

{version} is the version number of API. Yunpian's current API version is v2.

## 3. Resource URI

`/{resource}/{function}.{format}`

{resource} indicates the resource type. It can be user/tpl/sms, and so on.

{function} indicates the operation on the resource provided. Base operation like get/set/send, and so on.

{format} indicates the data format you want to receive. Yunpian currently support json representations.

The Yunpian REST API is served over HTTP/HTTPS. To ensure data privacy, HTTPS is strongly suggested.

## 4. Request Formats

### Credentials

All requests to Yunpian's REST API require you to authenticate using [HTTP basic auth](#) to convey your identity. Your identity is identified by apikey (a 32 character string) provide by Yunpian, which is on the Account Dashboard page.

## 5. Response Formats

Yunpian currently supports returning resource representations as JSON. Simply add the .json extension to any resource URI. Here is an example of /v2/user/get.json resource represented as JSON:

```
{
  "http_status_code":400, //after called the response http status code
  code: 1,
  msg: "请求参数缺失", //missing require parameters
  detail: "参数 apikey 必须传入" //parameter apikey is required
}
```

## 6. REST Resources

### 1) Account

#### Query Account

URL: <https://yunpian.com/v2/user/get.json>

Method: POST

Properties:

PROPERTY	REQUIRED	DESCRIPTION	EXAMPLE
apikey	Yes	User ID	9b11127a9701975c734b8aee81e e3526

JSON representations:

```
{
  "nick": "Jacky",
  "gmt_created": "2012-09-11 15:14:00",
  "mobile": "13812341234",
  "email": "jacky@company.com",
  "ip_whitelist": null, //IP whitelist
  "api_version": "v2", //api version
  "balance": 1000, //balance
  "alarm_balance": 100, //sms or email notify if balance less
  than alarm_balance
}
```

```
"emergency_contact": "Jacky", //contract
"emergency_mobile": "13812341234" //mobile
}
```

## 2) Sms

### Send Message

URL: [https://yunpian.com/v2/sms/single\\_send.json](https://yunpian.com/v2/sms/single_send.json)

Method: POST

Properties:

PROPERTY	REQUIRED	DESCRIPTION	EXAMPLE
apikey	Yes	User ID	9b11127a9701975c734b8aee81e e3526
mobile	Yes	The destination phone number. multi mobiles are separated by comma	13812341234 13800000001,13800000002
text	Yes	The text of the message you want to send, limited to 500 words	<b>【Yunpian】</b> Your Verify Code is 1234
extend	No	Extend	001

		Number. Contact us if need	
uid	No	Use your own sms ID. Contact us if need	10001

JSON representations:

```
{
  "code":0,
  "msg":"发送成功", // message
  "count": 1, //sms num count
  "fee": 1, //sms fee count
  "sid": 1097, //sms ID
  "unit": RMB, // charge unit
  "mobile":13012312312, // send cell mobile
}
```

### Rate limiting

You can queue as many messages as you like. However, Yunpian have some rules for Harassment Prevention: The same text can only be send once in 30 seconds, no more than three times in 5 minutes. Contact us if you have special needs.

### Handling Message Status And Replies

Message status indicate whether the message is successfully receive by the target phone.

Reply is the text message reply by the user who received the text you just send.

There are two ways to handle statuses and replies.

- 1) Pull status or reply records using API
- 2) Specify a URL (which you can edit on the dashboard) to receive status or reply records push by Yunpian.

- 1) Pull status or reply records using API

### Pull status

URL: [http://yunpian.com/v2/sms/pull\\_status.json](http://yunpian.com/v2/sms/pull_status.json)

Method: POST

Properties:

PROPERTY	REQUIRED	DESCRIPTION	EXAMPLE
apikey	Yes	User ID	9b11127a9701975c734b8aee81e e3526
page_size	No	default page size is 20; max page size is 100	20

JSON representations:

```
[
  {
    "sid": 9527, //sms ID
    "uid": null, // sms ID defined by user. null if not set
    "user_receive_time": "2014-03-17 22:55:21", //sms receive time
    "error_msg": "", //error code if sms is not receive successfully
    "mobile": "15205201314", //target phone
    "report_status": "SUCCESS" //status:SUCCESS/FAIL/UNKNOWN
  },
  {
    "sid": 9528, //sms ID
    "uid": null, // sms ID defined by user. null if not set
    "user_receive_time": "2014-03-17 22:55:22", //sms receive time
    "error_msg": "", //error code if sms is not receive successfully
    "mobile": "15205201314", //target phone
    "report_status": "SUCCESS" //status:SUCCESS/FAIL/UNKNOWN
  }
]
```

### Pull reply

URL: [http://yunpian.com/v2/sms/pull\\_reply.json](http://yunpian.com/v2/sms/pull_reply.json)

Method: POST

Properties:

PROPERTY	REQUIRED	DESCRIPTION	EXAMPLE
apikey	Yes	User ID	9b11127a9701975c734b8aee81e e3526
page_size	No	default page size is 20; max page size is 100	20

JSON representations:

```
[
  {
    "mobile": "15205201314", // which phone reply this message
    "reply_time": "2014-03-17 22:55:21", //reply time
    "text": "Awesome, I love it! Thx", //reply text
    "extend": "01", //extend number if you set before, null by default
    "base_extend": "8888" //system default extend number
  },
  {
    "mobile": "15205201314", // which phone reply this message
    "reply_time": "2014-03-17 22:55:21", //reply time
    "text": "Awesome, I love it! Thx", //reply text
    "extend": "01", //extend number if you set before, null by default
    "base_extend": "8888" //system default extend number
  }
]
```

2) Specify a URL (which you can edit on the dashboard) to receive status or reply records push by Yunpian.

Yunpian will push status or reply to your URL in realtime. The URL your provided should able to process the http POST request, and response with an string of "0" or "SUCCESS" if you have rightly processed. Otherwise, Yunpian will try every 5 minutes for 3 times for status records, and every minute for 5 times for reply records.

Data format for status record

JSON representations:

```
[{
```

```

    "sid": 9527,
    "uid": null,
    "user_receive_time": "2014-03-17 22:55:21",
    "error_msg": "",
    "mobile": "15205201314",
    "report_status": "SUCCESS"
  },
  {
    "sid": 9528,
    "uid": null,
    "user_receive_time": "2014-03-17 22:55:23",
    "error_msg": "",
    "mobile": "15212341234",
    "report_status": "SUCCESS"
  }
}

```

Yunpian will post data like this:

Firstly, encode the data:

```
url_encode_json = urlencode(json_data)
```

Then, post data:

```
curl --data "sms_status=url_encode_json" http://your_receive_url_address
```

Data should be receive like this:

Firstly, decode the data:

```
json_data = urldecode(sms_status)
```

Then, return "0" or "SUCCESS"

Data format for reply record

JSON representations:

```

{
  "id": "2a70c6bb4f2845da816ea1bfe5732747",
  "mobile": "15205201314",
  "reply_time": "2014-03-17 22:55:21",
  "text": "copy that, over",
  "extend": "01",
  "base_extend": "8888"
}

```

Yunpian will post data like this:

Firstly, encode the data:

```
url_encode_json = urlencode(json_data)
```

Then, post data:

```
curl --data "sms_reply=url_encode_json" http://your_receive_url_address
```

Data should be receive like this:

Firstly, decode the data:

```
json_data = urldecode(sms_reply)
```

Then, return "0" or "SUCCESS"

So far, you are able to send message, check status of the message, or receive replies by using Yunpian's APIs.

For more APIs, please visit <http://www.yunpian.com>.

Question or advise of this document, please email to [kefu@yunpian.com](mailto:kefu@yunpian.com)

API usage demos with different languages are available on <http://www.yunpian.com/api/demo.html>

And, complete example codes are available on <https://github.com/yunpian>, which include example codes provide by third party developers. Any code contributions to Yunpian are great appreciated.