

Document Status	Document No.	YKZR_201908_100
<input checked="" type="checkbox"/> Draft <input type="checkbox"/> Released <input type="checkbox"/> Under revision	Prepared by	Xiang GAO
	Prepared on	August 15, 2019
	Secrecy level	Confidential
	Document version	1.9.2

## Open API Development Access Document of Xinye Cloud

### Project Version History

Date	Version	Description	Author	Reviewed by	Approved by
August 15, 2019	1.0.0	New document	Xiang GAO		
April 27, 2020	1.1.0	Add optional parameter mode at the print order interface	Xiang GAO		
July 31, 2020	1.2.0	Add print label order interface	Xiang GAO		
August 10, 2020	1.3.0	Add an amount broadcast interface; The print small ticket order interface supports the amount broadcast function	Xiang GAO		
September 10, 2020	1.4.0	Label printer supports 2D code printing	Xiang GAO		
December 24, 2020	1.5.0	The label printer supports 2D code printing, adding interface error code description and img label description	Li Gong		
July 12, 2021	1.5.1	Refine the interface description and add some label instructions	Li Gong		
September 18, 2021	1.5.2	Replace the two-dimensional code label of the printer interface, and add order timeliness parameter control	Li Gong		
March 15, 2022	1.6.0	Add instructions for POS instruction interface, cashbox control interface and voice extension interface	Li Gong		
June 29, 2022	1.7.0	Instructions for adding font selection settings to the label printer interface, adding font magnification of more than 2 times to the bill printer interface, and optimizing the print interface	Li Gong		
October 08, 2022	1.9.0	Interface for obtaining basic printer information , Currency symbol print label	Li Gong		
October 09, 2022	1.9.1	Clear som print order queues, Print interface added additional parameter usage description	Li Gong		
August 2, 2022	1.9.2	Sign SHA1 encryption example, Custom line to print the startion position label<LINE><HT> ,	Li Gong		

		Multilingual tag			
--	--	------------------	--	--	--

## Table of Contents

I. Access guide.....	4
II. Common parameters.....	5
III. Interface list.....	6

## I. Access Guide

### Steps for API access

#### Step one: Register a developer account

Go to [sg.open.xpyun.net](http://sg.open.xpyun.net) (hereinafter referred to as “Xinye cloud platform”) to register a developer account and obtain the UserKEY. The account and UserKEY are important identification parameters for developers to access the API.

#### Step two: Dock with the platform API

1. Add printer to the developer account (batch available) [required]

Note: Before printing, SN and KEY must be added to the developer account (check the label of the printer). The developer can add them manually on the management page of Xinye cloud platform or through this interface program.

2. Set the printer voice type

3. Print orders[billing machine must be connected]

4. Print label orders[Labeling machine must be connected]

5. Delete printer (batch available)

6. Update printer information

7. Delete the queue to be printed

8. Query whether the order is printed successfully

9. Query the order statistics of the printer on a certain day

10. Query printer status

11. Batch query printers status

12. Amount broadcast

## Special Announcement

1. API interface supports both the formats of http and https.

2. The request parameters of API include two parts: common parameters and private parameters. For details, see the corresponding API description below.

3. The request and return results of API are json format data.

Special note: the request written by the developer submits data in the form , and the request header needs to use content type: application/json; Charset=utf-8, please do not use other methods to submit data

## II. Common Parameters

### Description of request common parameters

### Parameters that must be passed for all requests.

Parameter name	Required or not	Type	Description
user	Required	string	Registered account of Xinye cloud platform (developer ID)
timestamp	Required	string	Current UNIX timestamp in 10 bits, accurate to seconds
sign	Required	string	<p>After the parameter <code>user + UserKEY + timestamp</code> is spliced (+ indicates connector), SHA1 encryption will produce a signature. The value is a 40-bit lowercase string, where <code>UserKEY</code> is a developer key for the user.</p> <p>For Example:</p> <p><code>user: 11223355@qq.com</code> ,</p> <p><code>UserKEY: 73f43bc289d44f5aad343d16bf2178451</code> ,</p> <p><code>timestamp : 1689065333</code></p> <p>combine :</p> <p><a href="#">11223355@qq.com73f43bc289d44f5aad343d16bf21784511689065333</a></p> <p>SHA1 encrypted data:</p> <p><code>0ec76a27a6396f00ad16e0b0ff7f547de138ea9f</code></p>
debug	Optional	string	When debug is equal to 1, it will return to non-json data, which is only used for testing.

Description of the sign parameter: for example: user=acc, userkey=abc, timestamp=acbc, then spell the string accabcacbc, and then encrypt the string with SHA1 to get the sign.

### Description of return common parameters

Parameter name	Type	Description
code	int	Return code, 0 indicates correct return, [Note: please use this return parameter to judge whether the result is correct], non-zero indicates return error.
msg	string	The result prompt message, "ok" for correct return. For an error, there will be a message prompt for return error.
data	not specified	See private return parameter data for data type and content. In case of an error, it will return to null.
serverExecutedTime	int	Server program executed time, unit: milliseconds.

### III. Interface List

#### 1. Add printers in batches

##### Function description:

When adding printers in batches, please refer to the request format description strictly.

##### Request interface:

https://sg.open.xpyun.net/api/openapi/xprinter/addPrinters

##### Request mode:

POST

##### Special note:

The request written by the developer himself should submit data in forms. The request header should use Content-Type: application/json;charset=UTF-8. Please do not submit data in other ways.

##### Common parameters:

Parameters that must be passed for all requests.

Parameter name	Required or not	Type	Description
user	Required	string	Registered account of Xinye cloud platform (developer ID)
timestamp	Required	string	Current UNIX timestamp in 10 bits, accurate to seconds
sign	Required	string	After the parameter <code>user + UserKEY + timestamp</code> is spliced (+ indicates connector), SHA1 encryption will produce a signature. The value is a 40-bit lowercase string, where <code>UserKEY</code> is a developer key for the user.
debug	Optional	string	When debug is equal to 1, it will return to non-json data, which is only used for testing.

Description of the sign parameter: for example: user=acc, userkey=abc, timestamp=acbc, then spell the string accabcacbc, and then encrypt the string with SHA1 to get the sign.

### Private parameter:

Parameter name	Required or not	Type	Description
items	Required	array	Array elements are json objects. { "cardno": "traffic card number", "idcode": "printer identifying code", "name": "printer name", "sn": "serial number of printer" } The fields of sn and name are required, and a maximum 50 sets can be added each time.

### Description of return parameter:

Parameter name	Type	Description
data	object	Return one json object, including successful and failed information. See the example for details.

### Request example:

```
{
  "items": [
    { "sn": "XPY123456789A", "name": "X58A", "idcode": "123", "cardno": "13432457418" },
    { "sn": "XPY987654321B", "name": "X58B", "idcode": "456", "cardno": "13531574658" }
  ],
  "user": "testuser@google.com",
  "timestamp": "1565417654",
  "sign": "82bdcb2cf6ac4923339b13c2aad1f95ddf0b0a8",
  "debug": "0"
}
```

### Return example:

**Example interpretation:** a total of 4 printer devices are added, 2 of which are added successfully and 2 of which are added unsuccessfully. The specific failure reason can be queried in the interface return code through the reason code 1010 in the failmsg field.

**Succeeded:**

```
{
  "msg": "ok",
  "code": 0,
  "data": { "success": ["XPY123456789A", "XPY987654321B"], "fail": [] },
  "serverExecutedTime": 1
}
```



**Failed:**

```
{
  "msg": "REQUEST_USER_NOT_REGISTER",
  "code": -4,
  "data": null,
  "serverExecutedTime": 1
}
```

**2. Set the voice type of the printer****Function description:**

Please refer to the request format description strictly to set the printer voice type.

**Request interface:**

<https://sg.open.xpyun.net/api/openapi/xprinter/setVoiceType>

**Request mode:**

POST

**Special note:**

The request written by the developer himself should submit data in forms. The request header should use Content-Type: application/json;charset=UTF-8. Please do not submit data in other ways.

**Common parameters:**

Parameters that must be passed for all requests.

Parameter name	Required or not	Type	Description
user	Required	string	Registered account of Xinye cloud platform (developer ID)
timestamp	Required	string	Current UNIX timestamp in 10 bits, accurate to seconds
sign	Required	string	After the parameter <code>user + UserKEY + timestamp</code> is spliced (+ indicates connector), SHA1 encryption will produce a signature. The value is a 40-bit lowercase string, where <code>UserKEY</code> is a developer key for the user.
debug	Optional	string	When debug is equal to 1, it will return to non-json data, which is only used for testing.

### Private parameter:

Parameter name	Required or not	Type	Description
sn	Required	string	The serial number of the printer
voiceType	Required	int	Voice type: The printer firmware version is v10.xx.Machine value:0 human voice (loud) 1 human voice (medium) 2 human voice (low) 3 Ticking 4 Mute sound Others::0 human voice (loud) 3 Ticking 4 Mute sound
volumeLevel	Optional	int	Voice: 0loud) 1medium 2low 3off <b>Description:</b> This parameter is supported by machines whose printer firmware version is not <b>V10.xx</b>

### Description of return parameter:

Parameter name	Type	Description
data	bool	Return to a Boolean type: true indicates a successful setting and false indicates a failed setting.

### Request example:

```
{
  "sn": "XPY123456789A",
  "voiceType": 0,
  "user": "testuser@google.com",
  "timestamp": "1565417654",
  "sign": "82bdcbe2cf6ac4923339b13c2aad1f95ddf0b0a8",
  "debug": "0"
}
```

### Return example:

Succeeded:

```
{
  "msg": "ok",
  "code": 0,
  "data": true,
  "serverExecutedTime": 1
}
```

Failed:

```
{
  "msg": "PRINTER_OFFLINE",
  "code": 1003,
  "data": false,
  "serverExecutedTime": 1
}
```

### 3. Print orders

#### Function description:

Send the order content that users need to Xinye cloud printer. This interface is only used to print bill orders. If you need to print label orders, please use print label orders.

#### Request mode:

<https://sg.open.xpyun.net/api/openapi/xprinter/print>

#### Request mode:

POST

#### Special note:

The request written by the developer himself should submit data in forms. The request header should use Content-Type: application/json;charset=UTF-8. Please do not submit data in other ways.

#### Common parameters:

Parameters that must be passed for all requests.

Parameter name	Required or not	Type	Description
user	Required	string	Registered account of Xinye cloud platform (developer ID)
timestamp	Required	string	Current UNIX timestamp in 10 bits, accurate to seconds
sign	Required	string	After the parameter <code>user + UserKEY + timestamp</code> is spliced (+ indicates connector), SHA1 encryption will produce a signature. The value is a 40-bit lowercase string, where <code>UserKEY</code> is a developer key for the user.
debug	Optional	string	When debug is equal to 1, it will return to non-json data, which is only used for testing.

Description of the sign parameter: for example: user=acc, userkey=abc, timestamp=acbc, then spell the string accabcacbc, and then encrypt the string with SHA1 to get the sign.

**Private parameter:**

Parameter name	Required or not	Type	Description
sn	Required	string	The serial number of the printer
content	Required	string	The content to be printed is judged by GBK code and can't exceed 12K.
copies	Optional	int	The number of printed copies is <b>1</b> by default.
voice	Optional	int	Sound playback mode, 0 Cancel order, 1 Mute, 2 为 Incoming orders, 3 users applying for chargeback, default 2
mode	Optional	int	Print mode: If the value is <b>0</b> or not specified, it will check whether the printer is online. If not online, it will not generate a print order and directly return the status code of an offline device. If online, it will generate a print order and return the print order number. If the value is <b>1</b> , it will not check whether the printer is online, directly generate a print order and return the print order number. If the printer is not online, the order will be cached in the print queue and will be printed automatically when the printer is normally online.
expiresIn	Optional	int	Unit of order validity period: second. When printing an order, the order that exceeds this time will not be automatically loaded for printing. When the parameter is set to 0, the system default setting value will be adopted. <b>When using this parameter, you need to set the parameter mode to 1.</b> Value range: $0 < \text{expiresIn} < 86400$ . Unit: second
backurlFlag	Optional	int	Print the order status callback ID. Value range: [ <b>1 - 5</b> ] integer, The print interface callback ID corresponding to the "function setting" menu interface in the management background of the Xinye Cloud platform.
payType	Optional	int	Payment method: Value range: 41~55: Alipay 41、WeChat 42、Cloud payment 43、UnionPay card swiping 44、UnionPay payment 45、Membership card consumption 46、Membership card recharge 47、Wing payment 48、Successful collection 49、Jialian payment 50、One wallet 51、JD payment 52、Quick payment 53、Wei payment 54、Money sharing payment 55 <b>Only used for Amount broadcast printer of Xinye.</b>

payMode	Optional	int	Payment or not: Value range:59~61 : refund 59 Account arrival 60 consumption 61。 <b>Only used for Amount broadcast printer of Xinye.</b>
money	Optional	float	Payment amount: At most 2 decimal places are allowed。 <b>Only used for Amount broadcast printer of Xinye.</b>
attached	Optional	string	The added value of push interface, generated by the developer platform, is pushed to the developer through the order status interface callback, and is valid only when the order status callback interface is connected. Cloud storage is valid for 24 hours.
backurlFlag	Optional	int	Printing order status callback identification. Value range: [ 1 - 5 ]integer. Corresponding to the print interface callback ID of the open platform management background function setting menu interface.

**Print template typesetting control tag description: (please refer to print template sample and SDK sample for details)**

<BR> : line break (if there is a closing tag (e.g. </C> ), it should be placed in front of the closing tag, two consecutive line breaks indicate adding a null string.  
 <L> </L> : left aligned  
 <C> </C> : center aligned  
 <R> </R> : right aligned  
 <N> </C> : normal font size  
 <HB> </HB> : double font in height  
 <WB> </WB> : double font in width  
 <B> </B> : double font in size  
 <CB> </CB> : double font in size centred  
 <HB2> </HB2> : three times the font in height  
 <WB2> </WB2> : three times the font in width  
 <B2> </B2> : three times the font in size  
 <BOLD> </BOLD> : bold font  
 <LOGO> </LOGO> : logo (the tag content is a character string in Base64 format, temporarily not opened)  
 <OR> </QR> : QR code (the tag content is a value of QR code, which cannot exceed 256 characters)  
 <BARCODE> </BARCODE> : barcode (the content is a value of barcode)  
 <CUT> : cutter command (active paper cutting, only valid for cutter printer. Note: the print order of cutter printer has a cutter instruction by default in the end.)  
 <HB2></HB2>: Double the font height  
 <WB2></WB2>: Double the font width  
 <B2></B2>: Double the font size  
 <BOLD></BOLD>: Bold font  
 <IMG></IMG>: To print a logo image, you need to log in to the open platform and upload it by setting the logo function under [print management →device management].Write empty tag here directly,For

example <IMG></IMG>.Please refer to the example。Picture width setting:Can be customized by <img> tag name, For example <IMG60> Indicates that the width is 60, Close label </IMG>You do not need to specify a height.<IMG> Label does not specify width, default is 40, The minimum value 20, The maximum value 200, Picture data Max 30KB

<QRCODE s=6 e=L l=center>QR code content</QRCODE>: QR code (The label content is a two-dimensional code value, which cannot exceed 256 characters at most) 。

s: QR code size, default 6, Value range 0-16, When s is 0, the QR code size will be dynamically adjusted according to the content size

e: QR code error correction level, default L, Value range L/M/Q/H

l: QR code printing position, default center, Value range left/center/right。 (L is the first letter of the location word)

<FONT w="0" h="0">Document contents to be enlarged</FONT>: Font label, font magnification.

Attribute w: Font width magnification.default 0, Do not enlarge, Value[0-7]

attribute h: Font height magnification.default 0, Do not enlarge, Value[0-7]

<BARCODE t=CODE128 w=2 h=100 p=2>Barcode value</BARCODE>: Barcode value (Label content is barcode value) 。

If the one-dimensional code cannot be printed normally, a <br> label shall be added before <barcode>

t: Barcode Type, Value range: UPCA、EAN13、EAN8、CODE39、ITF、CODABAR、CODE93、CODE128。 default: CODE128

w: Bar code width, Value range: 2-6; default : 2

h: Bar code height, Value range: 1-255; default : 100

p: Position of barcode value display, default: 0Do not display, 1 Above barcode, 2 Below barcode, 3Bar code is displayed above and below; default: 2

<RH n="3">Place the contents of line spacing to be changed</RH>: The label shall be set for the line spacing of the small ticket content, and it can only take effect when the content reaches two or more lines.<RH></RH>The content between is the content of the ticket to adjust the line spacing. The content can be the content wrapped by other labels of the ticket or plain text content.RHLabel needs to be placed on the base label<L></L>、 <C></C>or<R></R>take effect。 If the <bold> label is used, <rh> label shall be placed in the <bold> label.For the placement method, please refer to the use examples issued.Parameters are described as follows:

n: Line spacing value, n Value range[0,5], When setting, it is better to multiply n by 48 to be a non negative integer. If decimal places appear in the product, the decimal places will be discarded to get an integer.Default row spacing height 48 About3.75mm

#### Examples:

Print an example of a label with 3 line spacing and bold content

<R><BOLD><RH n="3">Welcome to Xinye printer, RH label is the label for setting the line spacing of the ticket content.It must to be placed between the basic labels L, C or R to take effect!</RH></BOLD></R><BR>

Customize inline print start position and jump <LINE p="3,6,16" /> <HT>

LINE define print positon, The parameters in p are separated by commas, not exceed 4, the parameter must be int, >0, <32 , <HT>turn to next position.The label can be printed in multiple columns with left alignment.No additional space is required for alignment.

#### Example:

<LINE p="8,14,26" />1<HT>2<HT>3<HT>45<BR>1<HT>2<HT>3<HT>45<BR>1<HT>2<HT>3<HT>45<BR>Cash on

delivery 12232323waA<BR>

Every line defines three start position 8,14 and 26.1 Normal printing, <HT>skip to the first position 2 , <HT> skip the next print position 3 ... and so on. The last one is usually added with <BR>to indicate the end of the line.

#### Multilingual tag

<lc> Chinese (include English and figure)

<le> Russian( Page 59 include English and figure)

<lr> Russian( Page 17 include English and figure)

<ls> Spanish( include English and figure)

<lh> Thai( include English and figure)

<lu> Uighur( include English and figure, Only valid on the first line and must be on the first line.)

<lv> Vietnamese( include English and figure)

Each language can only be mixed with Chinese and English.Mutual printing is not supported, Example,support Chinese English and Uighur.Mixed printing in Thai and Vietnamese is not supported.

#### Exam:

<lc> Russian, corresponding to printe, use example:<BR><le>Никогда не бойтесь <BR>

<lc> Russian, corresponding to, use example:<BR><lr>просто приятно <BR>

<lc> Spanish, corresponding to prins:<BR><ls>pues el niño recién nacido había <BR>

<P></P>: Turn on horizontal left and right layout mode. It shall be used in combination with the area label.It is only used for machines supporting left-right layout. For details, please add QQ technology group of open platform or enterprise wechat group to inquire

<AREA></AREA>: Print by area and use in combination with P label.It is only used for machines that support left-right layout. For details, you can add the open platform QQ technology group or enterprise wechat group to inquire

x:The x-axis of the starting point of the print area;

y:The Y-axis of the starting point of the printing area;

w:Width of the print area;

h:Height of the print area.

For example: <P><AREA x=0 y=0 w=100 h=200>For the content to be printed in this area, the text inside the area will wrap automatically</AREA></P>

Bill label effect diagram:

Bill layout effect diagram:



### Description of return parameter:

Parameter name	Type	Description
data	string	Return to order No. correctly

### Request example:

```
{
  "sn": "XPY123456789A",
  "content": "----cash on delivery----",
  "copies": 1,
  "user": "testuser@google.com",
  "timestamp": "1565417654",
  "sign": "82bdcbe2cf6ac4923339b13c2aad1f95ddf0b0a8",
  "debug": "0"
}
```

### Return example:

Succeeded:

```
{
  "msg": "ok",
  "code": 0,
  "data": "OM19081005450140628261",
  "serverExecutedTime": 23
}
```



Failed:

```
{
  "msg": "PRINTER_NOT_REGISTER",
  "code": 1002,
  "data": null,
  "serverExecutedTime": 25
}
```

#### 4、Print label order

Function Description:

Send the order content to be printed to the Xinye label printer (This interface can only be used by the Labeling machine. Currently, it supports the model (XP-T271U、XP-320B) , Do not use this interface for other printers)

Request interface:

<https://sg.open.xpyun.net/api/openapi/xprinter/printLabel>

Request mode:

POST

#### Special note:

- 1、The request written by the developer submits data in the form form, and the request header needs to use content type: application/json; Charset=utf-8, please do not use other methods to submit data;
- 2、The greater than sign and less than sign in the printed content (except the label) need to be translated before normal printing. Especially "<" is represented by "<" and ">" by ">"
- 3、1mm=8dots.

Common parameters:

Parameters that must be passed for all requests.

Parameter name	Required or not	Type	Description
user	Required	string	Registered account of Xinye cloud platform (developer ID)

timestamp	Required	string	Current UNIX timestamp in 10 bits, accurate to seconds
sign	Required	string	After the parameter <code>user + UserKEY + timestamp</code> is spliced (+ indicates connector), SHA1 encryption will produce a signature. The value is a 40-bit lowercase string, where <code>UserKEY</code> is a developer key for the user.
debug	Optional	string	When debug is equal to 1, it will return to non-json data, which is only used for testing.

**Description of the sign parameter:** for example: `user=acc, userkey=abc, timestamp=acbc`, then spell the string `accabcacbc`, and then encrypt the string with SHA1 to get the sign.

#### Private parameter:

Parameter name	Required or not	Type	Description
sn	Required	string	The serial number of the printer
content	Required	string	The content to be printed is judged by GBK code and can't exceed 12K.
copies	Optional	int	The number of printed copies is <b>1</b> by default.
voice	Optional	int	Sound playback mode, 0 Cancel order, 1 Mute, 2 为Incoming orders, 3 users applying for chargeback, default 2
mode	Optional	int	Print mode: If the value is <b>0</b> or not specified, it will check whether the printer is online. If not online, it will not generate a print order and directly return the status code of an offline device. If online, it will generate a print order and return the print order number. If the value is <b>1</b> , it will not check whether the printer is online, directly generate a print order and return the print order number. If the printer is not online, the order will be cached in the print queue and will be printed automatically when the printer is normally online.
expiresIn	Optional	int	Unit of order validity period: second. When printing an order, the order that exceeds this time will not be automatically loaded for

			printing.When the parameter is set to 0, the system default setting value will be adopted. <b>When using this parameter, you need to set the parameter mode to 1.</b> Value range:0 < expiresIn < 86400。 Unit: second
backurlFlag	Optional	int	Print the order status callback ID.Value range:[ 1 - 5 ]integer, The print interface callback ID corresponding to the "function setting" menu interface in the management background of the Xinye Cloud platform.
payType	Optional	int	Payment method: Value range:41~55: Alipay 41、WeChat 42、Cloud payment 43、UnionPay card swiping 44、UnionPay payment 45、Membership card consumption 46、Membership card recharge 47、Wing payment 48、Successful collection 49、Jialian payment 50、One wallet 51、JD payment 52、Quick payment 53、Wei payment 54、Money sharing payment 55 <b>Only used for Amount broadcast printer of Xinye.</b>
payMode	Optional	int	Payment or not: Value range:59~61: refund 59 Account arrival 60 consumption 61。 <b>Only used for Amount broadcast printer of Xinye.</b>
money	Optional	float	Payment amount: At most 2 decimal places are allowed。 <b>Only used for Amount broadcast printer of Xinye.</b>
attached	Optional	string	The added value of push interface, generated by the developer platform, is pushed to the developer through the order status interface callback, and is valid only when the order status callback interface is connected。Cloud storage is valid for 24 hours.
backurlFlag	Optional	int	Printing order status callback identification。 Value range: [ 1 - 5 ]integer. Corresponding to the print interface callback ID of the open platform management background function setting menu interface.

## Print template typesetting control tag description: (please refer to print template sample and SDK sample for details)

**<PAGE n="2"></PAGE>:**

Paging is used to support printing multiple different label pages. Not using this label means that all elements are only printed on one label page  
attribute n Number of copies printed, int, default 1

**<SIZE>width,height</SIZE>:**

Set label paper width and height, width:Width of label paper (excluding back paper), height: Height of label paper (excluding back paper), Unit mm, for example <SIZE>40,30</SIZE> , And it must be placed in the page tab to take effect. **The following attributes X and y are in dot units.** for exampleLabel paper size: width 40mm, height 30mm. 1mm=8dot

The maximum value of X 40\*8=320, Value range 0-320.

The maximum value of Y 30\*8=240, Value range 0-240.

**<TEXT x="10" y="100" font="9" w="1" h="2" r="0">CONTENT</TEXT>:**

Print the text, where:

Attribute x horizontal starting point coordinates (default 0)

Attribute y Vertical starting point coordinates (default 0)

Attribute font , Only the following parameters are supported:

- 1、 8 x 12 dot English digital font
- 2、 12 x 20 dot English digital font
- 3、 16 x 24 dot English digital font
- 4、 24 x 32 dot English digital font
- 5、 32 x 48 dot English digital font
- 6、 14 x 19 dot English digital font OCR-B
- 7、 21 x 27 dot English digital font OCR-B
- 8、 14 x25 dot English digital font OCR-A
- 9、 Simplified Chinese 24 x 24 font (GB)

Attribute w Word width magnification1-10 (default 1)

Attribute h Word height magnification1-10 (default 1)

Attribute r Rotate angle for word(Clockwise, default 0), Only the following parameter value settings are supported:

- 0 0 degree
- 90 90 degree
- 180 180 degree
- 270 270 degree

**<BC128 x="10" y="100" h="60" s="1" n="1" w="1" r="0">1234567</BC128>:**

Print code128 one-dimensional code, where:

Attribute x Horizontal starting point coordinates (default 0)

Attribute y Vertical starting point coordinates (default 0)

Attribute h the bar code height (default 48)

Attribute s Whether it can be recognized by human eyes: 0 no, 1 yes (default 1)

Attribute n narrow bar width, Expressed in dots (default 1)

Attribute w width bar width, Expressed in dots (default 1)

Attribute r Rotate angle for word (Clockwise, default 0), Only the following parameter value settings are supported:

0 0 degree

90 90 degree

180 180 degree

270 degree

`<BC39 x="10" y="100" h="60" s="1" n="1" w="1" r="0">1234567</BC39>`:

Print code39 one-dimensional code, where:

Attribute x Horizontal starting point coordinates (default 0)

Attribute y Vertical starting point coordinates (default 0)

Attribute h the bar code height (default 48)

Attribute s Whether it can be recognized by human eyes: 0 no, 1 yes (default 1)

Attribute n narrow bar width, Expressed in dots (default 1)

Attribute w width bar width, Expressed in dots (default 1)

Attribute r Rotate angle for word (Clockwise, default 0), Only the following parameter value settings are supported:

0 0 degree

90 90 degree

180 180 degree

270 degree

`<QRC x="20" y="20" s="2" e="L">QR code content</QRC>`:

Print the QR code. The label content is the QR code value, which cannot exceed 256 characters, including:

Attribute x Horizontal starting point coordinates (default 0), Value range (0 to Label width\*8)

1 mm=8 dots

Attribute y Vertical starting point coordinates (default 0), Value range (0 to Label height\*8)

1 mm=8 dots

Attribute s QR code print size(default 2 Value range 1-10)

Attribute e QR code error correction level(default L Value range L/M/Q/H)

`<IMG x="16" y="32" w="100">`:

To print a logo image, you need to log in to the open platform and upload it by setting the logo function under [print management →device management].Directly here Write an empty tag. If a <page> tag is specified, <img> tag should be placed in the <page> tag, <img>, such as <img>, with

See the example for the volume. Of which:

Attribute x Horizontal starting point coordinates (default 0)

Attribute y Vertical starting point coordinates (default 0)

Attribute w logo picture maximum width (default 50), minimum value20, maximum value100.

The height and width of the logo picture are equal

```
<SEQ x="8" y="8" xe="100" ye="100" s="4">:
```

Print the box,where:

- Attribute x Horizontal starting point coordinates (default 0)
- Attribute y Vertical starting point coordinates (default 0)
- Attribute xe the horizontal end point coordinate (default 0)
- Attribute ye the vertical end point coordinate (default 0)
- Attribute s Print size for box lines (default 4 Value range 1-10) 。

```
<L x="20" y="20" w="4" h="250">:
```

Print table lines. Of which:

- Attribute x Horizontal starting point coordinates (default 0)
- Attribute y Vertical starting point coordinates (default 0)
- Attribute w Width of line (default 4 Value range 1-1000)
- Attribute h Height of line (default 4 Value range 1-1000)

Example drawing of printing effect of labeling machine:



Description of return parameter:

Parameter name	Type	Description
data	string	Return one json object, including successful and failed information. See the example for details.

Request example:

```
{
  "sn": "XPY123456789A",
  "content": "<TEXT x=\"10\" y=\"100\" w=\"1\" h=\"2\" r=\"0\">test content</TEXT>",
  "copies": 1,
  "voice": 2,
  "user": "testuser@google.com",
```

```
"timestamp": "1565417654",  
"sign": "82bdcbe2cf6ac4923339b13c2aad1f95ddf0b0a8",  
"debug": "0"  
}
```

### Return example:

**Succeeded:**When the printer is offline: MSG is printer\_ Offline otherwise, it is OK

```
{  
  "msg": "ok",  
  "code": 0,  
  "data": "OM19081005450140628261",  
  "serverExecutedTime": 23  
}
```

### Failed:

```
{  
  "msg": "PRINTER_NOT_REGISTER",  
  "code": 1002,  
  "data": null,  
  "serverExecutedTime": 25  
}
```

## 5. Delete printer in batches

### Function description:

Delete printer in batches

### Request interface:

<https://sg.open.xpyun.net/api/openapi/xprinter/delPrinters>

### Request mode:

POST

### Special note:

The request written by the developer himself should submit data in forms. The request header should use Content-Type: application/json;charset=UTF-8. Please do not submit data in other ways.

### Common parameters:

Parameters that must be passed for all requests.

Parameter name	Required or not	Type	Description
user	Required	string	Registered account of Xinye cloud platform (developer ID)
timestamp	Required	string	Current UNIX timestamp in 10 bits, accurate to seconds
sign	Required	string	After the parameter <code>user + UserKEY + timestamp</code> is spliced (+ indicates connector), SHA1 encryption will produce a signature. The value is a 40-bit lowercase string, where <code>UserKEY</code> is a developer key for the user.
debug	Optional	string	When debug is equal to 1, it will return to non-json data, which is only used for testing.

Description of the sign parameter: for example: user=acc, userkey=abc, timestamp=acbc, then spell the string accabcacbc, and then encrypt the string with SHA1 to get the sign.

### Private parameter:

Parameter name	Required or not	Type	Description
snlist	Required	array	A set of printer serial number, which is string array.

### Description of return parameter:

Parameter name	Type	Description
data	object	Return one json object, including successful and failed information. See the example for details.

### Request example:

```
{
  "snlist": ["XPY123456789A", "XPY987654321B"],
  "user": "testuser@google.com",
  "timestamp": "1565417654",
  "sign": "82bdcbe2cf6ac4923339b13c2aad1f95ddf0b0a8",
  "debug": "0"
}
```

### Return example:

Succeeded:



```
{
  "msg": "ok",
  "code": 0,
  "data": { "success": ["XPY123456789A", "XPY987654321B"], "fail":
    [] },
  "serverExecutedTime": 1
}
```

#### Failed:

```
{
  "msg": "REQUEST_USER_NOT_REGISTER",
  "code": -4,
  "data": null,
  "serverExecutedTime": 1
}
```

## 6. Update the printer information

### Function description:

Update the printer information

### Request interface:

<https://sg.open.xpyun.net/api/openapi/xprinter/updPrinter>

### Request mode:

POST

### Special note:

The request written by the developer himself should submit data in forms. The request header should use Content-Type: application/json;charset=UTF-8. Please do not submit data in other ways.

### Common parameters:

Parameters that must be passed for all requests.

Parameter name	Required or not	Type	Description
----------------	-----------------	------	-------------

user	Required	string	Registered account of Xinye cloud platform (developer ID)
timestamp	Required	string	Current UNIX timestamp in 10 bits, accurate to seconds
sign	Required	string	After the parameter <code>user + UserKEY + timestamp</code> is spliced (+ indicates connector), SHA1 encryption will produce a signature. The value is a 40-bit lowercase string, where <code>UserKEY</code> is a developer key for the user.
debug	Optional	string	When debug is equal to 1, it will return to non-json data, which is only used for testing.

**Description of the sign parameter:** for example: user=acc, userkey=abc, timestamp=acbc, then spell the string accabcacbc, and then encrypt the string with SHA1 to get the sign.

### Private parameter:

Parameter name	Required or not	Type	Description
sn	Required	string	Serial number of the printer
name	Required	string	Name of the printer
cardno	Optional	string	Traffic card number of the printer

### Description of return parameter:

Parameter name	Type	Description
data	bool	Return to a Boolean type: true indicates a successful setting and false indicates a failed setting.

### Request example:

```
{
  "sn": "XPY123456789A",
  "name": "X58C1",
  "cardno": "13031547528",
  "user": "testuser@google.com",
  "timestamp": "1565417654",
  "sign": "82bdcbe2cf6ac4923339b13c2aad1f95ddf0b0a8",
  "debug": "0"
}
```

### Return example:

Succeeded:

```
{
  "msg": "ok",
  "code": 0,
  "data": true,
  "serverExecutedTime": 1
}
```

```
}
```

**Failed:**

```
{
  "msg": "SN_USER_NOT_MATCH",
  "code": 1001,
  "data": false,
  "serverExecutedTime": 1
}
```

## 7. Delete the queue to be printed

**Function description:**

Delete the queue to be printed of a specific printer

**Request interface:**

<https://sg.open.xpyun.net/api/openapi/xprinter/delPrinterQueue>

**Request mode:**

POST

**Special note:**

The request written by the developer himself should submit data in forms. The request header should use Content-Type: application/json;charset=UTF-8. Please do not submit data in other ways.

**Common parameters:**

Parameters that must be passed for all requests.

Parameter name	Required or not	Type	Description
user	Required	string	Registered account of Xinye cloud platform (developer ID)
timestamp	Required	string	Current UNIX timestamp in 10 bits, accurate to seconds
sign	Required	string	After the parameter <code>user + UserKEY + timestamp</code> is spliced (+ indicates connector), SHA1 encryption will

			produce a signature. The value is a 40-bit lowercase string, where <a href="#">UserKEY</a> is a developer key for the user.
debug	Optional	string	When debug is equal to 1, it will return to non-json data, which is only used for testing.

### Private parameter:

Parameter name	Required or not	Type	Description
sn	Required	string	Serial number of the printer

### Description of return parameter:

Parameter name	Type	Description
data	bool	Return to a Boolean type: true indicates a successful setting and false indicates a failed setting.

### Request example:

```
{
  "sn": "XPY123456789A",
  "user": "testuser@google.com",
  "timestamp": "1565417654",
  "sign": "82bdcbe2cf6ac4923339b13c2aad1f95ddf0b0a8",
  "debug": "0"
}
```

### Return example:

#### Succeeded:

```
{
  "msg": "ok",
  "code": 0,
  "data": true,
  "serverExecutedTime": 1
}
```

#### Failed:

```
{
  "msg": "PRINTER_NOT_REGISTER",
  "code": 1002,
  "data": false,
  "serverExecutedTime": 1
}
```

## 8. Query whether the order is printed successfully

### Function description:

Query whether the order is printed successfully according to the order ID. The order ID is returned by the “print order” interface.

### Request interface:

<https://sg.open.xpyun.net/api/openapi/xprinter/queryOrderState>

### Request mode:

POST

### Special note:

The request written by the developer himself should submit data in forms. The request header should use Content-Type: application/json;charset=UTF-8. Please do not submit data in other ways.

### Common parameters:

Parameters that must be passed for all requests.

Parameter name	Required or not	Type	Description
user	Required	string	Registered account of Xinye cloud platform (developer ID)
timestamp	Required	string	Current UNIX timestamp in 10 bits, accurate to seconds
sign	Required	string	After the parameter <code>user + UserKEY + timestamp</code> is spliced (+ indicates connector), SHA1 encryption will produce a signature. The value is a 40-bit lowercase string, where <code>UserKEY</code> is a developer key for the user.
debug	Optional	string	When debug is equal to 1, it will return to non-json data, which is only used for testing.

Description of the sign parameter: for example: user=acc, userkey=abc, timestamp=acbc, then spell the string accabcacbc, and then encrypt the string with SHA1 to get the sign.

### Private parameter:

Parameter name	Required or not	Type	Description
orderId	Required	string	The order ID is returned by the “print order” interface.

### Description of return parameter:

Parameter name	Type	Description
data	bool	True indicates return after printed and false indicates return not printed.

**Request example:**

```
{
  "orderId": "OM19081005152569029225",
  "user": "testuser@google.com",
  "timestamp": "1565417654",
  "sign": "82bdcbe2cf6ac4923339b13c2aad1f95ddf0b0a8",
  "debug": "0"
}
```

**Return example:****Succeeded:**

```
{
  "msg": "ok",
  "code": 0,
  "data": true,
  "serverExecutedTime": 1
}
```

**Failed:**

```
{
  "msg": "ORDER_NOT_FOUND",
  "code": 1005,
  "data": false,
  "serverExecutedTime": 1
}
```

**9. Query the order statistics of a specified printer on a certain day****Function description:**

Query the order statistics of a specified printer on a certain day, and return the order quantity printed and to be printed.

**Request interface:**

<https://sg.open.xpyun.net/api/openapi/xprinter/queryOrderStatis>

**Request mode:**

POST

### Special note:

The request written by the developer himself should submit data in forms. The request header should use Content-Type: application/json;charset=UTF-8. Please do not submit data in other ways.

### Common parameters:

Parameters that must be passed for all requests.

Parameter name	Required or not	Type	Description
user	Required	string	Registered account of Xinye cloud platform (developer ID)
timestamp	Required	string	Current UNIX timestamp in 10 bits, accurate to seconds
sign	Required	string	After the parameter <code>user + UserKEY + timestamp</code> is spliced (+ indicates connector), SHA1 encryption will produce a signature. The value is a 40-bit lowercase string, where <code>UserKEY</code> is a developer key for the user.
debug	Optional	string	When debug is equal to 1, it will return to non-json data, which is only used for testing.

Description of the sign parameter: for example: user=acc, userkey=abc, timestamp=acbc, then spell the string accabcacbc, and then encrypt the string with SHA1 to get the sign.

### Private parameter:

Parameter name	Required or not	Type	Description
sn	Required	string	Serial number of the printer
date	Required	string	Query date, format in yy-MM-dd, e.g. 2019-08-15

### Description of return parameter:

Parameter name	Type	Description
data	object	Json object, return the order quantity printed and to be printed, e.g. {"printed": 2,"waiting": 0}

### Request example:

```
{
  "sn": "XPY123456789A",
  "date": "2019-08-15",
  "user": "testuser@google.com",
  "timestamp": "1565417654",
  "sign": "82bdcbe2cf6ac4923339b13c2aad1f95ddf0b0a8",
  "debug": "0"
}
```

### Return example:

**Succeeded:**

```
{
  "msg": "ok",
  "code": 0,
  "data": {"printed": 2, "waiting": 0},
  "serverExecutedTime": 1
}
```

**Failed:**

```
{
  "msg": "ORDER_DATE_INVALID",
  "code": 1006,
  "data": null,
  "serverExecutedTime": 1
}
```

**10. Obtain the status of a specified printer****Function description:**

Query the status of a specified printer and return to online or offline, normal or abnormal status information of the printer.

**Request interface:**

<https://sg.open.xpyun.net/api/openapi/xprinter/queryPrinterStatus>

**Request mode:**

POST

**Special note:**

The request written by the developer himself should submit data in forms. The request header should use Content-Type: application/json;charset=UTF-8. Please do not submit data in other ways.

**Common parameters:**

Parameters that must be passed for all requests.

Parameter name	Required or not	Type	Description
user	Required	string	Registered account of Xinye cloud platform (developer ID)
timestamp	Required	string	Current UNIX timestamp in 10 bits, accurate to seconds
sign	Required	string	After the parameter <code>user + UserKEY + timestamp</code> is spliced (+ indicates connector), SHA1 encryption will



			produce a signature. The value is a 40-bit lowercase string, where <b>UserKEY</b> is a developer key for the user.
debug	Optional	string	When debug is equal to 1, it will return to non-json data, which is only used for testing.

**Description of the sign parameter:** for example: user=acc, userkey=abc, timestamp=acbc, then spell the string accabcacbc, and then encrypt the string with SHA1 to get the sign.

#### Private parameter:

Parameter name	Required or not	Type	Description
sn	Required	string	Serial number of the printer

#### Description of return parameter:

Parameter name	Type	Description
data	int	Return to the printer status value, three types in total: 0 indicates offline status. 1 indicates online and normal status. 2 indicates online and abnormal status. Remarks: Abnormal status means lack of paper, if the printer has been out of contact with the server for more than 30s, it can be confirmed to be offline status.

#### Request example:

```
{
  "sn": "XPY123456789A",
  "user": "testuser@google.com",
  "timestamp": "1565417654",
  "sign": "82bdcbe2cf6ac4923339b13c2aad1f95ddf0b0a8",
  "debug": "0"
}
```

#### Return example:

##### Succeeded:

```
{
  "msg": "ok",
  "code": 0,
  "data": 1,
  "serverExecutedTime": 1
}
```

##### Failed:

```
{
  "msg": "REQUEST_PARAM_INVALID",
  "code": -2,
  "data": 0,
  "serverExecutedTime": 1
}
```

## 11、Obtain the status of a specified printer in batch

### Function description:

Obtain the status of the specified printer in batch, and return the online or offline, normal or abnormal status information of the printer.

### Request interface:

<https://sg.open.xpyun.net/api/openapi/xprinter/queryPrintersStatus>

### Request mode:

POST

### Special note:

The request written by the developer himself should submit data in forms. The request header should use Content-Type: application/json;charset=UTF-8. Please do not submit data in other ways.

### Common parameters:

Parameters that must be passed for all requests.

Parameter name	Required or not	Type	Description
user	Required	string	Registered account of Xinye cloud platform (developer ID)
timestamp	Required	string	Current UNIX timestamp in 10 bits, accurate to seconds
sign	Required	string	After the parameter <code>user + UserKEY + timestamp</code> is spliced (+ indicates connector), SHA1 encryption will produce a signature. The value is a 40-bit lowercase string, where <code>UserKEY</code> is a developer key for the user.
debug	Optional	string	When debug is equal to 1, it will return to non-json data, which is only used for testing.

Description of the sign parameter: for example: user=acc, userkey=abc, timestamp=acbc, then spell the string accabcacbc, and then encrypt the string with SHA1 to get the sign.

#### Private parameter:

Parameter name	Required or not	Type	Description
snlist	Required	array	A set of printer serial number (Query up to 20 devices at a time)

#### Description of return parameter:

Parameter name	Type	Description
data	array	Return to the printer status value, three types in total: 0 indicates offline status. 1 indicates online and normal status. 2 indicates online and abnormal status. Remarks: Abnormal status means lack of paper, if the printer has been out of contact with the server for more than 30s, it can be confirmed to be offline status.

#### Request example:

```
{
  "snlist": ["XPY123456789A", "XPY123456789B"],
  "user": "testuser",
  "timestamp": "1565417654",
  "sign": "82bdcbe2cf6ac4923339b13c2aad1f95ddf0b0a8",
  "debug": "0"
}
```

#### Return example:

Succeeded:

```
{
  "msg": "ok",
  "code": 0,
  "data": [0, 1],
  "serverExecutedTime": 1
}
```

Failed:

```
{
  "msg": "REQUEST_PARAM_INVALID",
  "code": -2,
}
```

```
"data":[0, 0],
"serverExecutedTime":1
}
```

## 12、Amount broadcast

### Function description:

Send the voice content that the user needs to broadcast to the Xinye printer that supports the amount broadcast

### Request interface:

<https://sg.open.xpyun.net/api/openapi/xprinter/playVoice>

### Request mode:

POST

### Special note:

The request written by the developer himself should submit data in forms. The request header should use Content-Type: application/json;charset=UTF-8. Please do not submit data in other ways.

### Common parameters:

Parameters that must be passed for all requests.

Parameter name	Required or not	Type	Description
user	Required	string	Registered account of Xinye cloud platform (developer ID)
timestamp	Required	string	Current UNIX timestamp in 10 bits, accurate to seconds
sign	Required	string	After the parameter <code>user + UserKEY + timestamp</code> is spliced (+ indicates connector), SHA1 encryption will produce a signature. The value is a 40-bit lowercase string, where <code>UserKEY</code> is a developer key for the user.
debug	Optional	string	When debug is equal to 1, it will return to non-json data, which is only used for testing.

**Description of the sign parameter:** for example: user=acc, userkey=abc, timestamp=acbc, then spell the string accabcacbc, and then encrypt the string with SHA1 to get the sign.

**Private parameter:**

Parameter name	Required or not	Type	Description
sn	Required	string	Serial number of the printer
payType	Optional	int	Payment method: Value range:41~55: Alipay 41、WeChat 42、Cloud payment 43、UnionPay card swiping 44、UnionPay payment 45、Membership card consumption 46、Membership card recharge 47、Wing payment 48、Successful collection 49、Jialian payment 50、One wallet 51、JD payment 52、Quick payment 53、Wei payment 54、Money sharing payment 55 Only used for Amount broadcast printer of Xinye.
payMode	Optional	int	Payment or not: Value range:59~61: refund 59 Account arrival 60 consumption 61。 Only used for Amount broadcast printer of Xinye.
money	Optional	float	Payment amount: At most 2 decimal places are allowed。 Only used for Amount broadcast printer of Xinye.

**Description of return parameter:**

Parameter name	Type	Description
data	string	Return 0 correctly

**Request example:**

```
{
  "sn": "XPY123456789A",
  "payType": 41,
  "payMode": 60,
  "money": 25.5,
  "user": "testuser@google.com",
  "timestamp": "1565417654",
  "sign": "82bdcbe2cf6ac4923339b13c2aad1f95ddf0b0a8",
  "debug": "0"
```

```
}
```

**Return example:****Succeeded:**

```
{  
  "msg": "ok",  
  "code": 0,  
  "data": "0",  
  "serverExecutedTime": 23  
}
```

**Failed:**

```
{  
  "msg": "PRINTER_NOT_REGISTER",  
  "code": 1002,  
  "data": null,  
  "serverExecutedTime": 25  
}
```

**13、POS printing instruction****Function description:**

After the order content to be printed by the user is encoded by POS instruction, base64 encryption is used to send it to the core Ye small ticket printer. For POS instruction programming documents, reference <https://www.xprinter.net/companyfile/1/> 58 Printer reference 《Xinye 58 series Chinese programming manual》 Or 80 Printer 《Xinye 80 series Chinese programming manual》

**Request interface:**

<https://sg.open.xpyun.net/api/openapi/xprinter/pos>

**Request mode:**

POST

**Special note:**

The request written by the developer himself should submit data in forms. The request header should use Content-Type: application/json;charset=UTF-8. Please do not submit data in other ways.

**Common parameters:**

Parameters that must be passed for all requests.

Parameter name	Required or not	Type	Description
user	Required	string	Registered account of Xinye cloud platform (developer ID)
timestamp	Required	string	Current UNIX timestamp in 10 bits, accurate to seconds
sign	Required	string	After the parameter <code>user + UserKEY + timestamp</code> is spliced (+ indicates connector), SHA1 encryption will produce a signature. The value is a 40-bit lowercase string, where <code>UserKEY</code> is a developer key for the user.
debug	Optional	string	When debug is equal to 1, it will return to non-json data, which is only used for testing.

**Description of the sign parameter: for example: user=acc, userkey=abc, timestamp=acbc, then spell the string accabcacbc, and then encrypt the string with SHA1 to get the sign.**

**Private parameter:**

Parameter name	Required or not	Type	Description
sn	Required	string	Serial number of the printer
content	Required	string	The print content and POS instruction data recognized by the printer after base64 encryption cannot exceed 12K
copies	Optional	int	The number of copies printed is 1 by default and the maximum value is 5. If it exceeds the range, it will be regarded as an invalid parameter

**Description of return parameter:**

Parameter name	Type	Description
data	string	Return 0 correctly

**Request example:**

```
{  
  "sn": "XPY123456789A",
```

```
"content": "QkJCNkQzQURDQUI5RDNDM0Q3QThEMkI1NTA0RjUzQzZCMUJFRERCNEYyRD  
DNBMUJCRkFCOUFCQ0JCRUM5RkFCMkZBQjVDNEQ1RUJCNEYyQTNBQ0M4QzhDM0Y0Q0  
ZCNUMxRDBCNEYyRDNDNMUJCRkEYMTBEMEE=",  
"copies": 1,  
"user": "testuser@google.com",  
"timestamp": "1565417654",  
"sign": "82bdcbe2cf6ac4923339b13c2aad1f95ddf0b0a8",  
"debug": "0"  
}
```

### Return example:

#### Succeeded:

```
{  
  "msg": "ok",  
  "code": 0,  
  "data": "OM19081005450140628261",  
  "serverExecutedTime": 23  
}
```

#### Failed:

```
{  
  "msg": "PRINTER_NOT_REGISTER",  
  "code": 1002,  
  "data": null,  
  "serverExecutedTime": 25  
}
```

## 14、Cashbox control

### Function description:

Use the printer to control the opening of the cash box. The input voltage of the cash box must be within the output voltage range of the cash box port of the printer, otherwise the cash box may not be opened normally

### Request interface:

<https://sg.open.xpyun.net/api/openapi/xprinter/controlBox>

### Request mode:

POST

### Special note:



The request written by the developer himself should submit data in forms. The request header should use Content-Type: application/json;charset=UTF-8. Please do not submit data in other ways.

### Common parameters:

Parameters that must be passed for all requests.

Parameter name	Required or not	Type	Description
user	Required	string	Registered account of Xinye cloud platform (developer ID)
timestamp	Required	string	Current UNIX timestamp in 10 bits, accurate to seconds
sign	Required	string	After the parameter <code>user + UserKEY + timestamp</code> is spliced (+ indicates connector), SHA1 encryption will produce a signature. The value is a 40-bit lowercase string, where <code>UserKEY</code> is a developer key for the user.
debug	Optional	string	When debug is equal to 1, it will return to non-json data, which is only used for testing.

Description of the sign parameter: for example: user=acc, userkey=abc, timestamp=acbc, then spell the string accabcacbc, and then encrypt the string with SHA1 to get the sign.

### Private parameter:

Parameter name	Required or not	Type	Description
sn	Required	string	Serial number of the printer

### Description of return parameter:

Parameter name	Type	Description
data	string	Return 0 correctly

### Request example:

```
{
  "sn": "XPY123456789A",
  "user": "testuser@google.com",
  "timestamp": "1565417654",
  "sign": "82bdcbe2cf6ac4923339b13c2aad1f95ddf0b0a8",
  "debug": "0"
```

```
}
```

**Return example:****Succeeded:**

```
{
  "msg": "ok",
  "code": 0,
  "data": "0",
  "serverExecutedTime": 23
}
```

**Failed:**

```
{
  "msg": "PRINTER_NOT_REGISTER",
  "code": 1002,
  "serverExecutedTime": 25
}
```

**15、Extended voice broadcast****Function description:**

Send the voice content that users need to broadcast to the Xinye printer that supports predefined voice

**Request interface:**

<https://sg.open.xpyun.net/api/openapi/xprinter/playVoiceExt>

**Request mode:**

POST

**Special note:**

The request written by the developer himself should submit data in forms. The request header should use Content-Type: application/json;charset=UTF-8. Please do not submit data in other ways.

**Common parameters:**

Parameters that must be passed for all requests.

Parameter name	Required or not	Type	Description
user	Required	string	Registered account of Xinye cloud platform (developer ID)
timestamp	Required	string	Current UNIX timestamp in 10 bits, accurate to seconds
sign	Required	string	After the parameter <b>user + UserKEY + timestamp</b> is spliced (+ indicates connector), SHA1 encryption will produce a signature. The value is a 40-bit lowercase string, where <b>UserKEY</b> is a developer key for the user.
debug	Optional	string	When debug is equal to 1, it will return to non-json data, which is only used for testing.

**Description of the sign parameter: for example: user=acc, userkey=abc, timestamp=acbc, then spell the string accabcacbc, and then encrypt the string with SHA1 to get the sign.**

#### Private parameter:

Parameter name	Required or not	Type	Description
sn	Required	string	Serial number of the printer
content	Required	string	Voice broadcast content, such as #62, will broadcast the voice content corresponding to the serial number of 62.This content needs to be added to the Xinye cloud technical support QQ group (856926694) consultation
voiceTime	Optional	int	Voice broadcast times, default 1
voiceInterval	Required	int	Voice broadcast many times. The time interval between the current broadcast and the next broadcast is valid only when the number of broadcasts is greater than 1.

#### Description of return parameter:

Parameter name	Type	Description
data	string	Return 0 correctly

#### Request example:

```
{
  "sn": "XPY123456789A",
  "content": "#62#71#63#92",
  "voiceInterval": 1,
  "voiceTime": 1,
```

```
"user": "testuser@google.com",
"timestamp": "1565417654",
"sign": "82bdcbe2cf6ac4923339b13c2aad1f95ddf0b0a8",
"debug": "0"
}
```

**Return example:****Succeeded:**

```
{
  "msg":"ok",
  "code":0,
  "data":"0",
  "serverExecutedTime":23
}
```

**Failed:**

```
{
  "msg":"PRINTER_NOT_REGISTER",
  "code":1002,
  "data":null,
  "serverExecutedTime":25
}
```

**16、Obtain information of printer****Function description:**

Developers can use the printer information acquisition interface to make reasonable judgements based on the obtained printer information and business situation.

**Request interface:**

<https://open.xpyun.net/api/openapi/xprinter/printerInfo>

**Request mode:**

POST

**Special note:**

The request written by the developer himself should submit data in forms. The request header should use Content-Type: application/json;charset=UTF-8. Please do not submit data in other ways.

### Common parameters:

Parameters that must be passed for all requests.

Parameter name	Required or not	Type	Description
user	Required	string	Registered account of Xinye cloud platform (developer ID)
timestamp	Required	string	Current UNIX timestamp in 10 bits, accurate to seconds
sign	Required	string	After the parameter <code>user + UserKEY + timestamp</code> is spliced (+ indicates connector), SHA1 encryption will produce a signature. The value is a 40-bit lowercase string, where <code>UserKEY</code> is a developer key for the user.
debug	Optional	string	When debug is equal to 1, it will return to non-json data, which is only used for testing.

Description of the sign parameter: for example: user=acc, userkey=abc, timestamp=acbc, then spell the string accabcacbc, and then encrypt the string with SHA1 to get the sign.

### Private parameter:

Parameter name	Required or not	Type	Description
sn	Required	string	Serial number of the printer

### Description of return parameter:

Parameter name	Type	Description
----------------	------	-------------

data	Object	<p>JSON Object。Printer base information,field description:</p> <p>name: printer name</p> <p>sn: serial number of the printer</p> <p>status: printer status, Offline、OnlinNormal、OnlineMissingPaper</p> <p>type: printer type 00: 58 printer 01: 58 label printer 02: 80 printer 03: 80 label printer</p> <p>version: printer fixed version, Ex: V10.13</p> <p>paperSize: paper size, unit: mm, gap interlabel spacing, width label width, height label height or length , only for label printer 。 Ex : {"gap":2,"height":30,"width":40}</p> <p>cutter: cut or not, true cut false noncut</p> <p>supportUyghur: whether or not support Uyghur true support false nonsupport</p> <p>simValidity: SIM validity, only printers with 4G ports have this property.</p> <p>expiryDays: Remaining active days, only printers with 4G ports have this property.</p> <p>official: Official card or not, true Official、false Unofficial</p> <p>simStatus: sim Status, 1to be activated、2activated、4down、6testable、7in stock 、8pre-subscribed、9cancelled account、expired 5, only printers with 4G ports have this property.</p> <p>simStatusDesc: sim status description</p>
------	--------	---

### Request example:

```
{
  "sn": "XPY123456789A",
  "user": "testuser@google.com",
  "timestamp": "1565417654",
  "sign": "82bdcbe2cf6ac4923339b13c2aad1f95ddf0b0a8",
  "debug": "0"
}
```

### Return example:

#### Succeeded:

```
{
  "code": 0,
  "data": {
    "name": "Printer problem test",
    "sn": "XPY123456789A",
    "status": "Offline",
    "type": "00",
    "version": "V10.13",
```

```

        "cutter": false,
        "supportUyghur": false
    },
    "serverExecutedTime": 63
}

```

**Failed:**

```

{
    "code": 1002,
    "msg": "PRINTER_NOT_REGISTER",
    "serverExecutedTime": 54
}

```

## Annex 1: Description of return status code

Return Code	Status	Return to the msg Field Information	Failure Analysis
0	Succeeded	ok	
-1	Request header error	REQUEST_HEADER_ERROR	
-2	Parameter invalid	REQUEST_PARAM_INVALID	The parameter value is a null or the format and size do not meet the requirement.
-3	Parameter signature failed	REQUEST_SIGN_FAILED	Please refer to the sign field description of request common parameter.
-4	User not register	REQUEST_USER_NOT_REGISTER	
1001	The printer SN is not matched with the user.	SN_USER_NOT_MATCH	
1002	Printer not register	PRINTER_NOT_REGISTER	

1003	Printer offline	PRINTER_OFFLINE	
1004	Add order failed	ADD_ORDER_FAILED	
1005	Order not found	ORDER_NOT_FOUND	
1006	Order date format or size invalid	ORDER_DATE_INVALID	
1007	The print content cannot exceed 4096 bytes.	PRINT_CONTENT_MORE_THAN_4096_BYTES	
1008	User failed to modify printer record	PRINTER_RECORD_LOCK_FAIL	
1009	When the user adds a printer, the printer number or name cannot be empty	SN_OR_NAME_EMPTY	Equipment number and name are required and cannot be omitted
1010	Invalid printer device number	PRINTER_NUMBER_INVALID	The current printer is not a cloud printer or the printer number is incorrectly entered. You need to enter the number of a string of English numeric characters at the bottom of the printer or after the words (sn/pid) on the back
1011	The printer already exists. If the printer information cannot be found on the current open platform, please contact the after-sales technical support personnel for verification	PRINTER_EXIST	The printer has been added to the cloud platform. If you need to add it again, you need to delete it on the cloud platform or call the delete (delprinters) interface to delete it
1012	Failed to add printing device. Please try again later or contact after-sales technical support personnel	PRINTER_EXCEPTION	



1013	Trigger idempotency when printing an order	ORDER_IDEMPOTENT	According to the idempotent factor, check whether there is repeated submission of order print content, and the idempotent factor should be globally unique
1014	Idempotent factor too long	ORDER_IDEMPOTENT_LIMITLEN	The effective length of idempotent factor is 50 characters. You can refer to adjusting the length of idempotent factor
1016	LOGO File format error	LOGO_FORMAT_ERROR	(1)Please check that the uploaded logo file conforms to the image format(for example: png, jpg) (2)The uploaded image content needs to be encrypted with Base64 (3)The uploaded image data needs to be removed‘data:image/png;base64,’ 或‘data:image/jpeg;base64,’prefix
1017	LOGO document out of specified range	LOGO_SIZE_LARGE	Logo file size cannot exceed 30KB
1018	The number of logo uploads exceeds the limit	LOGO_UPLOAD_TIMES	The upload of logo file is limited to 3 times per day. If this prompt appears, wait for the next day to upload it. Developers do not need to do other operations
1020	Logo deletion failed	LOGO_DELETE_NO_EXIST	The logo file has been deleted or has not been uploaded. If you confirm that it has been uploaded but still prompt the error, you need to contact the technical personnel for troubleshooting
1021	Logo upload mode error	LABEL_MODE_ERROR	For uploading the logo file of the label printer, it needs to be uploaded in the label printing mode. For the small ticket mode, the operation is invalid. You can check the current printing mode of the label machine by printing the self

			inspection page
1022	This device is a customized device, and you are currently not authorized to use it		The printer has assigned an account number and cannot be added. Please contact customer service if you have any questions