Goodcom WebHook API

V 1.0

1. WebHook communication diagram

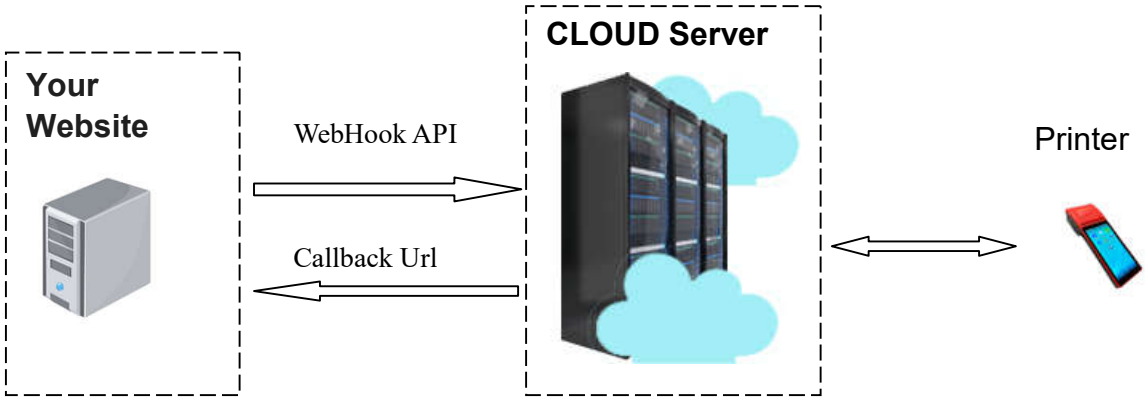

Figure 1

Your website submits the order to the CLOUD server through the WebHook API, and the CLOUD server sends the order to the printer. After the user confirms the order, the printer sends the confirmation information to the CLOUD server, and the CLOUD server sends it to the your website through the Callback Url.

2. WebHook API

The website submits the order through the webhook api. The order content format is json. The json must contain the API key. You can bring the Secret key as needed.

The Secret key is not used for JSON parameter passing, but for calculating the sign parameter of JSON

2.1 API key and Secret Key

You can view the api key and sercet key through the following steps:

a. Login: https://gcanyapps.com/

Enter account password and verification code.

b. Click Regions on the left

Select Regions and find the created region in the Region List, then click "Edit".

Note: If the region has not been created yet, click "Add Region" to create it, and click "Click to select App Accounts" to add the account to the region. Multiple accounts can be added to one region.

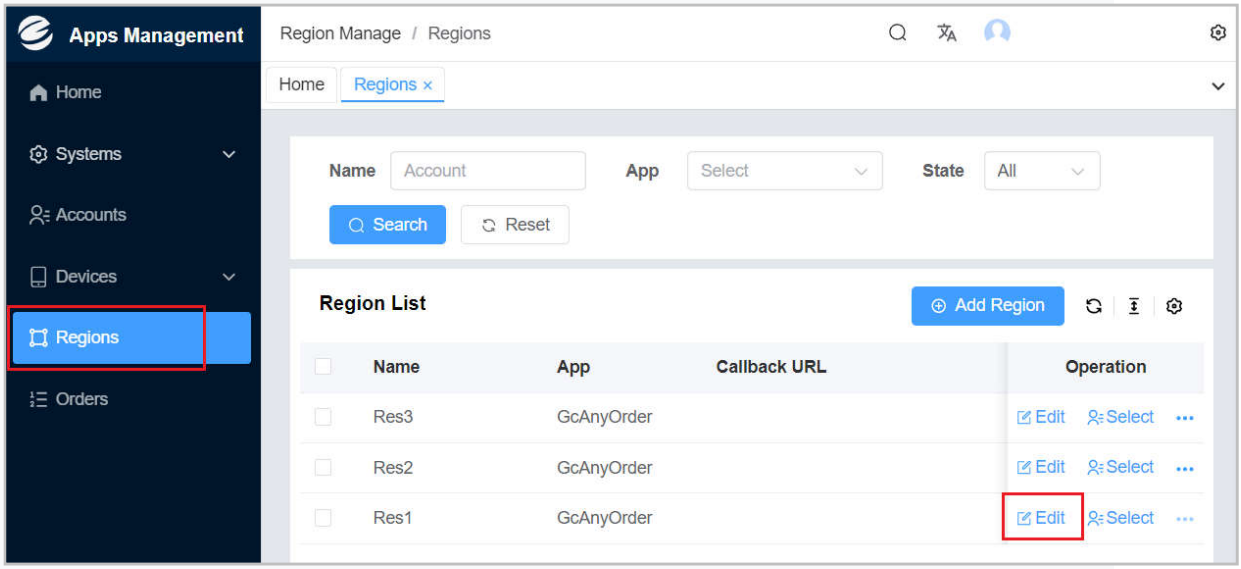

Figure 2

c. You can see the API key in the pop-up interface. If you need the Secret key, you need to click the "Dynamic Validation" switch to turn it on.

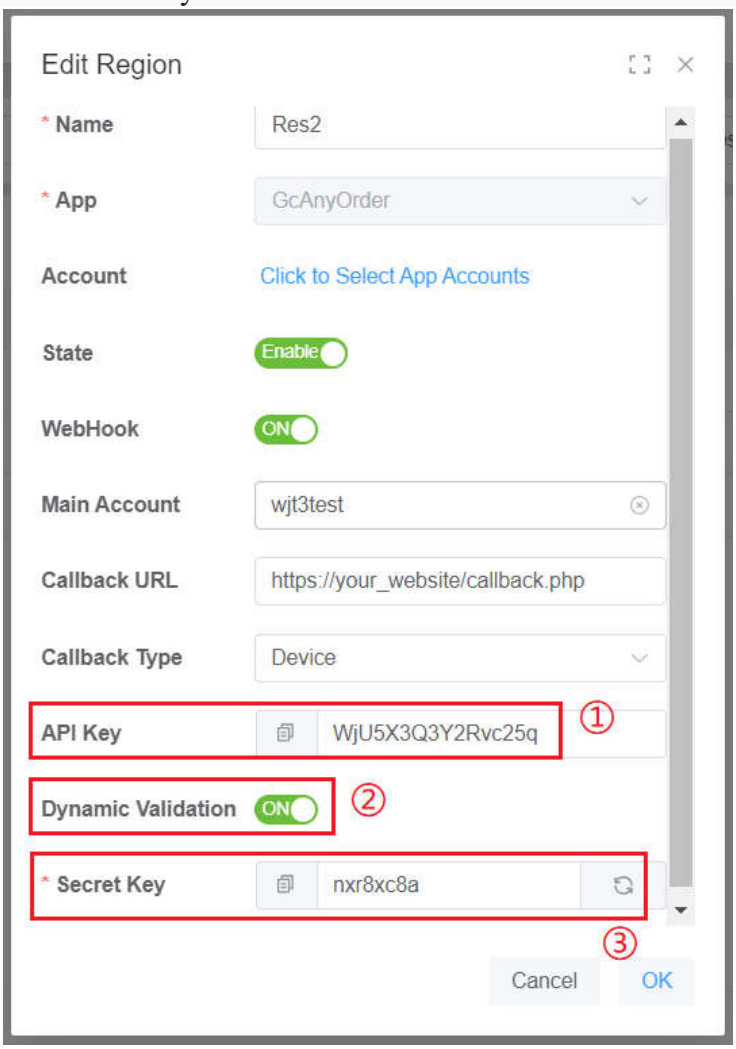

Figure 3

2.2 Order Submit URL

Name	Details
URL	https://www.GcAnyApps.com/api/submitorder
Method	POST
Content Format	JSON，Reference **3.1** for the definition of relevant key values

Submit your order details according to API guideline to this url using post method including API key, API sign etc. from your website to our cloud server.

2.3 Order Cancel URL

Name	Details
URL	https://www.GcAnyApps.com/api/cancelorder
Method	POST
Content Format	JSON，Reference **3.1** for the definition of relevant key values

Cancel order according to API guideline to this url using post method including API key, API sign etc. from your website to our cloud server

3. JSON Content Definition

Data structure:

```
{
  "api_key": "string",
  "cur_time": "string",
  "sign": "string",
  "callback_url": "string",
  "order_id": "string",
  "order_json": {
    "CreateTime":"string",
    "DeliveryType": "string",
    "OrderId":"string",
    "Order": [
      {
        "o_title":"string",
        "o_num": "string",
        "o_name": "string",
        "o_amt": "string",
```

```
                "OrderOption": [
                    {
                      " o_oName":"string",
                      " o_oAmt": "string"
                    }
                ]
              }
            ],
            "SubTotal":"string",
            "DiscountAmt": "string",
            "DiscountTaxes": "string",
            "DeliveryCharges": "string",
            "DeliveryTaxes": "string",
            "ExtraFeeAmt": "string",
            "TotalTaxes": "string",
            "Total": "string",
            "PaymentStatus": "string",
            "CustName": "string",
            "CustAddr": "string",
            "CustTown": "string",
            "CustPhone": "string",
            "RequestTime": "string",
            "PaidType": "string",
            "Comment": "string"
        },
        "order_txt": "string"
}
```


3.1 Fields for submitting orders

Field	Importance	Format	Details
api_key	Mandatory	String	The API key automatically generated by the cloud server when creating a region (Refer to ① in Figure 3)
cur_time	Mandatory	String	Current UNIX timestamp, accurate to milliseconds. As the unique identifier of the order. If the order is resent (for example, resending

Field	Importance	Format	Details
			after failure), this value remains the same. If Dynamic Validation is turned on (refer to ② in Figure 3), this content will be used as part of the encrypted password
sign	Optional	String	The string obtained by performing Md5 operation on the parameters api_key+secret key+cur_time (the + sign indicates the connector). **md5**(api_key+ secret key+cur_time) Among them, the secret key comes from ③ in Figure 3. This secret key is not transmitted on the network and is only used for encryption verification. (refer to "5.1 JSON order example")
callback_url	Optional	String	The URL for the web server to receive responses from the printer after accepting/rejecting orders. When the printer accepts/rejects the order and the status needs to be replied to the web server, it can be returned through this url. If this is not set, it is determined by the callback url of the regional account. If no url is set, no information will be returned to the web server
order_id	Mandatory	String	Order id for the order to be tracked (must be unique)
order_json	Optional	Object	Order content, the format is json content **Refer to the definition in 3.3**
order_txt	Optional	String	Order content, the format is string content

Note: order_json and order_txt are optional, if this is the content of canceling the order(Order cancel API), both of these are not needed, but if it is the content of

submitting the order(Order submit API), at least one is required. If both are present, order_json will be the main one;

3.2 Response

The response format uses json.

3.2.1 Response when received successfully

Field	Format	Details
status	Integer	Status code , 200
msg	String	ok

{
"status": 200,
"msg": "ok"
}

3.2.2 Response when reception fails

Field	Format	Details
status	Integer	Status code, 400
msg	String	Error message

{
"status": 400,
"msg": " Error message"
}

3.3 The definition of ‘order_json’ field

Field	Importance	Format	Details
CreateTime	Optional	String	Order creation time
DeliveryType	Optional	String	Order delivery type:

Field	Importance	Format	Details
			It can be Delivery, Take away, Pickup, Dine-in, etc.
OrderId	Mandatory	String	Order id for the order to be tracked (must be unique) Must be consistent with **'order_id'** above
Order ->o_title (String) ->o_num (String) ->o_name (String) ->o_amt (String) ->OrderOption (Array) --> o_oName(String) --> o_oAmt(String)	Mandatory at least one	Array	Order product items, including title, quantity, name, price and options, etc. Each subfield is optional
SubTotal	Optional	String	Subtotal, If there is no such content, it can be empty or without this field
DiscountAmt	Optional	String	Discounted fees, it can be empty
DiscountTaxes	Optional	String	Discounted fee tax, it can be empty
DeliveryCharges	Optional	String	Shipping fee, it can be empty
DeliveryTaxes	Optional	String	Shipping tax, it can be empty
ExtraFeeAmt	Optional	String	Additional fee, it can be empty
TotalTaxes	Optional	String	Total tax, it can be empty
Total	Optional	String	Total fee
PaymentStatus	Optional	String	Payment status, which can be " Order Paid" or "Order No Paid". It can also be empty if not required.

Field	Importance	Format	Details
CustName	Optional	String	Customer's name
CustAddr	Optional	String	Customer's address
CustTown	Optional	String	Customer's town
CustPhone	Optional	String	Customer's phone number
RequestTime	Optional	String	The time the customer requests food delivery/pickup The format can be as follows: hh:mm yyyy-MM-dd(e.g 12:10 2024-04-21) hh:mm MM-dd-yyyy (e.g 12:10 04-21-2024) hh:mm dd-MM-yyyy(e.g 12:10 21-04-2024)
PaidType	Optional	String	Payment type, such as Cash on delivery , PayPal ,Stripe, etc.
Comment	Optional	String	Customer's remarks

4. Callback

After the user confirms the order, the printer will notify the website of the confirmation status, and the notification path is the callback URL. The callback URL can be sent when the website submits the order, or pre-set on the cloud server.

The callback format is as follows:

For accepted order, the reply is that:

(1) GET callback_url?a=2&o=10004&ak=Accepted&m=OK&dt=21:51 &dd=210424&u=testuser &p=test&cur=2151342104 HTTP/1.1

(2) GET callback_url?a=2&o=323&ak=Accepted&m=10_Minutes&dt=21:51 &dd=210424 &u=testuser&p=test&cur=2151342104 HTTP/1.1

For reject (declined) order, the reply is that:

GET callback_url?a=2&o=323&ak=Rejected&m=FOOD_UNAVAILABLE &dt=02:36&dd=210424&u=testuser&p=test&cur=2151342104 HTTP/1.1

Parameter Description:

Parameter	Details
“a”	Res id which had been set in the device
“o”	Order id which come from the order.
“ak”	The operation of accepting/rejecting order. When the order is accepted, ak=Accepted. When the order is rejected, ak=Rejected
“m”	The description of accepting/Rejecting order. If accept the order and the accept reply type is time, m=OK, when the accept reply type is item, m=select item(example: m=10_Minutes). If reject order, “m” is the reject reason that the operator selected when rejecting the order.
“dt”	The time that the operator input when accepting order.
“dd”	The date that the operator input when accepting order. The format is ”ddmmyy”, example: dd=210424,it means the date is 2024-4-21.
“u”	The user name for login your website
“p”	The password for login your website.
“cur”	Current timestamp, if the website does not require this parameter, it can be ignored

5. Example

5.1 JSON order example

If API key = WjU5X3Q3Y2Rvc25q and Secret Key = **nxr8xc8a**,
then sign=MD5(API key+Secret key+cur_time)
=MD5(WjU5X3Q3Y2Rvc25qnxr8xc8a1713748630)
= 60a7d2d9698eb62c0b928467cde123db
The json order format is as follows:


```
{
  "api_key": "WjU5X3Q3Y2Rvc25q",
  "cur_time": "1713748630",
  "sign": "60a7d2d9698eb62c0b928467cde123db",
```

```
"callback_url": "http://goodcom.cn/order/order_callback.php",
"order_id": "10006",
"order_json": {
        "Create_time":"15:00 16-08-2022",
        "DeliveryType":"Delivery",
        "OrderId":"10006",
        "Order":[
                {
                        "o_title":"Title",
                        "o_num":"1",
                        "o_name":"Chicken",
                        "o_amt":"5.00",
                        "OrderOption":[
                                {
                                        "o_oName":"option1",
                                        "o_oAmt":"1.00"
                                },
                              {
                                        "o_oName":"option2",
                                        "o_oAmt":"2.00"
                                }
                        ]
                }
        ],
        "SubTotal":"0.00",
        "DiscountAmt":"0.00",
        "DiscountTaxes":"",
        "DeliveryCharges":"0.00",
        "DeliveryTaxes":"",
        "ExtraFeeAmt":"",
        "TotalTaxes":"",
        "Total":"29.10",
        "PaymentStatus":"Order Paid",
        "CustName":"Tom",
        "CustAddr":"Address of the Customer",
        "CustTown":"",
        "CustPhone":"00861234567890",
        "RequestTime":"15:30 16-08-2022",
        "PaidType":"Cash",
        "Comment":"comment",

},
"order_txt": ""
}
```

5.2 Example of PHP code generating json format orders

```php
<?php
//////////////////////////////////////
    $api_key='WjU5X3Q3Y2Rvc25q';
    $secret_key='nxr8xc8a';
    $submit_order_url='https://www.GcAnyApps.com/api/submitorder';
    $cancel_order_url='https://www.GcAnyApps.com/api/cancelorder';
    $callback_url='https://your_website/order/order_callback.php';
//////////////////////////////////////
function
generate_json_order($order_id,$api_key,$secret_key,$callback_url)
{
    $timeStamp = time();
    $delivery_type = 'Delivery';
////////////////////////////////
    $arr_data = array();
    $arr_data['api_key'] = $api_key;
    $arr_data['cur_time'] = $timeStamp;
    $arr_data['sign'] = md5($api_key . $secret_key . $timeStamp);
    $arr_data['order_id'] = $order_id;
    $arr_data['callback_url'] = $callback_url;

    $arr_order = array();
    $arr_order['Create_time'] = date('H:i d-m-Y', $timeStamp);
    $arr_order['DeliveryType'] = $delivery_type;
    $arr_order['OrderId'] = $order_id;

    $array_product = array();
    $array_products = array();
    for ($jj = 0; $jj < 2; $jj++) {
        $array_product['o_title'] = 'Title_' . $jj;
        $array_product['o_num'] = '3';
        $array_product['o_name'] = 'Chicken_' . $jj;
        $array_product['o_amt'] = '2.00';
        $array_options = array();
        for ($kk = 0; $kk < 2; $kk++) {
            $array_option = array();
            $array_option['o_oName'] = '+option_' . $jj;
            $array_option['o_oAmt'] = '0.00';
            $array_options[$kk] = $array_option;
        }
        $array_product['OrderOption'] = $array_options;
```

```
        $array_products[$jj] = $array_product;
    }
    $arr_order['Order'] = $array_products;
    $arr_order['DiscountAmt'] = '1.00';
    $arr_order['DiscountTaxes'] = '0.00';
    $arr_order['SubTotal'] = '30.00';
    $arr_order['DeliveryCharges'] = '0.00';
    $arr_order['DeliveryTaxes'] = '';
    $arr_order['ExtraFeeAmt'] = '';
    $arr_order['TotalTaxes'] = '0.00';
    $arr_order['Total'] = '60.00';
    $arr_order['PaymentStatus'] = 'Order Paid';
    $arr_order['CustName'] = 'Tom';
    $arr_order['CustAddr'] = 'Address of the Customer';
    $arr_order['CustTown'] = 'xiamen';
    $arr_order['CustPhone'] = '00861234567890';
    $arr_order['RequestTime'] = date('H:i d-m-y');
    $arr_order['PaidType'] = 'Cash';
    $arr_data['order_json'] = $arr_order;
    return json_encode($arr_data);
}
$order_id = '10006';
$json_data=generate_json_order($order_id,$api_key,$secret_key,$callba
ck_url);
```

5.3 Example of PHP Code submitting json order to cloud server

On the basis of 5.2, the content in json format can be sent to the webhook server through the following code

```
function api_send_to_server($url, $json_data) {

    set_time_limit(60);

    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $url);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $json_data);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_TIMEOUT, 30);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, FALSE);
    curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, 1);
```

```
    for ($i = 0; $i < 3; $i++) {
        $response = curl_exec($curl);
        if ($response !== false) {
            break;
        }
        sleep(5);
    }
    curl_close($curl);

    $ret=json_decode($response);

    if(empty($ret)){
        return 'error';
    }
    if($ret->status == 200){
        return 'ok';
    }

    return 'error';
}
// $submit_order_url and $json_data are defined in the 5.2 example
$ret = api_send_to_server($submit_order_url,$json_data);
if($ret=='ok'){
     //Sent successfully, do some processing

}
else{
    //If the delivery fails, handle accordingly, such as saving the order
number and resending regularly.
   ……
}
```

5.4 PHP Code example of order cancellation

```
    //$api_key,$secret_key are defined in 5.2 example
    $order_id = '10006';
    $timeStamp = time();
    $arr_data = array();
    $arr_data['api_key'] = $api_key;
    $arr_data['cur_time'] = $timeStamp;
    $arr_data['sign'] = md5($api_key . $secret_key . $timeStamp);
    $arr_data['order_id'] = $order_id;
```

```
    $json_data = json_encode($arr_data);
    // api_send_to_server is defined in the 5.3 example
    $ret = api_send_to_server($cancel_order_url,$json_data)
   if($ret=='ok'){
     //Sent successfully, do some processing

    }
     else{
         //If the delivery fails, handle accordingly, such as saving the
     order number and resending regularly.
     ……
   }
```

5.5 Example of how a website receives callback(PHP example)

When the website sends an order in json format to the cloud server, the json contains the callback url (or it does not, but the callback url is set on the cloud server), for example: "callback_url": "http://goodcom.cn/order/order_callback .php",

When the printer receives the order and the user confirms the order, the confirmation information will be sent to the website through the callback url(for example: order_callback.php). At this time, the website can read the information and save it to the corresponding order or modify the order status. The following demonstrates how order_callback.php reads the information returned by the printer:

```
$resId="";
$orderId="";
$status="";
$reason="";
$time="";
$date="";
$user="";
$pwd="";
$bat="";
$cur="";
if(isset($_GET['a'])){
    $resId=$_GET['a'];
}
if(isset($_GET['o'])){
    $orderId=$_GET['o'];
}
if(isset($_GET['ak'])){
    $status=$_GET['ak'];
```

```
}
if(isset($_GET['m'])){
    $reason=str_replace("_"," ",$_GET['m']);
}
if(isset($_GET['dt'])){
    $time=$_GET['dt'];
}
if(isset($_GET['dd'])){
    $date=$_GET['dd'];
}
if(isset($_GET['u'])){
    $user=$_GET['u'];
}
if(isset($_GET['p'])){
    $pwd=$_GET['p'];
}
if(isset($_GET['cur'])){
    $cur=$_GET['cur'];
}

//Information collection, can be ignored
$tips="Order callback\rres Id:".$resId."\r";
$tips.="Order id:".$orderId."\r";
$tips.="Status:".$status."\r";
$tips.="Reason:".$reason."\r";
$tips.="Time:".$time."\r";
$tips.="Date(ddMMYY):".$date."\r";
$tips.="User name:".$user."\r";
$tips.="Password:".$pwd."\r";
$tips.="Timestamp:".$cur."\r";


//Verify account and password
// if($user!='test')||($pwd!='testuser')){
// echo 'error';
//  exit;
//}

if($status=='Accepted'){
    //order accepted
    if($reason=='OK'){
        $accept_time=$time.' '.substr($date, 0, 2) . "-" . substr($date,
2, 2) . "-20" . substr($date, 4, 2);
        //Modify order status and accept time,the order id is $orderId
```

```
    }
    else{
        //Modify order status and accept content,the order id is
$orderId,the accept content is $reason

    }
}
else if($status=='Rejected'){
    //order rejected
    //Change order status to rejected,the rejected reason is $reason,the
order id is $orderId

}
else{
    // unknown status

}
```