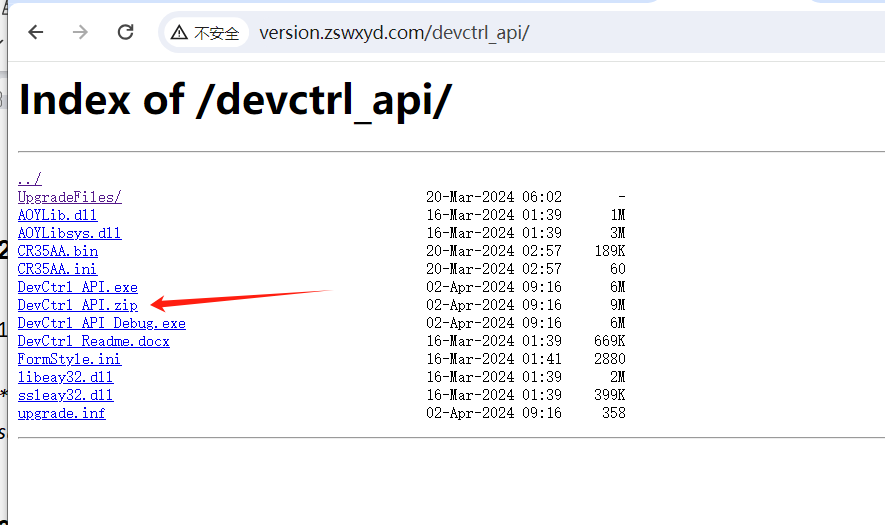
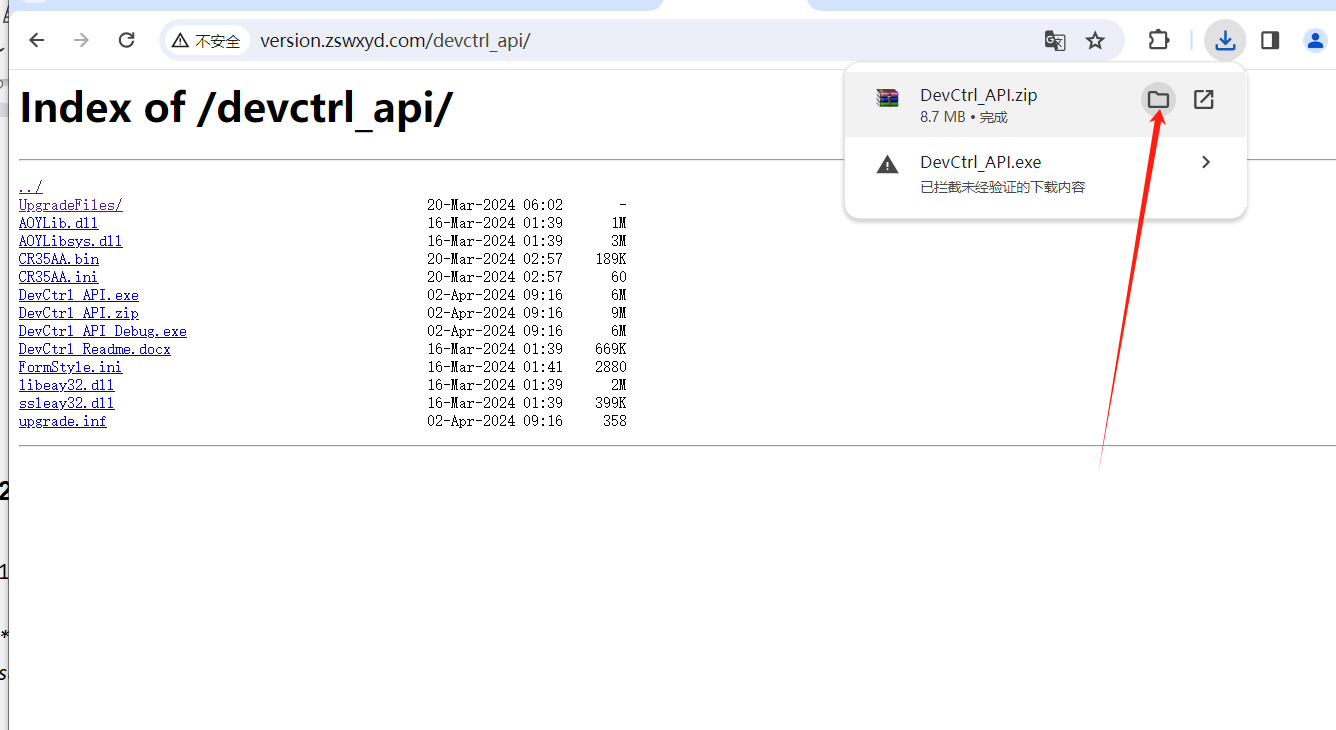
**DevCtrl\_API Manual**

###### Download DevCtrl\_Api.zip and unzip to local folder ( as D:\DevCtrl\_Api ).

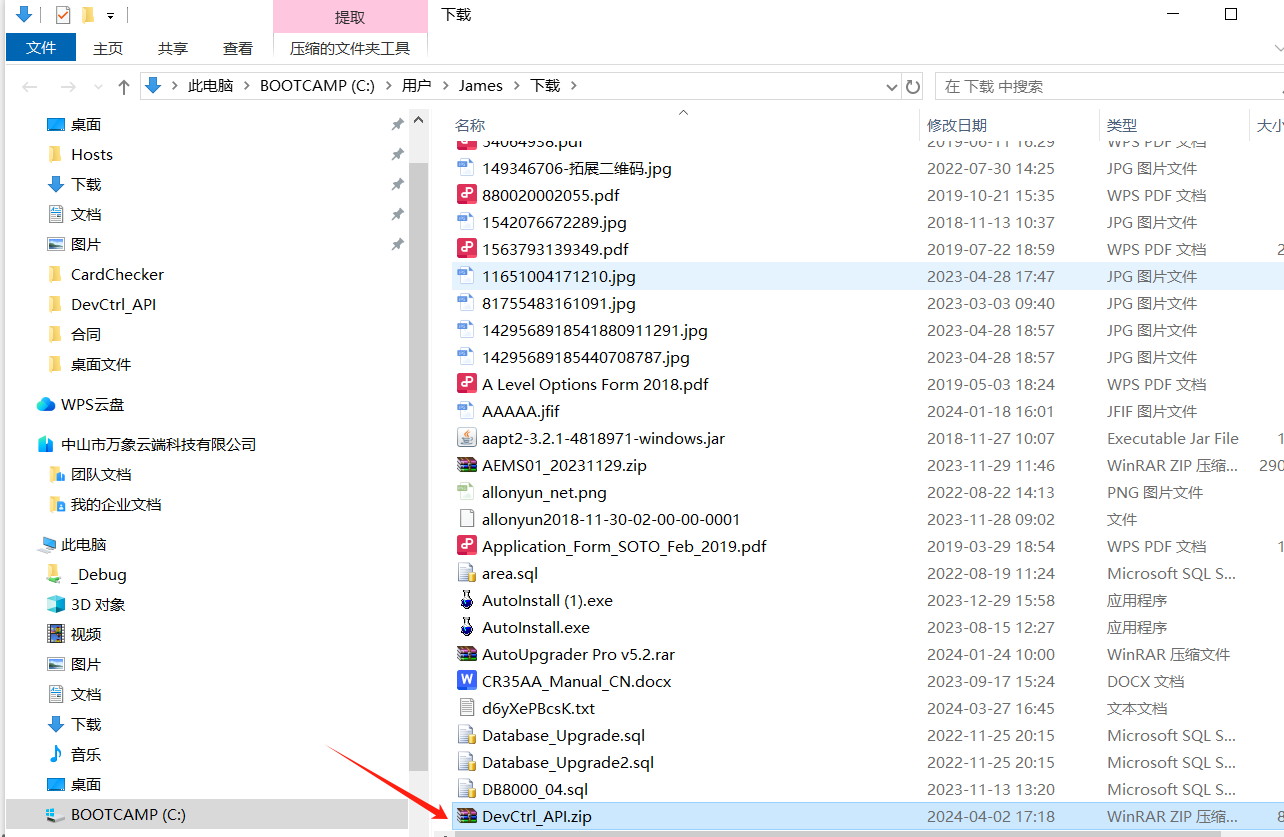
<http://version.zswxyd.com/devctrl_api/DevCtrl_API.zip>



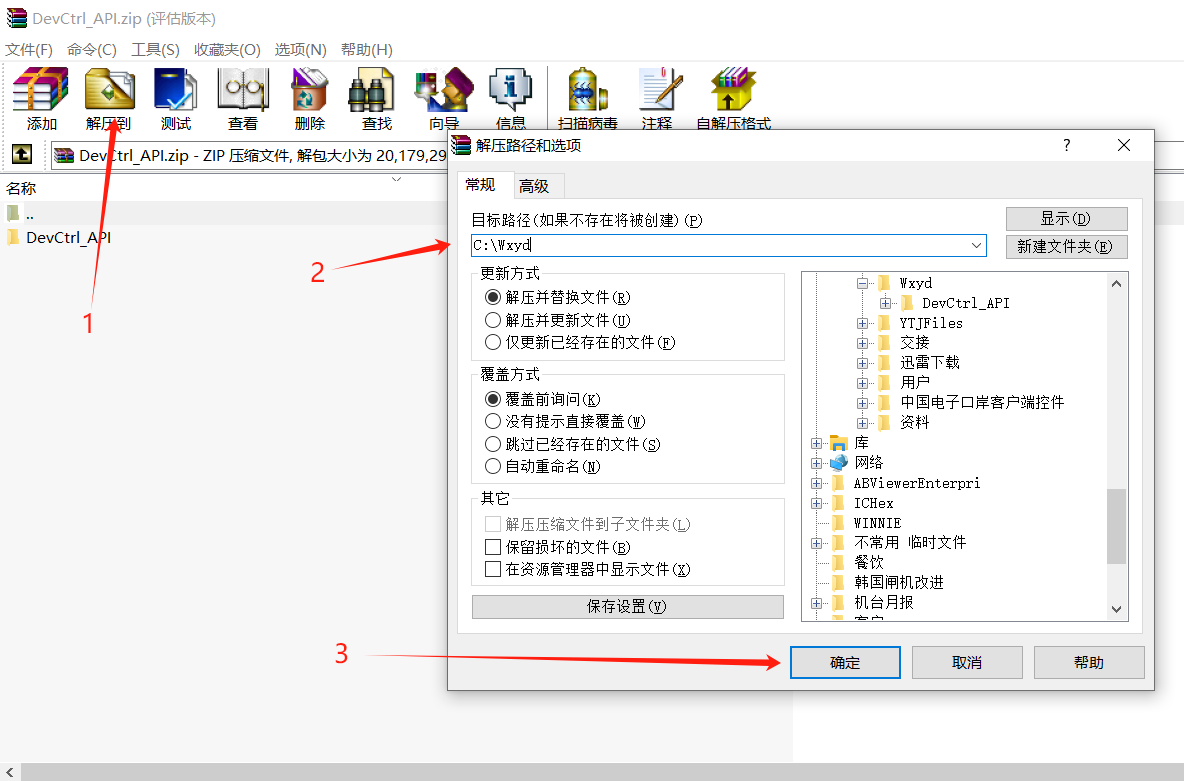
< open the website, download the DevCtrl\_API.zip file >



< Open the folder of download path >



< Find the download file at download folder >



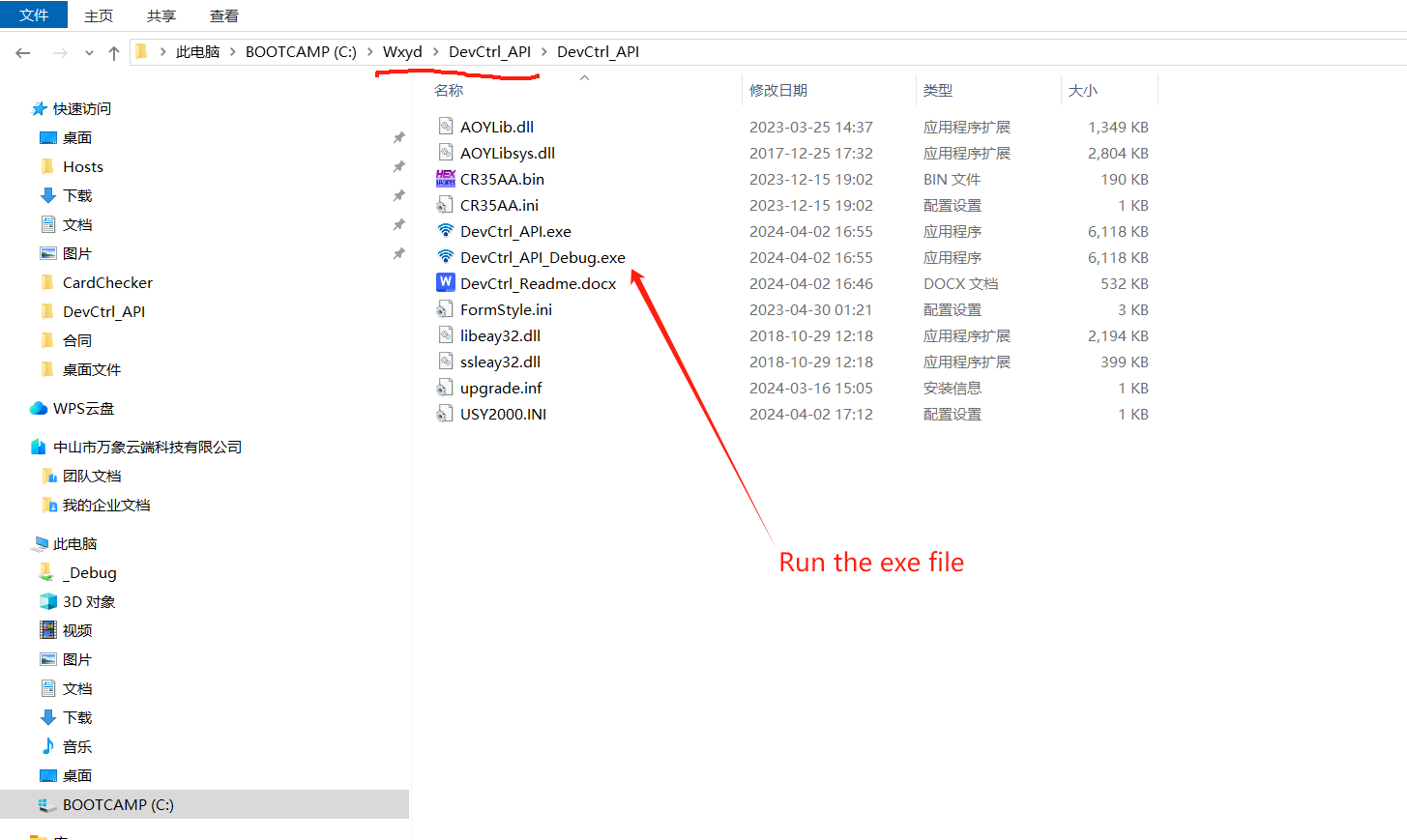
< unzip the Devctrl\_API.zip to local folder, C:\WXYD >

###### Run DevCtrl\_Api.exe and login.

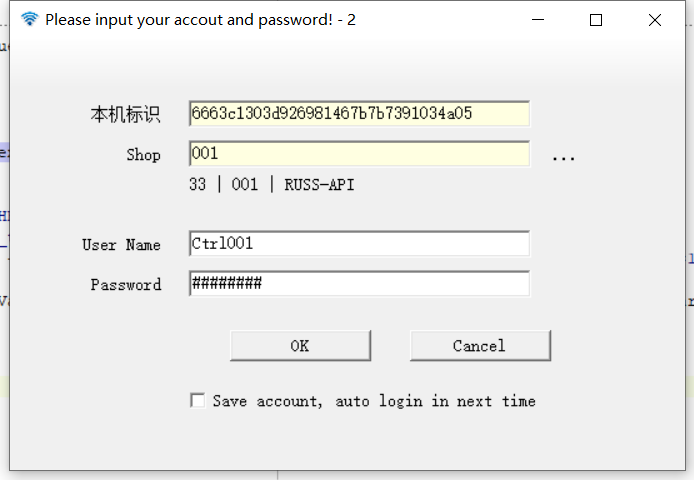
Please obtain your username and password from us, using email  [<james@zswxyd.com](mailto:<james@zswxyd.com)> or mobile (+86 13928127700).

Please write down the IP Address of the computer，example 172.16.0.5

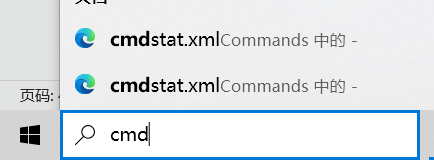
*\*\*\* You can connect to a local computer instead of a cloud computer, and we will explain this in detail in our subsequent steps*

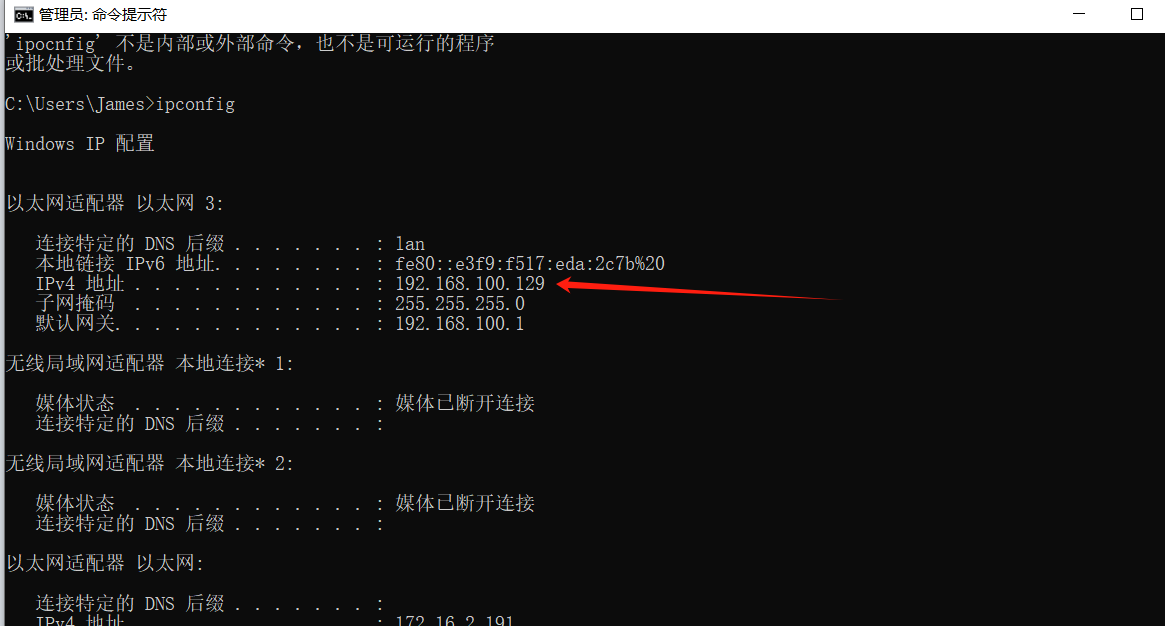


< run the execute file “DevCtrl\_API\_Debug.exe” >



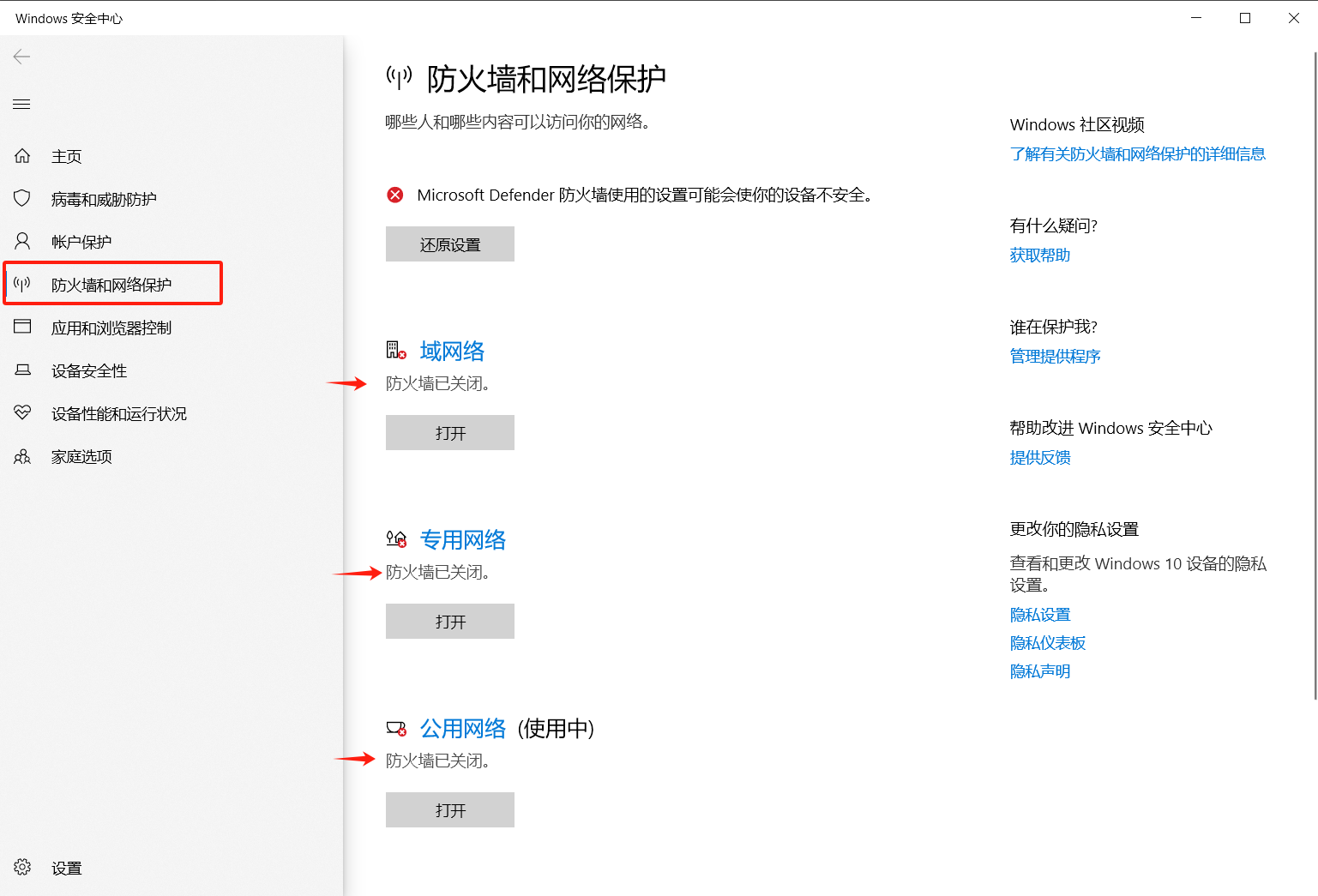
< Login with User Name and password. No user? Please call me >



< run cmd.exe open command window > 

< run cmd.exe open command window, and run ipconfig to get the your PC IP Address >

###### Turn off the Firewall of Server whick running “DevCtrl\_Api.exe”. Port=1883



< the card reader need TCP-Port 1883 connect to DevCtrl\_Api Program >

###### Setting your WIFI router. Need two SSID:

* 1. WIFI SSID Setting

SSID: DVESPUB, Password: DS123456

SSID: DVESWIFI, Password: 6220asWF

*\*\*\* You can modify your SSID and password, and we will explain this in detail in our subsequent steps*

* 1. Network

Router LAN IP Address: 172.16.0.1

DHCP: 172.16.0.50 - 172.16.8.255

Mask: 255.255.0.0

* 1. Sever IP Address: 172.16.0.5

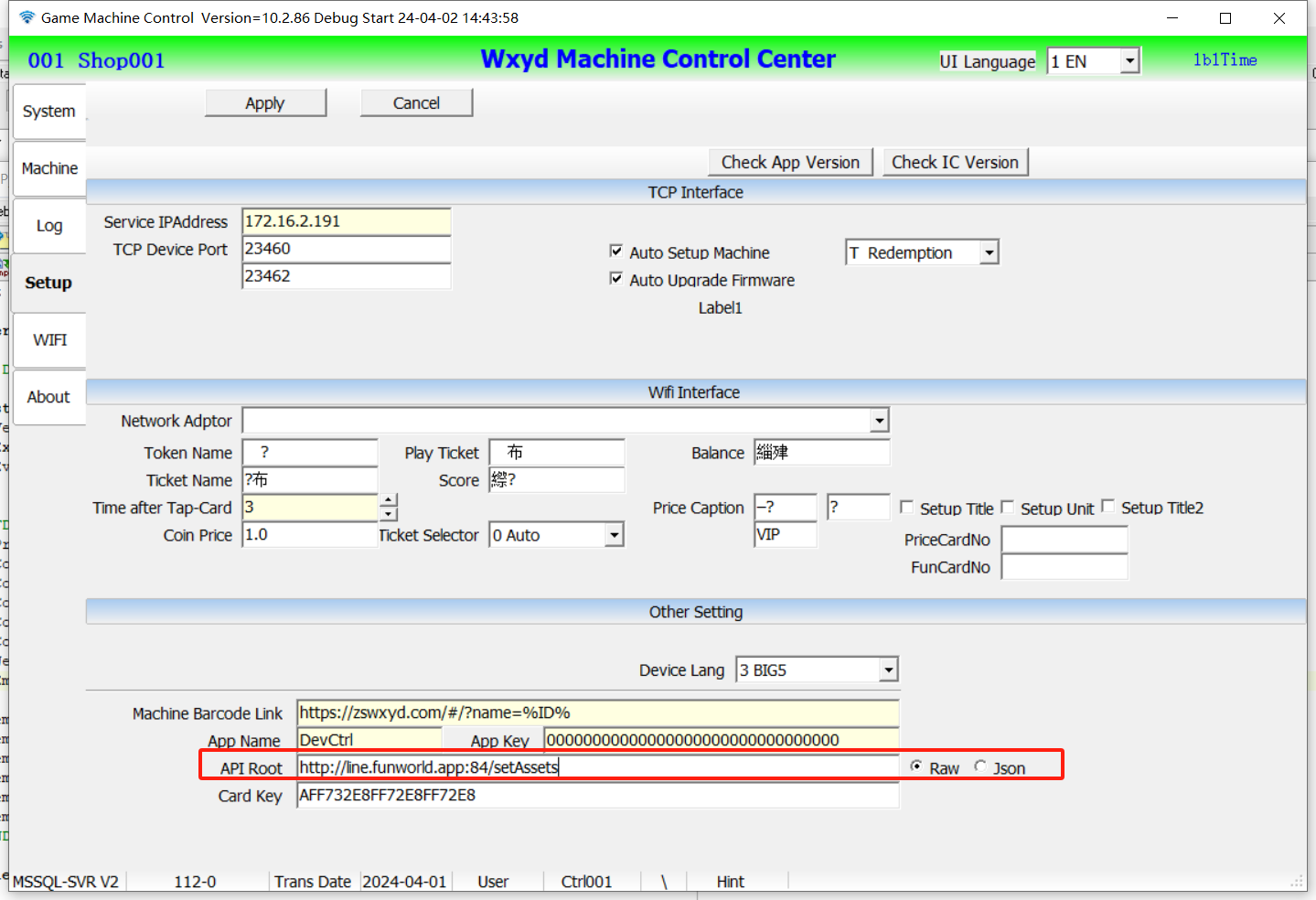
###### Start to power on the card reader.

*\*\*\* The default settings for the card reader are*

*“SSID=DVESWIFI, Password=6220asWF, Server=172.16.0.5”*

###### API Test

* 1. Prepare your http server, not https server.
  2. Tell Devctrl\_Api program your http server, port (if have), and function, and data format ( Recommend using JSON format).



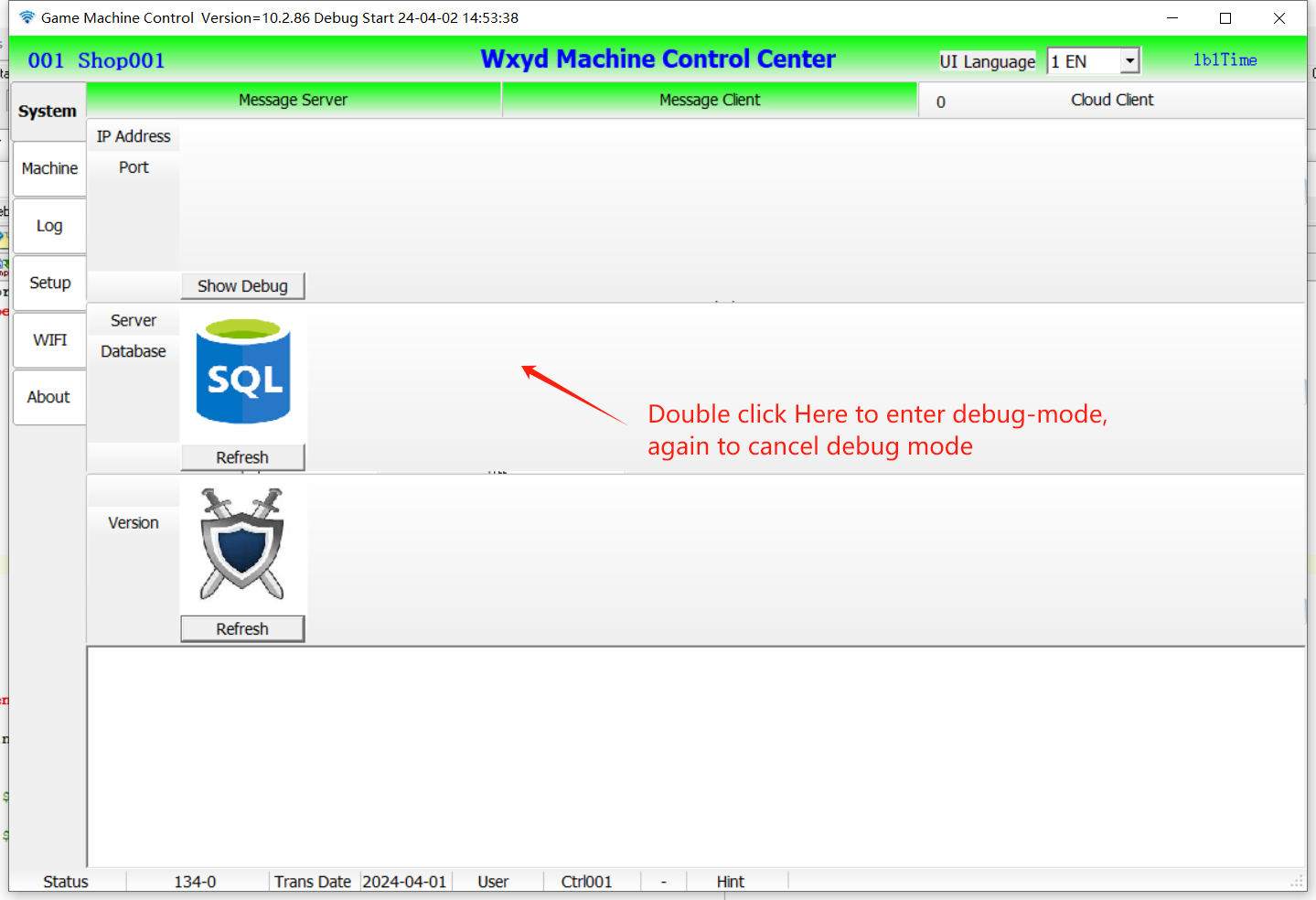
< Double click to make changes, apply to save >

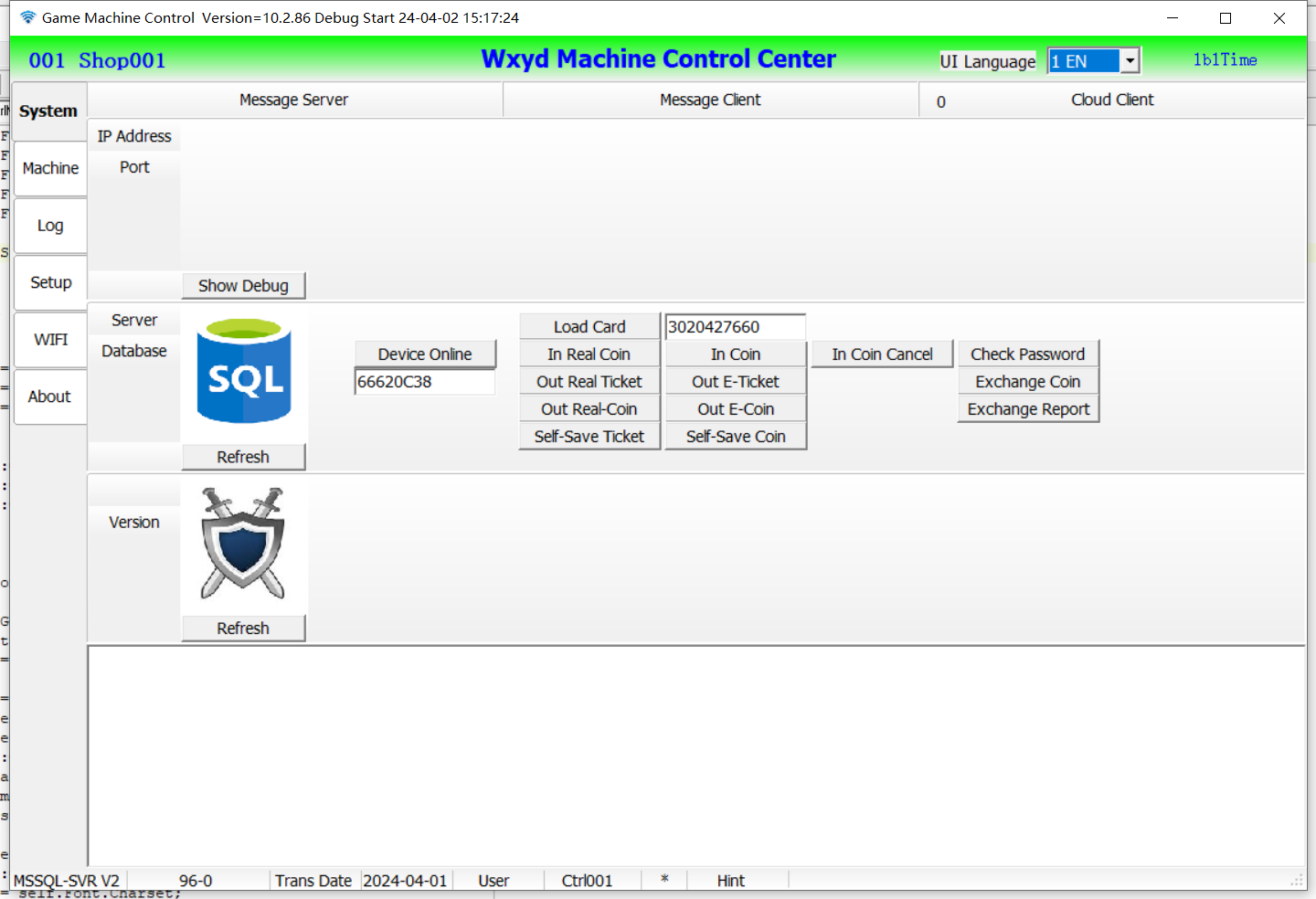
Data Format:

Raw: card\_no=1234567890&mach\_sn=66621234&tickets=0&tokens=0&xid=4256&event=load\_card

Json: {“card\_no”:”1234567890”,”mach\_sn”:”66621234”,”xid”:4256,”event”:”load\_card”}

* 1. Test your API by hand. You can just test your API with no card-reader.





( setup the reader\_sn and card\_no, click the function button )

* 1. Test API API by card reader.

###### API description

Require Json:

{

“card\_no”:”1234567890”, // card no

”mach\_sn”:”66621234”, // card reader sn

”tickets”:0, // tickets to add, negative for sub

”tokens”:0, // tokens to add or sub, negative for sub

”xid”:4256, // trans-id, check

”event”:”load\_card”, // sub function

“app\_id”:””, // \*\*\* app\_id

“time”:””, // \*\*\* timestamp

“sign”:”” // \*\*\* md5\_sign

}

Reply Json:

{

“card\_no”:”1234567890”, // card no

”mach\_sn”:”66621234”, // card reader sn

"eToken":5443, // card balance of coin

"eTicket":456, // card balance of ticket

"message":"Happy Birthday", // message show on reader screen

"status":"0" // result: =0 success, <>0 fail

“time”:””, // \*\*\* timestamp

“sign”:”” // \*\*\* md5

}

\*\*\*Status: =0 success, <>0 fail

\*\*\*message: show on card reader screen, tell customer why fail or success welcome.

\*\*\*app\_id, timestamp, sign no need

|  |  |  |  |
| --- | --- | --- | --- |
| API | Require Json | Reply Json | Remark |
| for\_test | {”event”:”for\_test”} | Same as require | Just a test |
| dev\_online | {  ”mach\_sn”:”66621234”,  ”event”:”dev\_online”  } | {  ”mach\_name”:”name”,  ”type”:”0”,  “price1”:XX.X,  “price2”:YY.Y,  “signal1”:XX,  “signal2”:YY,  "mach\_name":"mach\_name",  "message":"message on reader",  "status":"0" //success  } | Heartbeat  Type  =0 Game  =1 Kiosk  =2 Gate  =4 Sale  =5 Time |
| load\_card | {  “card\_no”:”1234567890”,  ”mach\_sn”:”66621234”,  ”event”:”load\_card”  } | {  “card\_no”:”1234567890”,  "eToken":1000, // balance1  "eTicket":2000, // balance2  "message":"message on reader",  "status":"0" //success  } | Member balance |
| use\_coin | {  “card\_no”:”1234567890”,  ”mach\_sn”:”66621234”,  ”tokens”:-100,  ”event”:”use\_coin”  } | {  “card\_no”:”1234567890”,  "eToken":900, // new balance1  "eTicket":2000, // balance 2  "message":"message on reader",  "status":"0"  } | Sub coin  from card  by game |
| out\_ticket | {  “card\_no”:”1234567890”,  ”mach\_sn”:”66621234”,  ”tickets”:200,  ”event”:”out\_ticket”  } | {  “card\_no”:”1234567890”,  "eToken":900, // balance1  "eTicket":2200, // new balance2  "message":"message on reader",  "status":"0"  } | Add ticket  to card  by game |
| out\_coin | {  “card\_no”:”1234567890”,  ”mach\_sn”:”66621234”,  ”tokens”:10,  ”event”:”out\_coin”  } | {  “card\_no”:”1234567890”,  "eToken":910, // new balance1  "eTicket":2200, // balance2  "message":"message on reader",  "status":"0"  } | Add coin  to card  by game |
| save\_ticket | {  “card\_no”:”1234567890”,  ”mach\_sn”:”66621234”,  ”tickets”:500,  ”event”:”save\_ticket”  } | {  “card\_no”:”1234567890”,  "eToken":910, // balance1  "eTicket":2700, // new balance2  "message":"message on reader",  "status":"0"  } | Add ticket  to card  by kiosk |
| save\_coin | {  “card\_no”:”1234567890”,  ”mach\_sn”:”66621234”,  ”tokens”:50,  ”event”:”save\_coin”  } | {  “card\_no”:”1234567890”,  "eToken":960, // new balance1  "eTicket":2700, // balance2  "message":"message on reader",  "status":"0"  } | Add coin  to card  by kiosk |
| real\_in\_coin | {  ”mach\_sn”:”66621234”,  ”tokens”:50,  ”event”:”real\_in\_coin”  } | {  ”mach\_sn”:”66621234”,  "message":"message on reader",  "status":"0"  } | In real coin  from game |
| real\_out\_ticket | {  ”mach\_sn”:”66621234”,  ”tickets”:200,  ”event”:”real\_out\_ticket”  } | {  ”mach\_sn”:”66621234”,  "message":"message on reader",  "status":"0"  } | Out real ticket  by game |
| real\_out\_coin | {  ”mach\_sn”:”66621234”,  ”tokens”:200,  ”event”:”real\_out\_coin”  } | {  ”mach\_sn”:”66621234”,  "message":"message on reader",  "status":"0"  } | Out real coin by game |
| real\_out\_gift | {  ”mach\_sn”:”66621234”,  ”gifts”:1,  ”event”:”real\_out\_gift”  } | {  ”mach\_sn”:”66621234”,  "message":"message on reader",  "status":"0"  } | Out gift  by game |
| check\_password | {  “card\_no”:”1234567890”,  ”mach\_sn”:”66621234”,  ”password”:”123456”,  ”event”:”check\_password”  } | {  “card\_no”:”1234567890”,  "eToken":910, // balance1  "eTicket":2200, // balance2  "message":"message on reader",  "status":"0"  } | Check  card password  by Kiosk |
| exchange\_coin | {  “card\_no”:”1234567890”,  ”mach\_sn”:”66621234”,  ”tokens”:-200,  ”event”:”exchange\_coin”  } | {  “card\_no”:”1234567890”,  "eToken":710, // new balance1  "eTicket":2200, // balance 2  "message":"message on reader",  "status":"0"  } | Dispense coin  from card  by kiosk |
| exchange\_coin\_rpt | {  “card\_no”:”1234567890”,  ”mach\_sn”:”66621234”,  ”tokens”:0,  ”event”:”exchange\_coin\_rpt”  }  tokens=0 dispense finish  tokens>0 Insufficient, refund to card | {  “card\_no”:”1234567890”,  "eToken":710, // new balance1  "eTicket":2200, // balance 2  "message":"message on reader",  "status":"0"  } | Result of dispense coin  by kiosk |

###### Check your card reader status

1. Starting,

< Lang, Direction, Network, Logo, Firmware version, SN >

1. Connecting to Network ( Work-SSID, Public-SSID)
2. Try to connecting Work-SSID
3. Success to connected to Work-SSID, skip to step-3.
4. No Work-SSID network, find new network config < new Work-SSID> from Public-SSID.
5. Reboot, Connecting to new Work-SSID.
6. Connecting to Server
7. Standby
8. Load Card
9. Use coin from card
10. Plug-in type
11. Swipe card type
12. Out ticket from Game
13. Cache out ticket: First out the ticket to the card reader, then choose to save to card or out the real ticket from card reader.
14. Only out real-ticket. All game ticket out by real ticket, no e-ticket.
15. Only out e-ticket. All game ticket out to last card, no real ticket.
16. Auto out real-ticket or e-ticket. Insert card then out e-ticket, no card inserted out real ticket.
17. Choose how to handle the cache ticket in card reader. Save to card, or just out a real ticket.

###### Change the card reader setting with Function-Card