# 新中新webapi读卡接口说明

2022-9-23 V3.3.3

**目 录**

[新中新webapi读卡接口说明 1](#_Toc91605028)

[一、接口说明 2](#_Toc91605029)

[1.简介 2](#_Toc91605030)

[2.适配读卡器类型 3](#_Toc91605031)

[二、部署 3](#_Toc91605032)

[三、测试 3](#_Toc91605033)

[1.身份证Webapi接口 4](#_Toc91605034)

[2.身份证Webapi接口参数 6](#_Toc91605035)

[3.身份证Socket.io接口 7](#_Toc91605036)

[~~4.人脸识别Webapi接口~~ 8](#_Toc91605037)

[5. M1卡Webapi接口 9](#_Toc91605038)

[6.指纹Webapi接口 10](#_Toc91605039)

[7.控制蜂鸣器 13](#_Toc91605040)

[8.F200语音控制 13](#_Toc91605041)

[9.获取读卡服务版本 14](#_Toc91605042)

[10.常见问题 14](#_Toc91605043)

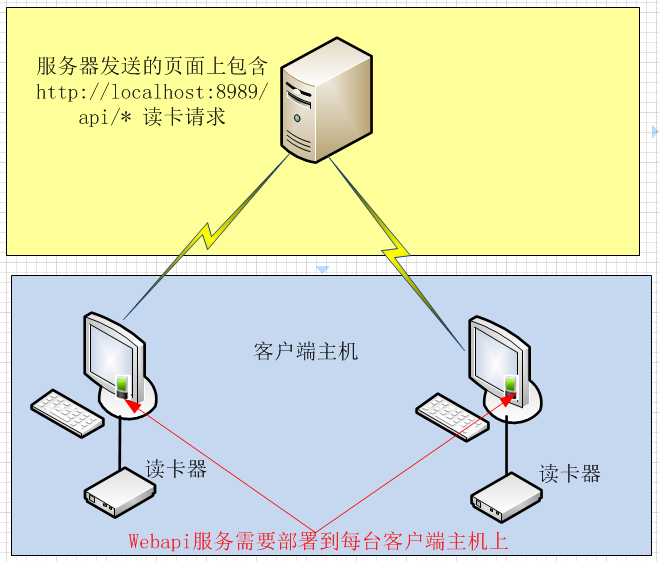
[四、更新说明 15](#_Toc91605044)

## 一、接口说明

### 1.简介

用GO实现的一个WEB程序，公开供浏览器调用的WEBAPI，替代原有的OCX方案，实现跨浏览器的目的。只要浏览器能支持JavaScript就可以使用，之前其他流行的Java、C#语言调用DLL只能使用32位版本，调用WEBAPI可以不用考虑64或32的问题。二代证读卡器存在比较大的问题是解码照片的时候需要向本地写入数据，在不同的操作系统上兼容性比较差，Java和C#由于是解释运行也经常在解码照片的时候出问题。

采用该方案之后，读卡和解码照片由服务程序完成，可以在最大程度上避免该问题的发生。可在浏览器上发送http get请求，WEBAPI服务接到请求后读卡，返回json格式的读卡记录信息。还可以使用socket.io的方式，WEBAPI服务读卡后向浏览器发送结果，避免浏览器轮询造成的资源浪费。



**公网测试地址：**<https://onecardok.com.cn:20100/>

### 2.适配读卡器类型

本webapi接口已适配联机型读卡器，包括需要安装驱动和免驱类型的二代证读卡器。还包括HF型二合一读卡器，可对M1卡进行操作。

## 二、部署

本读卡web服务程序需要安装到连接读卡器的电脑上，服务程序访问USB接口和读卡器建立通讯。

默认安装位置D:\Program Files\XZX\_WEBAPI。

config.ini文件和证书秘钥文件位置:

C:\Users\用户名\AppData\Roaming\Synjones

或C:\Documents and Settings\用户名\Application Data\Synjones

如果需要在批量安装的时候统一修改配置参数，可将配置好的config.ini和安装程序放到同一文件夹内，安装程序将在安装结束后将config.ini文件复制到对应的目录中。

程序使用config.ini来配置服务的端口：

其中webapi的默认端口HTTP webapiPort: 8989，HTTPS webapiPort: 9199， socket.io的默认端口socketioPort: 5000。

~~Https使用server.crt和server.key作为证书和秘钥文件，替换时请保持命名一致。~~(已弃用,Https页面可以直接调用HTTP的localhost接口)

LocalhostOnly：配置服务是否只在本机访问，默认只能本机访问，如果需要在其他的计算机上访问服务，需要设置为false

enableQuit：配置是否可以关闭服务，可以使用stop.bat来关闭服务

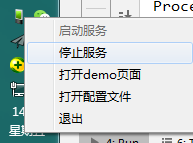
## 三、测试

公网测试地址：<https://onecardok.com.cn:20100/>

此测试页面位于安装目录下的\测试页面。

本地测试：

启动服务后，可右键单击服务图标，打开菜单，选择打开demo页面，或打开<http://localhost:8989/#/demo>进行测试。



如果使用IE浏览器测试可查看测试页面：测试页面位于安装目录下的\测试页面。

### 1.身份证Webapi接口

**读身份证卡片**

- 例如在config.ini使用默认webapiPort端口8989，Https使用9199端口，则可在浏览器的地址栏直接输入[http://localhost:8989/api/ReadMsg](http://127.0.0.1:8080/api/ReadMsg) 或[https://localhost:9199/api/ReadMsg](https://127.0.0.1:9199/api/ReadMsg) 来读卡，或可打开安装目录下"XZX\_WEBAPI/测试页面/JS读卡测试/CallWebAPI.html"进行测试。返回json字符串，格式如下：

**居民（港澳台）身份证信息返回示例：**

**{**

**"code"**: "0"**,** //返回值 0成功，其他错误

**"retcode"**: "0x90 0x1"**,** //状态值，第一个值为读卡状态，0x90成功，第二个值为照片解码状态，0x1成功

**"retmsg"**: "操作成功||相片解码解码正确"**,** //返回值信息

**"errmsg"**: ""**,**

**"name"**: "张红叶"**,** //姓名

**"sex"**: "女"**,** //性别

**"nation"**: "汉"**,** //民族或国籍(外国人居住证)

**"born"**: "19881118"**,** //出生日期

**"address"**: "河北省邯郸市临漳县称勾镇称勾东村复兴路25号"**,** //住址

**"cardno"**: "130423198811184328"**,** //证件号码

**"userlifeb"**: "20110330"**,** //有效期开始

**"userlifee"**: "20210330"**,** //有效期结束

**"police"**: "临漳县公安局"**,** //签发机关

**"photobase64"**: "iVBORw0……AAElFTkSuQmCC"**,** //证件头像

**"FpDescribe"**: ""**,** //证内指纹描述

**"FpData"**: null**,** //证内指纹原始数据

**"FpFingerName1"**: "右手拇指"**,** //指纹1名称

**"FpQuality1"**: "74"**,** //指纹1质量

**"FpRegResult1"**: "注册成功"**,** //指纹1注册结果

**"FpFeature1"**: "4301……"**,** //指纹1特征值

**"FpFingerName2"**: ""**,** //指纹2名称

**"FpQuality2"**: ""**,** //指纹2质量

**"FpRegResult2"**: ""**,** //指纹2注册结果

**"FpFeature2"**: ""**,** //指纹2特征值

**"CardType"**: "0"**,** //证件类型：内地为0，外国人为1，港澳台居住证此处为2

**"EngName"**: ""**,** //英文姓名

**"CertVol"**: "01"**,** //证件版本号

**"PassID"**: "000000000"**,**//港澳台居住证通行证号码

**"IssuesTimes"**: "01"**,** //港澳台居住证签发次数

**"frontImg"**: ""**,** //证件正面图片

**"backImg"**: "" //证件背面图片

**}**

**外国人居住证返回示例：**

**{**

**"code"**: "0"**,**

**"retcode"**: "0x90 0x1"**,**

**"retmsg"**: "操作成功||相片解码解码正确"**,**

**"errmsg"**: ""**,**

**"name"**: "证件样本"**,**

**"sex"**: "女"**,**

**"nation"**: "加拿大/CAN"**,**

**"born"**: "19810803"**,**

**"address"**: ""**,**

**"cardno"**: "CAN110081080319"**,**

**"userlifeb"**: "20151025"**,**

**"userlifee"**: "20251024"**,**

**"police"**: "1500"**,**

**"photobase64"**: "iVBO……lFTkSuQmCC"**,**

**"FpDescribe"**: ""**,**

**"FpData"**: null**,**

**"CardType"**: "1"**,**

**"EngName"**: "ZHENGJIAN,YANGBEN"**,**

**"CertVol"**: "01"**,**

**"frontImg"**: ""**,**

**"backImg"**: ""

**}**

**读取公安部模块编号**

[http://localhost:8989/api/GetSAMID](http://127.0.0.1:8080/api/GetSAMID)

**已过时的api**

~~# http://localhost:8989/api/ReadMsgJsonp 以Jsonp方式返回解决跨域问题~~

~~# http://localhost:8989/api/ReadMsgIE IE浏览器下使用~~

~~# http://localhost:8989/api/ReadFpMsg 读取带指纹的身份证信息~~

~~# http://localhost:8989/api/SetPort~~

~~# http://localhost:8989/api/GetSAMStatus~~

**~~获取身份证正反面照片~~**

[~~http://localhost:8989/api/GetImage~~](http://localhost:8080/api/GetImage)

~~返回正反面base64编码字符串~~

~~{"retcode":"","frontImg":"","backImg":""}~~

**读取身份证唯一序列号**

<http://localhost:8989/api/GetIDSerialNo>

{"retcode":"0","retmsg":"32472282800920A0 ","errmsg":""}

### 2.身份证Webapi接口参数

**ie=1：返回ie浏览器可显示的数据**

设置ie=1时，即设置"Content-Type"为"text/html;charset=utf-8"，默认为"application/json"。

**Fp=1: 读身份证时读取指纹信息**

**NoPhoto=1 ：设置不解码身份证头像照片，返回的photobase64将为空**

**PhotoQuality: 身份证照片JPG压缩质量，默认100.**

**PhotoQuality=101时将保存Bmp格式的照片**

**cardImg=1:返回身份证正反面图片，照片以Base64字符形式存储在Json返回结构的frontImg和backImg字段中。**

**readOnce=1: 同一张身份证只读第一次。**

**waitTime:** 读卡超时单位秒 最长3600秒

**callback** : **以JSONP方式返回数据**

示例：[http://localhost:8989/api/ReadMsg?ie=1&Fp=1&callback=GetIDCard](http://127.0.0.1:8080/api/ReadMsg?ie=1&Fp=1&callback=GetIDCard)

返回ie可显示的数据，读卡时读取卡内指纹信息,以jsonp方式返回数据

GetIDCard(**{**

**"code"**: "0"**,**

**"retcode"**: "0x90 0x1"**,**

**"retmsg"**: "操作成功||相片解码解码正确"**,**

**"errmsg"**: ""**,**

**"name"**: "北京"**,**

**"sex"**: "男"**,**

**"nation"**: "汉"**,**

**"born"**: "19961123"**,**

**"address"**: "住址住址住址撒旦法按时地方"**,**

**"cardno"**: "692474199611237822"**,**

**"userlifeb"**: "20041005"**,**

**"userlifee"**: "20241005"**,**

**"police"**: "签发机关诶我诶度毫微度"**,**

**"photobase64"**: "iVBORw0KG……FTkSuQmCC"**,**

**"FpDescribe"**: "指纹1:右手食指注册成功,质量96,指纹2:右手食指注册成功,质量96"**,**

**"FpData"**: "4301171……000000004b"**,** //16进制字符串

**"CardType"**: "0"**,**

**"EngName"**: ""**,**

**"CertVol"**: ""**,**

**"frontImg"**: ""**,**

**"backImg"**: ""

**}**);

### 3.身份证Socket.io接口

- socketioPort端口5000，可打开安装目录下的“XZX\_WEBAP/测试页面/socket.io/index.html”，或者在浏览器地址栏输入[http://localhost:5000/](http://127.0.0.1:5000/) 来加载安装目录下XZX\_WEBAPI/asset/index.html测试页面。

测试页面的源码中使用如下方式创建和使用sokcet

socket = io({'timeout': 300000,'reconnectionDelayMax':1000,'reconnectionDelay':500});

socket.emit('startRead');//发送读卡指令

socket.on('card message', function(msg){//接收读卡信息

var base = new Base64();

var result1 = base.decode(msg); //解码

$('#messages').append($('<li>').text(result1));//显示

});

打开测试页面时默认发送开始读卡指令，读卡成功后自动显示，不读卡的时候需要发送socket.emit('stopRead')来停止读卡。

### ~~4.人脸识别Webapi接口~~

此接口需要配合人脸识别硬件加密设备使用并需要安装人脸核验版本的webapi服务

测试页面位于安装目录下的\测试页面\JS读卡测试\

1、人脸比对(GET/POST方式提交)

[http://localhost:8989](http://127.0.0.1:8080)/api/FaceMatch?threshold=70&detect\_count=3&whole\_image=false&voice=true

header应设置 Content-Type 为 application/json

调用此接口时，会自动弹出对比窗口，显示正在对比的图像。

threshold设置核验阈值，默认为70，当对比分数高于此值时，对比窗口关闭，返回对比结果。

detect\_count 设置核验次数，超过次数即为核验失败，自动关闭核验窗口，默认值为3。

whole\_image 设置现场截取照片的大小，默认false只截取人脸，true为整张图片。

voice 设置是否开启语音提示，默认true开启。

图像数据使用base64编码传输，支持Jpg和Bmp格式的图片。

可在url或data中用参数image传递图像数据

post data数据为json格式

{

"image": "/9j/2……avUGB//9k=",//待比对的图像的base64编码数据

}

返回

{

"retcode": "0",//0为成功，其他值为错误

"retmsg": "成功"//状态描述

"Score": "99"//对比分数

}

### 5. M1卡Webapi接口

**M1卡复位(寻找M1卡)**

[http://localhost:8989/api/M1Reset](%20http:/127.0.0.1:8080/api/M1Reset)

返回卡号和卡类型

{

retcode:"0"

retmsg:"寻卡成功"

errmsg:""

CardSn1:"DE7B3BD2"

CardSn2:"D23B7BDE"

Size:"8"

data:""

}

**M1卡认证**

[http://localhost:8989/api/M1AuthenKey?KeyType=0&BlockNo=9&Key=ffffffffffff](http://127.0.0.1:8080/api/M1AuthenKey?KeyType=0&BlockNo=9&Key=ffffffffffff)

参数KeyType：验证秘钥类型，KeyType=0验证A秘钥，KeyType=1验证B秘钥；BlockNo：验证块号；Key：6字节秘钥，返回成功或失败信息：

{

retcode:"0"

retmsg:"验证成功"

errmsg:""

}

**读块数据**

[http://localhost:8989/api/M1ReadBlock?BlockNo=9](http://127.0.0.1:8080/api/M1ReadBlock?BlockNo=9)

读块数据 参数BlockNo：待读块号 返回16字节块数据（16进制）

{

retcode:"0"

retmsg:"读块成功"

errmsg:""

CardSn1:"" //卡号

CardSn2:"" //和CardSn1字节顺序颠倒的卡号 用户根据自己需要选择使用

Size:""

data:"00112233445566778899AABBCCDDEEEE"

}

**写块数据**

[http://localhost:8989/api/M1WriteBlock?BlockNo=9&Data=11223344556677889900112233445566](http://127.0.0.1:8080/api/M1WriteBlock?BlockNo=9&Data=11223344556677889900112233445566)

参数BlockNo：待写块号，Data：16字节块数据（16进制），返回

{

retcode:"0"

retmsg:"写卡成功"

errmsg:""

}

**停卡**

[http://localhost:8989/api/M1Halt](http://127.0.0.1:8080/api/M1Halt)

不操作M1卡时可调用此函数

{

retcode:"0"

retmsg:"停卡成功"

errmsg:""

}

### 6.指纹Webapi接口

使用F200型机器，可进行指纹的采集和核验。

**采集工作流程为：**

1. 初始化指纹仪InitFP，
2. 采集指纹GetFPData
3. 关闭指纹CloseFP

**指纹初始化**InitFP：

如果初始化失败，可能是没有安装驱动，可查看设备管理器，更新安装目录里的驱动。  
[http://localhost:8989/api/InitFP](http://127.0.0.1:8080/api/InitFP)

**获取指纹数据**GetFPData**：**  
[http://localhost:8989/api/GetFPData](http://127.0.0.1:8080/api/GetFPData)

**参数：**

**waitTime:**超时单位秒 最长60秒

**waitScanScore:等待扫描的图像质量大于该值的时候再返回**

返回  
{  
"retcode":"1", //1成功 其他值错误  
"retmsg":"采集成功",  
"errmsg":"",  
"score":"58", //采集图像的质量，范围0-100  
"FpBmpImg":"Qk02bAEAA……jp3tzy+A==", //指纹图像  
"FpFeature":"4301171……000000004b" //指纹特征值  
}

**关闭指纹**CloseFP**：**  
[http://localhost:8989/api/CloseFP](http://127.0.0.1:8080/api/CloseFP)

**核验的工作流程为：**

1. 初始化指纹仪InitFP，
2. 对比指纹GetFPMatch，待核验的指纹数据FpFeature来自采集的指纹特征或身份证内的指纹特征。
3. **FPMatch** 进行1:N的特征值比对
4. 关闭指纹CloseFP

**指纹初始化**InitFP：

如果初始化失败，可能是没有安装驱动，可查看设备管理器，更新安装目录里的驱动。  
[http://localhost:8989/api/InitFP](http://127.0.0.1:8080/api/InitFP)

**对比指纹**GetFPMatch：

参数

FpFeature是512或1024字节的指纹特征

threshold用来设置对比的阈值，默认是0.6

**waitTime:**超时单位秒 最长60秒

**waitScanScore:等待扫描的图像质量大于该值的时候再返回**  
[http://localhost:8989/api/GetFPMatch?threshold=0.8&FpFeature==4301171……000000004b](http://127.0.0.1:8080/api/GetFPMatch?threshold=0.8&FpFeature==4301171%A1%AD%A1%AD000000004b)    
返回值：

{  
    "retcode":"1", //1成功 其他值错误  
    "retmsg":"核验成功", //核验结果 根据阈值threshold和比对结果score给出成功或失败提示  
    "errmsg":"",  
    "score":"1.000000", //比对结果 指纹的相似度，范围0-1，默认阈值可选择大于0.6，认为指纹是同一个人的，如果应用的安全要求较高，可自定义较高的阈值。  
    "ScanScore":"80", //核验时采集的图像质量  
    "Finger":"" //给出核验匹配的指位  
}

**对比指纹特征 FPMatch**：

**请求方式：POST**

Form-data:

FpFeature 是512字节的指纹特征

threshold 用来设置对比的阈值，默认是0.6

**findfirst** 如果是0 则将列表里的全比对一遍 返回分数最高且大于阈值的

如果是1 则返回第一个高于阈值的

**list 待比对的特征值数组，使用id进行区分，每个特征值512字节，1024个字符。**

[

{

"id": "1",

"feature": "4301171……000000004b "

},

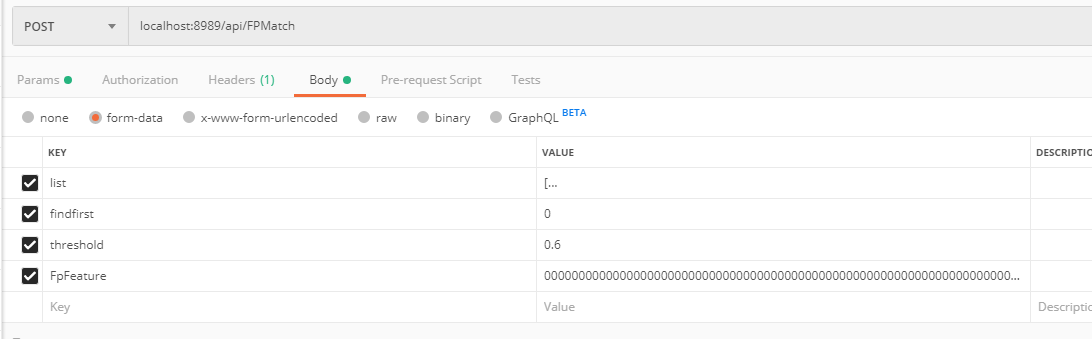
{

"id": "2",

"feature": "4301171……000000004b "

}

]



返回值：

{  
    "retcode":"1", //1成功 其他值错误  
    "retmsg":"核验成功", //核验结果 根据阈值threshold和比对结果score给出成功或失败提示  
    "errmsg":"",  
    "score":"1.000000", //比对结果 指纹的相似度，范围0-1，默认阈值可选择大于0.6，认为指纹是同一个人的，如果应用的安全要求较高，可自定义较高的阈值。  
    "ScanScore":"",

    "Finger":"1" //list中对应的id值  
}

**关闭指纹**CloseFP：  
[http://localhost:8989/api/CloseFP](http://127.0.0.1:8080/api/CloseFP)

### 7.控制蜂鸣器

仅部分免驱读卡器支持

[http://localhost:8989/api/Beep?BeepTime=500](http://127.0.0.1:8080/api/Beep?BeepTime=500)

控制蜂鸣器响，BeepTime控制蜂鸣器响的时间，以毫秒为单位，默认响1000毫秒。

### 8.F200语音控制

[http://localhost:8989/api/playSound?sound=左手食指](http://127.0.0.1:8080/api/playSound?sound=左手食指)

[http://localhost:8989/api/playSound?selfDefSound=11](http://127.0.0.1:8080/api/playSound?selfDefSound=11)

参数sound和selfDefSound对应关系

sound : selfDefSound

"右手拇指":0xb,

"右手食指":0xc,

"右手中指":0xd,

"右手环指":0xe,

"右手小指":0xf,

"左手拇指":0x10,

"左手食指":0x11,

"左手中指":0x12,

"左手环指":0x13,

"左手小指":0x14,

"采集成功":0x20,

"采集错误":0x21,

"核验完成":0x22,

"核验错误":0x23,

"请重新采集":0x24,

"读卡成功":0x25,

"读卡成功":0x26,

"读卡错误":0x27,

### 9.获取读卡服务版本

[http://localhost:8989/api/Version](http://127.0.0.1:8080/api/Version)

返回：

{"retcode":"0","retmsg":"v3.0","errmsg":""}

### 10.常见问题

* **解决谷歌浏览器最新chrome94版本CORS跨域问题**

CORS跨域问题：

升级谷歌浏览器最新chrome94版本后，提示Access to XMLHttpRequest at '[http://localhost:xxxx/api](https://link.zhihu.com/?target=http%3A//localhost%3A50022/api/get_version)' from origin '[http://xxx.xxx.com:x](https://link.zhihu.com/?target=http%3A//ydrm.yd-data.com%3A9123)xxx' has been blocked by CORS policy: The request client is not a secure context and the resource is in more-private address space `local`.

**解决办法：**

打开浏览器，进入chrome://flags/页面

搜索Block insecure private network requests

设置为Disabled，重新打开浏览器就好了。

* **并发访问**

所有接口不支持并发访问，当有接口正在处理时，新的请求将返回读卡中提示。

## 四、更新说明

V3.3.2 修复解码错误。2022-9-23

V3.3.2 民族增加穿青人。2022-1-7

V3.3.1 增加其他民族编码提示。2022-1-7

V3.3.0 并发访问时返回读卡器正忙，延长读卡等待最大时间。2021-12-29

V3.2.9 修改指纹对比接口传参方式 2021-12-21

V3.2.8 增加指纹特征值比对接口 2021-12-03

V3.2.7文档更新，程序增加BMP格式照片 2020-10-19

V3.2.6解决未连接读卡器时仍打开端口的问题 2020-07-31

V3.2.5增加usb读身份证序列号2020-07-03

V3.2.4修改IE demo 2019-10-24

V3.2.3增加读卡时指纹1和指纹2字段

V3.2.2增加usb插拔检测。2019-09-16

V3.2.1 增加返回值，增加新的demo页面2019-09-06

V3.2.0 解决xp上UI显示问题

V3.1.2增加指纹操作demo，修复指纹操作的一些问题，更新接口返回值。2019-08-30

V3.1.1增加照片解码错误处理，增加Socket.io方式同一张身份证只读一次的readOnce接口。2019-08-12

V3.1.0增加同一张身份证只读一次的readOnce接口。

V3.0.9安装时部署https证书（仅在Win7及以上系统可用），同时启用http和https接口，修改配置文件和证书文件位置。

V3.0.8修改人脸核验动态库，优化人脸检测。2019-01-16

V3.0.7增加F200语音提示。2019-01-03

V3.0.6修改临时文件存储位置，解决win7及以上系统安装在C盘时无权限写文件的问题，增加人脸核验拍照功能，修改默认可退出服务。2018-12-20

V3.0.5优化人脸核验打开速度，增加http url传递照片 2018-11-20

V3.0.4修改测试页面 2018-11-05

V3.0.3修改窗口显示和读卡器未连接处理方式 2018-11-01

V3.0.2增加人脸核验调用接口参数，增加截图大小配置，增加声音提示 2018-10-30

V3.0.1修改人脸核验调用接口 2018-10-19

V3.0.0增加指纹核验接口，增加不解码身份证照片的参数2018-10.10

V2.9.9增加人脸识别接口 2018-09.30

V2.9.8增加读取港澳台居民居住证正反面照片。2018-09-20

V2.9.7增加读取港澳台居民居住证。2018-08-20

V2.9.6增加Https接口。2018-08-01

V2.9.5修改获取身份证正反面照片接口，更新测试工具。2018-07-11

V2.9.4更新动态库。

V2.9.3修改未连接读卡器提示。

V2.9.2增加获取身份证正反面照片，修改测试页面。

V2.9.1 m1卡返回卡号格式化。2018-04-03

V2.9 修复个别身份证读指纹时无返回的bug。2018-03-30

V2.8 更新动态库，修复个别读卡器读卡问题，增加蜂鸣器调用。2018-03-28

V2.7 身份证照片修改为jpg格式，增加设置照片图像质量接口。2017-11-17

V2.6 修改读身份证时返回的指纹特征格式为16进制字符串。2017-11-13

V2.5 增加读取外国人居住证功能。2017-09-07

V2.4 XP系统下开机自动后台运行，不显示窗口。2017-09-06

V2.3 自动安装二代证模块USB驱动。2017-08-10

V2.2 修复点击程序无反应的情况。2017-07-27

V2.1 修复读卡属性值颠倒问题，修改测试页面 2017-07-13。

V2.0 修复并发读卡时错误，只允许一个程序实例允许。

V1.9 修复问题 2017-07-4

V1.8 安装包兼容winXp和win7及以上系统 2017-06-22

V1.7 增加安装包，开机自动运行，最小化到托盘等功能，修复未连接读卡器时操作M1卡返回值错误bug。安装时自动拷贝License.dat到C盘根目录。

V1.6 增加免驱读M1卡功能，增加调用时传递参数

V1.5 增加免驱读二代证功能

V1.4 增加解码license