

识别服务开发文档

安全解决方案

修订版本: V5.5.1.8

日期: 2018 年 04 月 08 日

文档修改目录

日期	修改说明	变更人	审核人
2018-04-08	V5.5.1.8 版本号更新	杨敏、李朝需	

目 录

第一章 服务简介.....	5
1.1 服务版权说明.....	4
1.2 客户端运行环境.....	4
1.3 识别服务主要功能描述.....	4
1.4 核验服务主要功能描述.....	4
1.5 识别服务调用约束.....	5
1.6 核验服务调用约束.....	5
第二章 识别服务接口调用.....	6
2.1 识别服务接口说明.....	6
2.1.1 客户端调用 http 识别服务接口.....	7
2.1.3 移动客户端调用 webservice 识别服务接口.....	15
2.1.4 辅助函数 setRecgnPlainParam 说明.....	16
2.2 服务结果返回字符串格式.....	18
2.2.1 识别结果字符串示例说明.....	18
2.3 示例程序.....	21
2.3.1 PC 客户端调用识别服务示例程序.....	21
2.4 支持格式.....	22
第三章 核验服务接口调用.....	23
3.1 核验服务接口说明.....	23
3.1.1 客户端调用核验服务 http 接口.....	24
3.1.2 客户端调用核验服务 WebService 接口.....	28
3.2 支持格式.....	30
第四章 注意事项.....	32
4.1 服务需要连网调用.....	32
4.2 对识别图像的要求.....	32
4.2.1 一般要求.....	32
4.2.2 对扫描图像的要求.....	32
4.2.3 对拍摄图像的要求.....	33
附录一：证件类型编码.....	34
技术支持.....	37

第一章 服务简介

1.1 服务版权说明

识别服务版权归属北京中安未来科技有限公司所有，未经北京中安未来科技有限公司授权，不得擅自进行使用。

1.2 客户端运行环境

当客户端使用的编程语言为 Java 时，请使用 1.6 及以上版本 JRE。

1.3 识别服务主要功能描述

服务主要实现了以下功能：

1)扫描或拍照设备采集各种证件（包括一代身份证、二代身份证，二代身份证背面，驾照、护照、军官证、回乡证、港澳通行证、行驶证等）、银行卡、车牌、名片等彩色图像文件，通过 **http** 或 **WebService** 两种接口方式进行调用识别；

2) 将识别结果以 XML 格式或 json 格式返回给调用者；

1.4 核验服务主要功能描述

服务主要实现了以下功能：

1) **身份证号核验服务**：可快速联网核查身份证姓名和号码是否一致。

2) **身份证核验服务**：可快速联网核查身份证姓名和号码是否一致，并核验上传身份证的头像是否为真。

3) **身份核验服务**：可快速联网核查身份证姓名和号码是否一致，并比对现场人像与权威数据源中身份证人像的相似度，甄别持证人和证件是否为同一人。

4) **手机号实名核验服务**：可快速核验姓名、身份证号与该手机号实名登记的姓名、身份证号是否一致；

5) **银行卡实名核验服务**：可快速核验姓名、身份证号、手机号与该银行卡开户时预留的姓名、身份证号、手机号是否一致。

6)通过 **http** 或 **WebService** 两种方式进行核验，将识别结果以 XML 或 json 格式返回给调用者；

1.5 识别服务调用约束

在调用识别服务前，请您确保客户端程序或设备与识别服务器网络连接畅通。要求图像文件大小在 **300KB** 左右，建议扫描图像分辨率为 **300DPI**。

1.6 核验服务调用约束

在调用核验服务前，请您确保客户端程序或设备与核验服务器网络连接畅通。核验服务器端能够连接外网，并且已获取 **OCRKey** 和 **OCRSecret** (<http://netocr.com/>注册购买可获取 **OCRKey** 和 **OCRSecret**)

第二章 识别服务接口调用

本章将对识别服务接口进行阐述。由于实际所使用到的编程语言较多，此文档不会一一涉及。本章最后小节将只就 [Java 语言如何调用识别服务接口](#) 进行代码示例讲解。其它语言如何进行调用，用户可参照接口调用例程。

2.1 识别服务接口说明

使用识别服务前请确认服务器 ip 地址及端口号。

2.1.1 客户端调用 http 识别服务接口

2.1.1.1(证件,银行卡,车牌,名片,人脸,文档,VIN)识别调用 http 接口说明:

1)图像以 base64 编码调用说明:

http 方法:	必须为 POST
url:	http://IP:Port/cxfServerX/doAllCardRecon 如:http://127.0.0.1:8080/cxfServerX/doAllCardRecon
Post 实体内容:	Post 实体内容仅为一个 json 对象。 json 对象中封装了 4 对参数, 详情见下文
返回值	String 类型的 xml 字符串

Post 实体内容: 客户端将 4 对 String 类型的参数, 以键值对的形式, 封装到一个 json 对象中。

4 个参数如下:

String username 用户白名单, 需与服务器配置相同如 “test”;

String paramdata 图像文件 base64 编码与其他参数拼接而成;
由开发包中的 setRecgnPlainParam 函数获得, 详情请参考代码示例。可调用 setRecgnPlainParam 方法得到;
(setRecgnPlainParam 方法说明见 2.1.4)

String signdata 为“NULL” (注意:此处为带引号字符串格式且大写)

String imgtype 为图片格式, 如 “jpg、bmp” ;

注意: 若为人脸调用则需要将两张图片的 base64 以 faceRecon 分割组成一个串

String paramdata = strsrc1 + "faceRecon" + strsrc2;

两个人脸识别的图片的格式, 以 faceRecon 分割按前后顺序组成一个串

String imgtype = imgtype1+ "faceRecon" + imgtype2;

将上述 4 对参数封装到 json 对象中, java 代码参考如下:

```
JSONObject json = new JSONObject();//封装json对象
String username="test";
String paramdata=strsrc;
String signdata="NULL";
String imgtype="jpg";
json.put("username", username);
json.put("paramdata", paramdata);
json.put("signdata",signdata);
```

```
json.put("imgtype",imgtype);
```

2) (证件, 银行卡, 车牌, 名片, 文档, VIN) 图像以文件流编码调用说明:

http 方法:	必须为 POST
url:	http://ip:port/cxfServerX/DoAllCardReconFile 如: http://192.168.0.54:8888/cxfServerX/DoAllCardReconFile?username=test&typeid=2
返回值	String 类型的 xml 字符串或 json 字符串

String username 用户白名单, 需与服务器配置相同如 “test”;
String typeid 编码类型 (详见附录一)

```
// 读取文件上传到服务器
File file = new File(filePath);
FileInputStream fileInputStream = new FileInputStream(file);
byte[] bytes = new byte[1024];
int numReadByte = 0;
while ((numReadByte = fileInputStream.read(bytes, 0, 1024)) > 0) {
    out.write(bytes, 0, numReadByte);
}
out.flush();
fileInputStream.close();
```

2.1.1.2 (人证识别) 调用调用 http 接口说明:

http 方法:	必须为 POST
url:	http://IP:Port/cxfServerX/DoFaceCardRecon 如: http://192.168.0.158:8080/cxfServerX/DoFaceCardRecon
Post 实体内容:	Post 实体内容仅为一个 json 对象。 json 对象中封装了 4 对参数, 详情见下文
返回值	String 类型的 xml 格式字符串或 json 格式字符串

Post 实体内容: 客户端将 4 对 String 类型的参数, 以键值对的形式, 封装到一个 json 对象中。

4 个参数如下:

String username 用户白名单, 需与服务器配置相同如 “test”;
String faceimg 人脸图像文件 base64 编码;
String idcardimg 证件类型图像文件 base64 编码;
String cardtype 证件类型编码 (编码类型见附录一);

将上述 4 对参数封装到 json 对象中, java 代码参考如下:

```
JSONObject json = new JSONObject(); 封装json对象
String username="test";
String faceimg=Base64.GetBase64StrFromImage("D:/TestYb/t.jpg");;
String idcardimg=Base64.GetBase64StrFromImage("D:/TestYb/erdai.jpg");
String cardtype="2";
json.put("username", username);
json.put("faceimg", faceimg);
json.put("idcardimg",idcardimg);
json.put("cardtype",cardtype);
```

2.1.1.3(营业执照)识别调用 http 接口说明:

http 方法:	必须为 POST
url:	http://IP:Port/cxfServerX/DoBulicenseRecon 如: http://192.168.0.158:8080/cxfServerX/DoBulicenseRecon
Post 实体内容:	Post 实体内容仅为一个 json 对象。 json 对象中封装了 4 对参数, 详情见下文
返回值	String 类型的 xml 格式字符串或 json 格式字符串

Post 实体内容: 客户端将 4 对 String 类型的参数, 以键值对的形式, 封装到一个 json 对象中。

4 个参数如下:

String username	用户白名单, 需与服务器配置相同如 “test”;
String imgbase64	营业执照图像文件 base64 编码;
String cardtype	营业执照编码类型(2008)
String outvalue	营业执照输出字段(0 输出所有字段, 1-18 选择输出, 输出多个字段用,隔开如: String outvalue=1,2,3;)。 输出对应字段见营业执照识别结果示例;

```
JSONObject json = new JSONObject(); 封装json对象
String username="test";
String imgbase64=Base64.GetBase64StrFromImage("D:/TestYb/1.jpg");
String cardtype="2008";
String outvalue = "0";//(可选择输出字段)
json.put("username", username);
json.put("imgbase64", imgbase64);
json.put("cardtype",cardtype);
json.put("outvalue",outvalue);
```

2.1.1.4(发票)识别调用http接口说明:

http 方法:	必须为 POST
url:	http://IP:Port/cxfServerX/DoVatinvoice 如: http://192.168.0.158:8080/cxfServerX/DoVatinvoice
Post 实体内容:	Post 实体内容仅为一个 json 对象。 json 对象中封装了 4 对参数, 详情见下文
返回值	String 类型的 xml 格式字符串或 json 格式字符串

Post 实体内容: 客户端将 4 对 String 类型的参数, 以键值对的形式, 封装到一个 json 对象中。

4 个参数如下:

String username 用户白名单, 需与服务器配置相同如 “test”;
 String imgbase64 增值税票图像文件 base64 编码;
 String cardtype 类型编码 (编码类型见附录一)
 String outvalue 发票输出字段(0 输出所有字段, 选择输出字段用, 隔开如: String
 outvalue=1,2,3; 发票自动分类字段全部返回不支持选择字段输出)。

```
JSONObject json = new JSONObject(); 封装json对象
String username="test";
String imgbase64=Base64.GetBase64StrFromImage("D:/TestYb/1.jpg");
String cardtype="2009";
String outvalue = "0";//(可选择输出字段)
json.put("username", username);
json.put("imgbase64", imgbase64);
json.put("cardtype", cardtype);
json.put("outvalue", outvalue);
```

2.1.2 PC 客户端调用 webservice 识别服务接口

2.1.2.1 (证件, 银行卡, 车牌, 名片, 人脸, 文档, VIN)识别调用 webservice 接口说明:

服务地址:	http://IP:port/cxfServerX/ImgReconCard?wsdl 如:http://192.168.0.157:8080/cxfServerX/ImgReconCard?wsdl (具体地址请询问中安公司开发人员)
服务方法:	String doAllCardRecon(String username, String paramdata, String signdata, String imtype);
功能:	识别
返回值:	String 类型的 xml 格式字符串或 json 格式字符串

其中:

doAllCardRecon 方法的 4 个参数说明如下:

String username 用户白名单, 需与服务器配置相同如 “test”;

String paramdata 图像文件 base64 编码与其他参数拼接而成; 由开发包中的 **setRecgnPlainParam** 函数获得, 详情请参考代码示例。可调用 **setRecgnPlainParam** 方法得到;(**setRecgnPlainParam** 方法说明见 2.1.4)

String signdata 固定为 **null**;(注意:不带引号)

String imtype 为图片格式, 如 “jpg、bmp”;

注意: 若为人脸调用则需要将两张图片的 base64 以 **faceRecon** 分割组成一个串

String paramdata = strsrc1 + "faceRecon" + strsrc2;

两个人脸识别的图片的格式, 以 **faceRecon** 分割按前后顺序组成一个串

String imgtype = imgtype1+ "faceRecon" + imgtype2;

java 代码参考如下:

```
// String doAllCardRecon(String username, String paramdata,  
String signdata, String imtype);  
String result = cardImgreconAll.doAllCardRecon("test", strsrc, null,"jpg");
```

2.1.2.2 （人证识别）调用 webservice 接口说明：

服务地址：	http://IP:port/cxfServerX/ImgReconCard?wsdl 如:http://192.168.0.157:8080/cxfServerX/ImgReconCard?wsdl (具体地址请询问中安公司开发人员)
服务方法：	doFaceCardRecon(String username, String faceimg,String idcardimg,String cardtype)
功能：	识别
返回值：	String 类型的 xml 格式字符串或 json 格式字符串

doFaceCardRecon 方法的 4 个参数说明如下：

String username 用户白名单，需与服务器配置相同如 “test”;;

String faceimg 人脸图像文件 base64 编码；

String idcardimg 证件图像文件 base64 编码；

String cardtype 证件类型编码（编码类型见附录一）；

java 代码参考如下：

```
// String doFaceCardRecon(String username, String faceimg,  
String idcardimg , String cardtype );  
String result = cardImgreconAll.doFaceCardRecon("test", faceimg, idcardimg,"2");
```

2.1.2.3 （营业执照）识别调用 webservice 接口说明：

服务地址：	http://IP:port/cxfServerX/ImgReconCard?wsdl 如:http://192.168.0.157:8080/cxfServerX/ImgReconCard?wsdl (具体地址请询问中安公司开发人员)
服务方法：	doBulicenseRecon(String username, String imgbase64, String cardtype,String outvalue);
功能：	识别
返回值：	String 类型的 xml 格式字符串或 json 格式字符串

doBulicenseRecon 方法的 4 个参数说明如下：

String username 用户白名单，需与服务器配置相同如“test”；

String imgbase64 营业执照图像文件 base64 编码；

String cardtype 营业执照类型编码（2008）；

String outvalue 营业执照输出字段(0 输出所有字段，1-18 选择输出，
输出多个字段用,隔开如：String outvalue=1,2,3;)。
输出对应字段见营业执照识别结果示例；

java 代码参考如下：

```
doBulicenseRecon(String username, String imgbase64, String cardtype,String outvalue);  
String result = cardImgreconAll.doBulicenseRecon("test", imgbase64, "2008", "0");
```

2.1.2.4 （发票）识别调用 webservice 接口说明：

服务地址：	http://IP:port/cxfServerX/ImgReconCard?wsdl 如:http://192.168.0.157:8080/cxfServerX/ImgReconCard?wsdl (具体地址请询问中安公司开发人员)
服务方法：	doVatinvoice(String username, String imgbase64, String cardtype,String outvalue);
功能：	识别
返回值：	String 类型的 xml 格式字符串或 json 格式字符串

doVatinvoice 方法的 4 个参数说明如下：

String username 用户白名单，需与服务器配置相同如“test”；

String imgbase64 增值税票图像文件 base64 编码；

String cardtype 增值税票类型编码（2009）；

String outvalue 发票输出字段(0 输出所有字段，选择输出字段用,隔开如：String outvalue=1,2,3; 发票自动分类字段全部返回不支持选择字段输出)。

java 代码参考如下：

```
doVatinvoice(String username, String imgbase64, String cardtype,String outvalue);  
String result = cardImgreconAll.doVatinvoice("test", imgbase64, "2009", "0");
```

2.1.3 移动客户端调用 webservice 识别服务接口

服务地址:	http://IP:port/cxfServerX/ImgReconCard?wsdl 如:http://127.0.0.1:8080/cxfServerX/ImgReconCard?wsdl (具体地址请询问中安未来公司开发人员)
服务方法:	@WebResult(name = "result") public String doAllCardReconforPhone(@WebParam(name = "username") String username, @WebParam(name = "paramdata") String paramdata, @WebParam(name = "signdata") String signdata, @WebParam(name = "imgtype") String imgtype);
功能:	识别
返回值:	识别结果, 参见下一小节。

其中:

doAllCardReconforPhone 方法的 4 个参数说明如下:

String username 用户白名单, 需与服务器配置相同如 “test”;

String paramdata 图像文件 base64 编码与其他参数拼接而成; 由开发包中的 **setRecgnPlainParam** 函数获得, 详情请参考代码示例。可调用 **setRecgnPlainParam** 方法得到;(**setRecgnPlainParam** 方法说明见 2.1.4)

String signdata 固定为 **null**; (注意:不带引号)

String imtype 为图片格式, 如 “jpg、bmp”;

java 代码参考如下:

```
// String doAllCardRecon(String username, String paramdata,  
String signdata, String imtype);  
String result = cardImgreconAll.doAllCardRecon("test", strsrc, null,"jpg");
```

2.1.4 辅助函数 setRecgnPlainParam 说明

String setRecgnPlainParam(String imgfilename, String type, String option, String password);

功能：构造识别请求数据

参数：

String imgfilename: 图片绝对路径及文件名；如 “D:/测试样本/test.jpg” ；

String type: 证件类型;二代证 2;行驶证 6;驾照 5;银行卡 17;车牌 19;如 “2” ；

String option: “”；（固定为带引号空字符串）

String password: null；（固定为 null，不带引号）

返回值：构造字符串

方法说明:

将图片转为二进制流，并进行 base64 编码后转为字符串类型，如下:

```
String imgBase64Str = Base64.GetBase64StrFromImage(imgfilename);
```

并在该字符串后追加字符串并返回，如下:

```
return imgBase64Str + "===###" + type + "===###" + option + "===###" + password;
```

如客户端使用 java 调用，可以直接引用 pki.jar 到项目依赖库中，并调用其中已经封装好的 CIPHERAdaptor 对象的 setRecgnPlainParam 方法即可。

如: CIPHERAdaptor clientAdaptor = new CIPHERAdaptor();

```
String strsrc = clientAdaptor.setRecgnPlainParam("D:/ 测试样本 /test.jpg",  
                                                "19","", null);
```

strsrc 即为图片加密后的串;

如不调用 pki.jar, 可以自行实现该方法, 详细实现 java 代码如下:

```
//接收识别参数返回 String 的字符串用于后续加密等操作
public String setRecgnPlainParam(String imgfilename, String type, String option,String
password){
    String imgBase64Str = Base64.GetBase64StrFromImage(imgfilename);
    return imgBase64Str + "===##" + type + "===##" + option + "===##" + password;    }

/*@param imagePath 传入的图片路径包括文件名
 * @return imgBase64Str 图片转化后的 Base64 编码 */
public static String GetBase64StrFromImage(String imagePath) {//将图片文件转化为字节数组
字符串, 并对其进行 Base64 编码处理
    String imgFile = imagePath;//待处理的图片
    InputStream in = null;
    byte[] data = null;
    try {//读取图片字节数组
        in = new FileInputStream(imgFile);
        data = new byte[in.available()];
        in.read(data);
        in.close();
    } catch (IOException e) {    e.printStackTrace();    }
    //对字节数组 Base64 编码
    BASE64Encoder encoder = new BASE64Encoder();
    return getOneLine(encoder.encode(data));//将 Base64 编码字节数组转为字符串返回
}

public static String getOneLine(String withr){
    StringReader sr = new StringReader(withr);
    BufferedReader br = new BufferedReader(sr);
    String line = null;
    StringBuffer temp = new StringBuffer();
    try{
        while((line=br.readLine())!=null){
            temp.append(line);
        }
        br.close();
        sr.close();
    }catch(Exception e){
        e.printStackTrace();
    }
    return temp.toString();
}
```

2.2 服务结果返回字符串格式

不论用户客户端是 PC 还是手机，返回的识别结果都是 XML 或 json 格式字符串。

2.2.1 识别结果字符串示例说明

1. 格式说明

参见下节示例，证件编码及字段编码详见附录一和二。

status 标签显示了识别状态，大于等于“0”代表识别成功，小于“0”代表识别失败，失败的原因在 value 中描述。

2. 识别结果 xml 示例：

下面是二代身份证识别结果示例：

```
<?xml version='1.0' encoding='UTF-8'?>
<data>
  <message>
    <status>2</status>
    <value>识别成功</value>
  </message>
  <cardsinfo>
    <card type="2">
      <item desc="保留"></item>
      <item desc="姓名">王青</item>
      <item desc="性别">女</item>
      <item desc="民族">汉</item>
      <item desc="出生">1976-08-09</item>
      <item desc="住址">北京市昌平区回龙观</item>
      <item desc="公民身份号码">XXXXXXXXXXXXXXXXXX</item>
      <item desc="头像">头像 BASE64 编码</item>
    </card>
  </cardsinfo>
</data>
```

下面是银行卡识别结果示例：

```
<?xml version='1.0' encoding='UTF-8'?>
<data>
  <message>
    <status>0</status>
    <value>识别成功</value>
```

```

</message>
<cardsinfo>
  <card type="17">
    <item desc="卡号"><![CDATA[6215673****53273]]></item>
    <item desc="银行卡类型"><![CDATA[借记卡]]></item>
    <item desc="银行卡名称"><![CDATA[京卡借记卡]]></item>
    <item desc="银行名称"><![CDATA[北京银行]]></item>
    <item desc="银行编号"><![CDATA[04031000]]></item>
  </card>
</cardsinfo>
</data>

```

下面是车牌识别结果示例：

```

<?xml version='1.0' encoding='UTF-8'?>
  <data>
    <message>
      <status>19</status>
      <value>识别成功</value>
    </message>
    <cardsinfo>
      <card type="19">
        <item desc="车牌号"><![CDATA[皖 E4**89]]></item>
        <item desc="车牌颜色"><![CDATA[蓝]]></item>
        <item desc="车牌颜色"><![CDATA[1]]></item>
        <item desc="车牌类型"><![CDATA[1]]></item>
        <item desc="整牌可信度"><![CDATA[80]]></item>
        <item desc="亮度评价"><![CDATA[0]]></item>
        <item desc="车牌运动方向"><![CDATA[0]]></item>
        <item desc="车牌位置
(left_top_right_bottom)"><![CDATA[265_370_394_404]]></item>
        <item desc="时间"><![CDATA[40]]></item>
      </card>
    </cardsinfo>
  </data>

```

下面是人证识别结果示例：

```

<?xml version='1.0' encoding='UTF-8'?>
  <data>
    <message>
      <status>0</status>
      <value>比对完成</value>
    </message>
    <cardsinfo>
      <card type="2">
        <item desc="判定值"><![CDATA[0.9635524]]></item>
        <item desc="判定结果"><![CDATA[是]]></item>

```

```

<item desc="保留"><![CDATA[]]></item>
<item desc="姓名"><![CDATA[王**]]></item>
<item desc="性别"><![CDATA[男]]></item>
<item desc="民族"><![CDATA[汉]]></item>
<item desc="出生"><![CDATA[1981-01-21]]></item>
<item desc="住址"><![CDATA[浙江省****]]></item>
<item desc="公民身份号码"><![CDATA[*****]]></item>
</card>
</cardsinfo>
</data>

```

下面是营业执照识别结果示例(选择输出字段 1-18 按顺序对应输出):

```
<?xml version='1.0' encoding='UTF-8'?>
```

```

<data>
  <message>
    <status>0</status>
    <value>识别完成</value>
  </message>
  <cardsinfo>
    <card type="2008">
      <item desc="统一社会信用代码"><![CDATA[****]]></item>
      <item desc="组织机构代码"><![CDATA[****]]></item>
      <item desc="税务登记证号"><![CDATA[****]]></item>
      <item desc="社保登记号"><![CDATA[****]]></item>
      <item desc="统计证证号"><![CDATA[****]]></item>
      <item desc="名称"><![CDATA[*****有限公司]]></item>
      <item desc="类型"><![CDATA[有限责任公司(自然人投资或控股)]]></item>
      <item desc="住所"><![CDATA[安徽省*****]]></item>
      <item desc="法定代表人"><![CDATA[**]]></item>
      <item desc="组成形式"><![CDATA[]]></item>
      <item desc="注册资本"><![CDATA[壹仟零捌拾捌万元人民币整]]></item>
      <item desc="成立日期"><![CDATA[2014 年 11 月 04 日]]></item>
      <item desc="营业期限"><![CDATA[2014 年 11 月 04 日至 2034 年 11 月 03 日]]></item>
      <item desc="经营范围"><![CDATA[研发施工**]]></item>
      <item desc="登记机关"><![CDATA[]]></item>
      <item desc="登记日期"><![CDATA[]]></item>
      <item desc="二维码"><![CDATA[]]></item>
      <item desc="副本"><![CDATA[倒裱]]></item>
    </card>
  </cardsinfo>
</data>

```

3. 识别结果 json 示例

下面是二代证正面识别结果示例：

```
{"data":{"message":{"status":2,"value":"识别完成"},"cardsinfo":{"card":{"item":[{"desc":"保留"}, {"content":"王青","desc":"姓名"}, {"content":"男","desc":"性别"}, {"content":"汉","desc":"民族"}, {"content":"1911-02-21","desc":"出生"}, {"content":"浙江省*****村**二组","desc":"住址"}, {"content":"33*****39","desc":"公民身份号码"}]}},"type":2}}}
```

下面是人证识别结果示例：

```
{"data":{"message":{"status":0,"value":"比对完成"}, "cardsinfo":{"card":{"item":[{"content":"0.9635524","desc":"判定值"}, {"content":"是","desc":"判定结果"}, {"content":"-1","desc":"证件 status 值"}, {"content":"识别失败!ReconClientX:imageReconx is null","desc":"证件 value 值"}]}},"type":2}}}
```

下面是营业执照识别结果示例:(选择输出字段 1-18 按顺序对应输出)

```
{"data":{"message":{"status":0,"value":"识别完成"},"cardsinfo":{"card":{"item":[{"content":"4**","desc":"统一社会信用代码"}, {"desc":"组织机构代码"}, {"desc":"税务登记证号"}, {"desc":"社保登记号"}, {"desc":"统计证证号"}, {"content":"安徽****有限公司","desc":"名称"}, {"content":"有限责任公司(自然人投资或控股)","desc":"类型"}, {"content":"安徽省***湖 11#30","desc":"住所"}, {"content":"***","desc":"法定代表人"}, {"desc":"组成形式"}, {"content":"壹仟零捌拾万元人民币整","desc":"注册资本"}, {"content":"2014 年 11 月 04 日","desc":"成立日期"}, {"content":"2014 年 11 月 04 日至 2024 年 11 月 03 日","desc":"营业期限"}, {"content":"建筑装饰**","desc":"经营范围"}, {"desc":"登记机关"}, {"desc":"登记日期"}, {"desc":"二维码"}, {"content":"***","desc":"副本"}]}},"type":2008}}}
```

2.3 示例程序

2.3.1 PC 客户端调用识别服务示例程序

详见 **cardWebserviceClient** 中的项目代码。

2.4 支持格式

产品	Windows 平台	Linux 平台
证件	JPG、BMP、PNG、TIF	JPG、BMP、PNG
银行卡	JPG、BMP、PNG、TIF	JPG、BMP、PNG
车牌	JPG、BMP、PNG、TIF	JPG、BMP
名片	JPG、BMP、PNG、TIF	JPG、BMP
人脸	JPG、BMP、PNG、TIF	JPG、BMP、PNG、TIF
发票	JPG、PNG	JPG、PNG
营业执照	JPG、PNG	JPG、PNG
VIN	JPG、PNG、BMP	JPG、BMP

第三章 核验服务接口调用

本章将对核验接口进行阐述。由于实际所使用到的编程语言较多，此文档不会一一涉及。本章最后小节将只就 [Java 语言如何调用识别服务接口](#) 进行代码示例讲解。其它语言如何进行调用，用户可参照接口调用例程。

3.1 核验服务接口说明

使用识别服务前请确认服务器 ip 地址及端口号

3.1.1 客户端调用核验服务 http 接口

3.1.1.1 身份证号核验

http 方法:	必须为 POST
url:	http://IP:Port/cxfServerX/DoVerIdenNo 如:http://192.168.0.54:8888/cxfServerX/DoVerIdenNo
Post 实体内容:	Post 实体内容仅为一个 json 对象。 json 对象中封装了 3 对参数, 详情见下文
返回值	String 类型的 xml 字符串或 json 字符串

Post 实体内容: 客户端将 2 对 String 类型的参数, 以键值对的形式, 封装到一个 json 对象中。2 个参数如下:

String trueName 姓名
String idenNo 身份证号

将上述 2 对参数封装到 json 对象中, java 代码参考如下:

```
JSONObject json = new JSONObject();//封装json对象
String trueName="李XX";
String idenNo="13053XXXXXXXXXXXXX";
json.put("trueName", trueName);
json.put("idenNo", idenNo);
```

3.1.1.2 身份证核验

http 方法:	必须为 POST
url:	http://IP:Port/cxfServerX/DoVerIdCard 如:http://192.168.0.54:8888/cxfServerX/DoVerIdCard
Post 实体内容:	Post 实体内容仅为一个 json 对象。 json 对象中封装了 3 对参数, 详情见下文
返回值	String 类型的 xml 字符串或 json 字符串

Post 实体内容: 客户端将 3 对 String 类型的参数, 以键值对的形式, 封装到一个 json 对象中。3 个参数如下:

String trueName 姓名
String idenNo 身份证号
String img 身份证图片 (※Base64.NO_WRAP)

将上述 3 对参数封装到 json 对象中, java 代码参考如下:

```
JSONObject json = new JSONObject();//封装json对象
String trueName="李XX";
String idenNo="13053XXXXXXXXXXXXX";
String img = "/9j/4AAQSkZJRgABAQEAY.....ABgAAD/2wBDAA;
json.put("trueName", trueName);
json.put("idenNo", idenNo);
json.put("img", img);
```


3.1.1.3 身份核验

http 方法:	必须为 POST
url:	http://IP:Port/cxfServerX/DoVerIden 如:http://192.168.0.54:8888/cxfServerX/DoVerIden
Post 实体内容:	Post 实体内容仅为一个 json 对象。 json 对象中封装了 3 对参数, 详情见下文
返回值	String 类型的 xml 字符串或 json 字符串

Post 实体内容: 客户端将 3 对 String 类型的参数, 以键值对的形式, 封装到一个 json 对象中。3 个参数如下:

String trueName 姓名
String idenNo 身份证号
String img 场景图片 (※Base64. NO_WRAP)

将上述 3 对参数封装到 json 对象中, java 代码参考如下:

```
JSONObject json = new JSONObject();//封装json对象
String trueName="李XX";
String idenNo="13053XXXXXXXXXXXXX";
json.put("trueName", trueName);
json.put("idenNo", idenNo);
json.put("img", img);
```

3.1.1.4 手机号实名核验

http 方法:	必须为 POST
url:	http://IP:Port/cxfServerX/DoVerTel 如:http://192.168.0.54:8888/cxfServerX/DoVerTel
Post 实体内容:	Post 实体内容仅为一个 json 对象。 json 对象中封装了 3 对参数, 详情见下文
返回值	String 类型的 xml 字符串或 json 字符串

Post 实体内容: 客户端将 3 对 String 类型的参数, 以键值对的形式, 封装到一个 json 对象中。3 个参数如下:

String tel 手机号
String trueName 姓名
String idenNo 身份证号

将上述 3 对参数封装到 json 对象中, java 代码参考如下:

```
JSONObject json = new JSONObject();//封装json对象
String tel="137*****";
String trueName="李XX";
String idenNo="13053XXXXXXXXXXXXX";
json.put("tel", tel);
json.put("trueName", trueName);
json.put("idenNo", idenNo);
```

3.1.1.5 银行卡实名认证

http 方法:	必须为 POST
url:	http://IP:Port/cxfServerX/DoVerBank 如:http://192.168.0.54:8888/cxfServerX/DoVerBank
Post 实体内容:	Post 实体内容仅为一个 json 对象。 json 对象中封装了 4 对参数, 详情见下文
返回值	String 类型的 xml 字符串或 json 字符串

Post 实体内容: 客户端将 4 对 String 类型的参数, 以键值对的形式, 封装到一个 json 对象中。4 个参数如下:

String bankNo 银行卡号
String trueName 姓名
String idenNo 身份证号
String tel 手机号

将上述 4 对参数封装到 json 对象中, java 代码参考如下:

```
JSONObject json = new JSONObject();//封装json对象
String bankNo="6217*****";
String tel="137*****";
String trueName="李XX";
String idenNo="13053XXXXXXXXXXXX";
json.put("tel", tel);
json.put("trueName", trueName);
json.put("idenNo", idenNo);
json.put("bankNo", bankNo);
```

3.1.1.6 发票核验

http 方法:	必须为 POST
url:	http://IP:Port/cxfServerX/DoverInvoice 如:http://192.168.0.54:8888/cxfServerX/DoverInvoice
Post 实体内容:	Post 实体内容仅为一个 json 对象。 json 对象中封装了 4 对参数, 详情见下文
返回值	String 类型的 xml 字符串或 json 字符串

String invoiceCode 发票代码
String invoiceNumber 发票号码
String billingDate 开票日期 YYYY-MM-DD
String totalAmount 合计金额 (不含税), 必须精确到两位小数 (专票、货运专票、
 机动车专票必填)
String checkCode 校验码后 6 位 (普票、电子普票、卷式普票必填)

将上述 5 对参数封装到 json 对象中, java 代码参考如下:

```
JSONObject json = new JSONObject();//封装json对象
String invoiceCode="*****";
String invoiceNumber="*****";
String  billingDate="2017-07-20";
String  totalAmount="61538.46";
String  checkCode="*****";
json.put("invoiceCode", invoiceCode);
json.put("invoiceNumber", invoiceNumber);
json.put("billingDate", billingDate);
json.put("totalAmount", totalAmount);
json.put("checkCode", checkCode);
```

3.1.2 客户端调用核验服务 WebService 接口

3.1.2.1 身份证号核验接口说明：

服务地址：	http://IP:port/cxfServerX/ImgReconCard?wsdl 如:http://192.168.0.157:8080/cxfServerX/ImgReconCard?wsdl (具体地址请问中安公司开发人员)
服务方法：	doveridenNo(String trueName,String idenNo);
功能：	身份证号核验
返回值：	String 类型的 xml 格式字符串或 json 格式字符串

doveridenNo 方法的 2 个参数说明如下：

String trueName	姓名
String idenNo	身份证号

java 代码参考如下：

```
doveridenNo(String trueName,String idenNo);
String result = cardImgreconAll.doveridenNo(name, idno);
```

3.1.2.2 身份证核验接口说明：

服务地址：	http://IP:port/cxfServerX/ImgReconCard?wsdl 如:http://192.168.0.157:8080/cxfServerX/ImgReconCard?wsdl (具体地址请问中安公司开发人员)
服务方法：	doveridcard(String trueName, String idenNo, String img);
功能：	身份证核验
返回值：	String 类型的 xml 格式字符串或 json 格式字符串

doveridcard 方法的 3 个参数说明如下：

String trueName	姓名
String idenNo	身份证号
String img	身份证图像 (※Base64. NO_WRAP)

java 代码参考如下：

```
doveridcard(String trueName, String idenNo, String img);
String result = cardImgreconAll.doveridcard(trueName, idenNo, img);
```

3.1.2.3 身份核验接口说明:

服务地址:	http://IP:port/cxfServerX/ImgReconCard?wsdl 如:http://192.168.0.157:8080/cxfServerX/ImgReconCard?wsdl (具体地址请问中安公司开发人员)
服务方法:	doveriden(String trueName, String idenNo, String img);
功能:	身份核验
返回值:	String 类型的 xml 格式字符串或 json 格式字符串

doveriden 方法的 3 个参数说明如下:

String trueName	姓名
String idenNo	身份证号
String img	场景照片 (※Base64. NO_WRAP)

Java 代码参考如下:

```
doveriden(String trueName, String idenNo, String img);  
String result = cardImgreconAll.doveriden(trueName, idenNo, img);
```

3.1.2.4 手机号实名核验接口说明:

服务地址:	http://IP:port/cxfServerX/ImgReconCard?wsdl 如:http://192.168.0.157:8080/cxfServerX/ImgReconCard?wsdl (具体地址请问中安公司开发人员)
服务方法:	dovortel(String tel, String trueName, String idenNo);
功能:	手机号实名核验
返回值:	String 类型的 xml 格式字符串或 json 格式字符串

dovortel 方法的 3 个参数说明如下:

String tel	手机号
String trueName	姓名
String idenNo	身份证号

Java 代码参考如下:

```
dovortel(String tel, String trueName, String idenNo);  
String result = cardImgreconAll.dovortel(tel, trueName, idenNo);
```

3.1.2.5 银行卡实名核验接口说明：

服务地址：	http://IP:port/cxfServerX/ImgReconCard?wsdl 如 : http://192.168.0.157:8080/cxfServerX/ImgReconCard?wsdl (具体地址请问中安公司开发人员)
服务方法：	doverbank(String bankNo,String trueName,String idenNo,String tel);
功能：	银行卡实名核验
返回值：	String 类型的 xml 格式字符串或 json 格式字符串

doverbank 方法的 4 个参数说明如下：

String bankNo	手机号
String trueName	姓名
String idenNo	身份证号
String tel	手机号

Java 代码参考如下：

```
doverbank(String bankNo,String trueName,String idenNo,String tel);
String result = cardImgreconAll.doverbank(bankNo, trueName, idenNo, tel);
```

3.1.2.6 发票核验接口说明：

服务地址：	http://IP:port/cxfServerX/ImgReconCard?wsdl 如: http://192.168.0.157:8080/cxfServerX/ImgReconCard?wsdl (具体地址请问中安公司开发人员)
服务方法：	doverinvoice(String invoiceCode, String invoiceNumber, String billingDate, String totalAmount, String checkCode)
功能：	发票核验
返回值：	String 类型的 xml 格式字符串或 json 格式字符串

doverinvoice 方法的 5 个参数说明如下：

String invoiceCode	发票代码
String invoiceNumber	发票号码
String billingDate	开票日期 YYYY-MM-DD
String totalAmount	合计金额（不含税），必须精确到两位小数（专票、货运专票、机动车专票必填）
String checkCode	校验码后 6 位（普票、电子普票、卷式普票必填）

Java 代码参考如下：

```
doverinvoice(String invoiceCode, String invoiceNumber, String billingDate, String totalAmount, String checkCode);
```

```
String result = cardImgreconAll.doverbank(invoiceCode , invoiceNumber , billingDate ,  
totalAmount, checkCode );
```

3.2 支持格式

核验服务支持 Jpg、Jpeg、png 格式图片识别

第四章 注意事项

4.1 服务需要连网调用

由于识别服务基于 http 及 Webservice 技术实现，用户调用服务前请确保您的设备可以连通识别服务器，并保证一定的带宽。否则由于识别服务的图像数据网络传输较慢，会引起较长时间等待，**建议图像文件大小在 300KB 左右**。

4.2 对识别图像的要求

4.2.1 一般要求

图像文字清晰，人眼可辨别；

对比度、亮度适中；

尽量不要有背景，即证件充满图片，如果不能保证没有背景，请选择纯黑背景而且背景不宜过多。

4.2.2 对扫描图像的要求

扫描时选择分辨率 300dpi，不能低于 150 dpi，分辨率最好不要过高。

4.2.3 对拍摄图像的要求

拍摄时注意光照的影响，尽量避免反光和黑影，尤其是证件有薄膜覆盖的时候；

拍摄时做好聚焦，以免图像模糊不清；

拍摄时将证件放正；

尽量不要使拍摄角度倾斜过大，以免造成图像变形；

拍摄时请不要选择过高的分辨率拍摄，一方面造成图片过大，不利于传输，一方面分辨率过高也会影响识别率。拍摄时可以选择 1280*960 分辨率拍摄，低一些的分辨率也可以选择，但是要确保图像文字清晰。

附录一：产品类型编码

证件产品			
证件名称			证件 MAINID（10进制表示）
二代身份证正面			2
二代身份证证背面			3
二代身份证正面背面自动分类			101
临时身份证			4
驾照			5
行驶证			6
军官证			7
士兵证（暂不支持）			8
机读证件	护照幅面	中华人民共和国往来港澳通行证	9
		台湾居民往来大陆通行证	10
		大陆居民往来台湾通行证	11
		签证	12
		护照	13
	卡幅面	港澳居民来往内地通行证正面	14
		港澳居民来往内地通行证背面	15
户口本			16

居住证	1000
香港永久性居民身份证	1001
登机牌（拍照设备目前不支持登机牌的识别）	1002
边民证（A）（照片页）	1003
边民证（B）（个人信息页）	1004
澳门身份证	1005
领取凭证(AVA6 支持)	1006
律师证（A）（信息页）	1007
律师证（B）（照片页）	1008
组织机构代码证	1011
深圳居住证	1013
全民健康保险卡	1030
台湾身份证正面	1031
台湾身份证背面	1032
厦门社会保障卡-照片页	1039
福建社会保障卡-照片页	1041
卡式港澳台通行证	22
新版台湾居民往来大陆通行证正面	25
新版台湾居民往来大陆通行证背面	26
中国驾驶证副页	28
马来西亚身份证	2001
美国加利福尼亚驾驶证	2002
新西兰驾驶证	2003
新加坡身份证	2004
印度尼西亚居民身份证	2010
泰国国民身份证	2011
瑞典驾驶证	2020

银行卡产品	
银行卡	17
车牌产品	
车牌	19
名片产品	
名片	20
人脸产品	
人脸	21
文档产品	
文档+简体中文	1991
文档+繁体	2991
文档+纯英文	3991
长微博+简体中文	1992
长微博+繁体	2992
长微博+纯英文	3992
营业执照产品	
营业执照	2008
发票产品	
增值税发票	2009
火车票	20092
机动车发票	20094
发票自动分类	20090
VIN 产品	
VIN	2007

技术支持

电话：010—62800056 转 157 /杨敏/李朝需/杨旭