**人证核验设备接口**

**规范说明书**

**（V2.0）**

**北京大河茂源科技有限公司**

**2020年9月**

**版本历史**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **作者** | **日期** | **说明** |
| V1.0 | 付贵阳 | 2018/1/23 |  |
| V1.0.315 | 郑立 | 2018/3/15 | 增加了mode字段。 |
| V1.0.316 | 付贵阳 | 2018/3/16 | 修改备注信息 |
| V1.0.319 | 付贵阳 | 2018/3/19 | 增加数据项类型和签名说明 |
| V1.0.1224 | 付贵阳 | 2018/12/24 | 增加实名信息采集方式及说明 |
| V1.0.1321 | 付贵阳 | 2019/1/21 | 修改备注及说明信息 |
| V1.0.2088 | 董国杰 | 2019/8/8 | 增加证件类型、英文姓名、换证次数数据项 |
| V2.0.408 | 张帆 | 2020/4/8 | 大版本更新，重新整理文档。不兼容旧版本，不影响已上线设备运行。 |
| V2.0.421 | 张帆 | 2020/4/21 | 新增外部设备被动接收命令接口内容的说明 |
| V2.0.911 | 付贵阳 | 2020/9/11 | 增加第七章示例数据 |
|  |  |  |  |
|  |  |  |  |

目录

[第一章 概述 5](#_Toc50730143)

[1、API使用对象 5](#_Toc50730144)

[2、支持的平台 5](#_Toc50730145)

[3、适用范围 5](#_Toc50730146)

[第二章 调用WebApi命令 6](#_Toc50730147)

[1、术语表 6](#_Toc50730148)

[2、调用WebApi命令 7](#_Toc50730149)

[3、相关命令 8](#_Toc50730150)

[第三章 WebApi命令 9](#_Toc50730151)

[3.1 身份证人像核验服务 9](#_Toc50730152)

[3.2 身份证指纹核验服务 13](#_Toc50730153)

[3.3 未带证核验服务 18](#_Toc50730154)

[第四章 外部设备被动接收命令接口 21](#_Toc50730155)

[4.1、接口说明 21](#_Toc50730156)

[4.2、外部设备发送“心跳“信息格式 21](#_Toc50730157)

[4.3、旅业系统向外部设备发送任务请求的数据格式 22](#_Toc50730158)

[第五章 签名说明 25](#_Toc50730159)

[5.1、MD5签名函数（C#示例） 25](#_Toc50730160)

[5.2、加密（C#示例） 26](#_Toc50730161)

[第六章 错误码说明 29](#_Toc50730162)

[6.1、公共错误码说明 29](#_Toc50730163)

[6.2、身份证人像核验服务错误码说明 30](#_Toc50730164)

[6.3、身份证指纹核验服务错误码说明 30](#_Toc50730165)

[6.4、未带证核验服务错误码说明 30](#_Toc50730166)

[第七章 示例数据格式 31](#_Toc50730167)

[7.1、身份证人像核验数据示例 31](#_Toc50730168)

[7.2、身份证指纹核验数据示例 32](#_Toc50730169)

[7.3、未带证核验数据示例 33](#_Toc50730170)

# 第一章 概述

本接口规范说明书主要用于明确“宾馆酒店旅客登记软件”和“人证核验设备”数据交换方式、数据格式和参数的相关说明，为开发人员提供开发指导。

## 1、API使用对象

所有被允许接入的接口服务商（一般为软件开发公司或合作公司），均可以拥有一个接口服务账号，利用该账号，访问WebApi接口。

## 2、支持的平台

采用WebApi提供接口服务器，对各种平台均有很好的支持，如：.NET、Java、Javascript（ajax）等。

## 3、适用范围

本接口规范说明书主要编写人证核验设备对接的WebApi服务接口，包括接口格式、接口功能和接口参数等，适用于通过http/https协议发起调用请求、POST提交数据，统一采用UTF-8编码规则、JSON字符串格式提交，提交结果使用JSON字符串格式返回。

# 第二章 调用WebApi命令

## 1、术语表

|  |  |
| --- | --- |
| **术语** | **说明** |
| account | 接口服务账号名称，简称**“服务账号”**，由平台服务商提供。 |
| client | **“单位编码”**，每个企业（旅馆）都拥有唯一的单位编码，如：“4108831987”, 由平台服务商提供。 |
| cmd | WebApi**“接口命令”**，标识提供的服务功能。接口命令不区分大小写。 |
| data | WebApi接口命令提交的**“接口数据”**，为json字符串，具体根据命令的约定。 |
| nonce | **“随机数”**，接口调用方自行产生，可使用时间、GUID或随机发生器。 |
| timestamp | **“时间戳”**，格式为yyyyMMddHHmmss，即年年年年月月日日时时分分秒秒，共14个字符，如：“20200226121212”。 |
| signKey | **“签名密钥”**，由接口调用方与服务平台约定，接口调用方可自行维护。 |
| signatrue | 接口**“签名信息”**，信息项cmd + account + nonce + client + timestamp + data + signKey的MD5签名。 |
| aes | 标识接口调用方提交的接口数据data是否为密文，取值为“true”/“false”。不传该参数或参数值为“false”表示接口调用提交的接口数据data为明文。 |
| aesKey | 接口数据data的“密文密钥”，由接口调用方与服务平台约定，接口调用方可自行维护。 |

## 2、调用WebApi命令

**请求方式：**POST

**请求地址：**https://服务器地址[:端口号]/exchange/data 或

http://服务器地址[:端口号]/exchange/data

注：各省服务平台地址均不相同，部署时请咨询项目技术人员。

**POST数据：** 为json字符串

{

account: "…" , // 服务账号名

client: " 4108831987", // 单位编码

cmd: " FaceVerify ", // 接口命令字

data: "......", // 接口数据，根据aes判断是否加密，不加密时为json字符串，具体根据命令的约定，加密时为加密后的字符串

nonce: "34583453423", // 随机数

timestamp: "20170831121212",// 时间戳，发送数据时的14位时间戳，格式：yyyyMMddHHmmss

signatrue: "......", // 信息签名，对cmd+ account + nonce + client + timestamp + data + signKey进行MD5的加密信息

aes: "false" // 数据是否加密，缺省为“false”

}

**调用成功：**返回json字符串，如下所示：

{

errCode: 0, // 错误码，为0值表示执行正确

errMsg: "ok",

… // 其他信息，具体参见各个命令的约定

}

**调用失败：**调用失败或异常，将返回json字符串，如下所示：

{

errCode: 0, // 错误码，为非0值表示执行有错误

errMsg: "......" // 错误信息描述字符串

}

## 3、相关命令

|  |  |  |
| --- | --- | --- |
| **命令（cmd）** | **调用类型** | **说明** |
| FaceVerify |  | 通过人证核验设备以人脸核验方式汇集人员信息 |
| FPVerify |  | 通过人证核验设备以指纹核验方式汇集人员信息 |
| NoPaper |  | 人员未携带身份证件信息,通过人证核验设备采集姓名、身份证号、现场照片，调用此接口命令，接口服务反馈人证核验结果 |
| Heartbeat |  | 设备向接口发送心跳命令，以便接口能确定设备状态 |
| TaskOver |  | 接口向设备发送请求命令，设备根据命令执行对应操作 |

注：调用类型值为“实时”或者为空。当为“实时”的命令需要在客户端执行完对应的操作后及时调用，并需要处理上传失败等异常情况，否则会导致数据的延时、不一致甚至丢失；其他类型为按需调用。

# 第三章 WebApi命令

## 3.1 身份证人像核验服务

**应用场景：**通过人证核验设备使用人像核验服务采集到人员信息进行汇集存储。

**cmd：FaceVerify**

**data说明：**data为json字符串，对应的json对象定义如下：

{

name: "", // 姓名   
enname: "", // 英文名  
paperType: "", // 证件类型  
number: "", // 证件号码  
sex: "", // 性别  
folk: "", // 民族  
country: "", // 国家/地区  
address: "", // 详细地址  
birthday: "", // 出生日期  
signOrgan: "", // 签发机关  
valid: "", // 有效期限

idCard: "", // 身份证件物理卡号

passNumber : "", // 通行证号码

changeNum: "", // 换证次数

headImage: "", // 居民身份证照片，base64字符串   
snapshot: "", // 现场抓拍人脸图像, base64字符串  
similarity: "", // 比对相似度

isPass: "true", // 核验结果  
fpFeatrue1: "", // 指纹特征1  
fpFeatrue2: "", // 指纹特征2  
sceneFP1: "", // 现场指纹特征1

sceneFP2: "", // 现场指纹特征2  
fpImage1: "", // 现场指纹图像1

fpImage2: "", // 现场指纹图像2

deviceSN"" // 设备序列号

}

**数据项**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **必须** | **说明** | **备注** |
| name | 是 | 姓名 | 例: "张三" |
| enName | 否 | 英文姓名 | 格式：英文名,英文姓  paperType值为12时，必填 |
| paperType | 是 | 证件类型 | 1-二代身份证、  12-外国人永久居留身份证  13-港澳居民居住证、  14-台湾居民居住证、 |
| number | 是 | 证件号码 |  |
| sex | 是 | 性别 | 对应代码种类：性别(gb2261)  可以传入：“男”、“男性”、“女”、“女性”等。 |
| folk | 否 | 民族 | 对应代码种类：民族(gb3304)  也可以为民族的名称  例："汉族"、"汉"等，用身份证读取到的民族信息即可。  paperType值为1时，必填 |
| birthday | 是 | 出生日期 | 例："20100509" |
| address | 是 | 详细地址 |  |
| country | 否 | 国家/地区 | paperType值为12时,必填 例: "CHN"代表中国，  "ITA"代表意大利 |
| signOrgan | 否 | 签发机关 | 例："北京市公安局西城分局" |
| valid | 否 | 证件有效期限 | 例：  "2007.02.26-2017.02.26"  长期有效为：  "2014.07.09-长期" |
| idCard | 是 | 身份证的物理卡号 |  |
| passNumber | 否 | 通行证号码 | paperType值为13、14时,必填 |
| changeNum | 否 | 换证次数 | paperType值 为12、13、14必填，为12时，填写证件版本号 |
| headImage | 是 | 居民身份证照片 | Base64字符串 |
| snapshot | 是 | 现场人脸图像 | Base64字符串 |
| similarity | 是 | 人像比对相似度 | （0~1），大于0小于1 |
| isPass | 是 | 人证核验是否通过 | 值为：true/false |
| fpFeatrue1 | 否 | 身份证信息中的指纹特征1 | Base64字符串  如读取的是三代身份证，则指纹信息必填 |
| fpFeatrue2 | 否 | 身份证信息中的指纹特征2 | Base64字符串  如读取的是三代身份证，则指纹信息必填 |
| sceneFP1 | 否 | 现场核验指纹特征1 | Base64字符串  如需采集指纹信息，必填 |
| sceneFP2 | 否 | 现场核验指纹特征2 | Base64字符串  如需采集指纹信息，必填 |
| fpImage1 | 否 | 现场核验指纹图像 | Base64字符串  如保存有采集指纹图像，必填 |
| fpImage1 | 否 | 现场核验指纹图像 | Base64字符串  如保存有采集指纹图像，必填 |
| deviceSN | 是 | 设备序列号 | 设备唯一序列号。如果没有独立的设备序列号，如有身份证SAM模块，可以使用SAM编号。 |

**返回结果**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **必须** | **说明** | **备注** |
| errCode | 是 | 返回结果代码 |  |
| errMsg | 是 | 返回结果描述 |  |
| playVoice | 否 | 设备是否进行“语音播放” | true/false |
| voiceText | 否 | 接口返回需设备播放的语音文字内容 | playVoice值为“true”时，必填 |

**返回结果：**成功时返回以下json内容：

**{**

errCode: 0, // 结果码,0表示成功

errMsg: "成功", // 状态描述信息，errCode不为0时表示失败的原因

playVoice：true， // 设备是否播放语音

voiceText："……"， // 语音文字内容

**}**

## 3.2 身份证指纹核验服务

**应用场景：**通过人证核验设备使用指纹核验服务采集到人员信息进行汇集存储。

**cmd：FPVerify**

**data说明：**data为json字符串，对应的json对象定义如下：

{

name: "", // 姓名   
enName: "", // 英文名  
paperType: "", // 证件类型  
number: "", // 证件号码  
sex: "", // 性别

folk: "", // 民族  
birthday: "", // 出生日期  
address: "", // 详细地址  
country: "", // 国家/地区  
signOrgan: "", // 签发机关  
valid: "", // 有效期限

idCard: "", // 身份证件物理卡号

passNumber : "", // 通行证号码

changeNum: "", // 换证次数

headImage: "", // 居民身份证照片，base64字符串   
snapshot: "", // 现场抓拍人脸图像, base64字符串

similarity: "", // 比对相似度

isPass: "true", // 核验结果  
fpFeatrue1: "", // 指纹特征1  
fpFeatrue2: "", // 指纹特征2  
sceneFP1: "", // 现场指纹特征  
fpImage1: "", // 现场指纹图像

deviceSN"" // 设备序列号

}

**数据项**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **必须** | **说明** | **备注** |
| name | 是 | 姓名 | 例: "张三" |
| enName | 否 | 英文姓名 | 格式：英文名,英文姓  PaperType值为12时，必填 |
| paperType | 是 | 证件类型 | 1-二代身份证、  12-外国人永久居留证  13-港澳居民居住证、  14-台湾居民居住证、 |
| number | 是 | 证件号码 |  |
| sex | 是 | 性别 | 对应代码种类：性别(gb2261)  可以传入：“男”、“男性”、“女”、“女性”等。 |
| folk | 否 | 民族 | 对应代码种类：民族(gb3304)  也可以为民族的名称  例："汉族"、"汉"等，用身份证读取到的民族信息即可。  PaperType值为1时，必填 |
| birthday | 是 | 出生日期 | 例："20100509" |
| address | 是 | 详细地址 |  |
| country | 否 | 国家/地区代码（3位） | PaperType值为12时,必填 例: "CHN"代表中国，  "ITA"代表意大利 |
| signOrgan | 否 | 签发机关 | 例："北京市公安局西城分局" |
| valid | 否 | 证件有效期限 | 例：  "2007.02.26-2017.02.26"  长期有效为：  "2014.07.09-长期" |
| idCard | 是 | 身份证的物理卡号 |  |
| passNumber | 否 | 通行证号码 | PaperType值 为13、14必填 |
| changeNum | 否 | 换证次数 | PaperType值 为12、13、14必填，为12时，填写证件版本号 |
| headImage | 是 | 居民身份证照片 | Base64字符串 |
| snapshot | 否 | 现场人脸图像 | Base64字符串 |
| similarity | 是 | 指纹比对相似度 | （0~1），大于0小于1 |
| isPass | 是 | 人证核验是否通过 | 值为：true/false |
| fpFeatrue1 | 是 | 身份证信息中的指纹特征1 | Base64字符串  读取身份证中的  指纹特征1 |
| fpFeatrue2 | 是 | 身份证信息中的指纹特征2 | Base64字符串  读取身份证中的  指纹特征2 |
| sceneFP1 | 是 | 现场核验指纹特征1 | Base64字符串  现场核验时，录入的指纹特征，必填 |
| fpImage1 | 否 | 现场核验指纹图像1 | Base64字符串  如保存有现场核验指纹图片，必填 |
| deviceSN | 是 | 设备序列号 | 设备唯一序列号。如果没有独立的设备序列号，如有身份证SAM模块，可以使用SAM编号。 |

**返回结果**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **必须** | **说明** | **备注** |
| errCode | 是 | 返回结果代码 |  |
| errMsg | 是 | 返回结果描述 |  |
| playVoice | 否 | 设备是否进行“语音播放” | true/false |
| voiceText | 否 | 接口返回需设备播放的语音文字内容 | playVoice值为“true”时，必填 |

**返回结果：**成功时返回以下json内容：

**{**

errCode: 0, // 结果码,0表示成功

errMsg: "成功", // 状态描述信息，errCode不为0时表示失败的原因

playVoice：true， // 设备是否播放语音

voiceText："……"， // 语音文字内容

**}**

## 3.3 未带证核验服务

**应用场景：**在人员未携带证件的情况下，通过人证核验设备采集人员的姓名、身份证号和现场人员人脸图像等信息，调用该接口服务。调用该接口服务后，接口服务进行可信身份认证，并进行后续业务的处理。

**cmd：NoPaper**

**data说明：**data为json字符串，对应的json对象定义如下：

{

name: "", // 姓名   
number: "", // 身份证号

folk: "", // 民族  
address: "", // 详细地址

snapshot: "", // 现场抓拍人脸图像, base64字符串

sceneFP1: "", // 现场指纹特征1

sceneFP2: "", // 现场指纹特征2  
fpImage1: "", // 现场指纹图像1

fpImage2: "", // 现场指纹图像2

deviceSN"" // 设备序列号

}

**数据项**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **必须** | **说明** | **备注** |
| name | 是 | 姓名 | 例: "张三" |
| number | 是 | 身份证号 |  |
| folk | 否 | 民族 | 对应代码种类：民族(gb3304)  也可以为民族的名称，例："汉族"、"汉"等 |
| address | 否 | 详细地址 |  |
| snapshot | 是 | 现场人脸图像 | Base64字符串 |
| sceneFP1 | 否 | 现场核验指纹特征1 | Base64字符串  如需采集指纹信息，必填 |
| sceneFP2 | 否 | 现场核验指纹特征2 | Base64字符串  如需采集指纹信息，必填 |
| fpImage1 | 否 | 现场核验指纹图像 | Base64字符串  如保存有采集指纹图像，必填 |
| fpImage1 | 否 | 现场核验指纹图像 | Base64字符串  如保存有采集指纹图像，必填 |
| deviceSN | 是 | 设备序列号 | 设备唯一序列号。如果没有独立的设备序列号，如有身份证SAM模块，可以使用SAM编号。 |

**返回结果**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **必须** | **说明** | **备注** |
| errCode | 是 | 返回结果代码 |  |
| errMsg | 是 | 返回结果描述 |  |
| playVoice | 否 | 设备是否进行“语音播放” | true/false |
| voiceText | 否 | 接口返回需设备播放的语音文字内容 | playVoice值为“true”时，必填 |

**返回结果：**成功时返回以下json内容：

**{**

errCode: 0, // 0表示认证成功，1-表示认证不符，其他表示失败,-999表示接口调用失败

errMsg: "ok", // errCode不为0时表示失败的原因

playVoice：true， // 设备是否播放语音

voiceText："……"， // 语音文字内容

**}**

# 第四章 外部设备被动接收命令接口

## 4.1、接口说明

外部设备利用http协议，建立一个WebApi数据接收服务，被动接收旅馆业终端软件发起的任务请求（以POST方式请求）。如下图所示：



## 4.2、外部设备发送“心跳“信息格式

外部设备每隔一段时间（建议30秒钟）向旅馆业终端系统发送一次json数据（“心跳”数据），一是能够让旅业系统知道外部人证核验设备在线，二是及时通知旅业系统外部设备的WebApi服务地址，“心跳”数据格式如下：

**cmd：Heartbeat**

**data说明：**data为json字符串，对应的json对象定义如下：

{

webapi: "……", // 外部设备的WebApi地址

}

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **必须** | **说明** | **备注** |
| webapi | 是 | 外部设备的WebApi服务地址 | 如：”http://192.168.0.66:1234/task” |

**返回结果**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **必须** | **说明** | **备注** |
| errCode | 是 | 返回结果代码 |  |
| errMsg | 是 | 返回结果描述 |  |

旅业系统接收到该信息后将返回如下json字符串：

{

errCode: ”0”, // 结果码，0表示成功

errMsg:”成功”, // 如果code不等于”0”时，为错误原因

}

## 4.3、旅业系统向外部设备发送任务请求的数据格式

旅业系统在需要外部设备做一些任务（如拍照）时，利用HTTP协议的POST方式，向外部设备的WebApi服务地址发送任务请求，外部设备接收到该请求后，执行请求的任务（如拍照），然后返回任务执行结果。旅业任务请求数据格式如下：

**data说明：**data为json字符串，对应的json对象定义如下：

{

taskid: "23445345fdfdf56fdf5", //任务内部标识

tasktype: "1001", // 任务类型，”1001”—请求设备拍照

taskparam：””

}

注：数据项类型均为字符型

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **必须** | **说明** | **备注** |
| taskid | 是 | 任务内部标识，每次任务都有一个不同的值 |  |
| tasktype | 是 | 任务类型，目前支持：  ”1001” - 请求设备拍照  “1002” – 请求设备拍照并比对 |  |
| taskparam | 否 | 任务参数 |  |

**返回结果**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数** | **必须** | **说明** | **备注** |
| taskid | 是 | 返回执行任务对应的taskid |  |
| tasktype | 是 | 任务类型，目前支持：  ”1001”—请求设备拍照 |  |
| taskCode | 是 | 任务执行结果码 | ”0”表示成功，其他表示错误 |
| taskMsg | 是 | 任务执行结果描述 | 任务执行失败时，返回执行失败的原因 |
| taskdata | 是 | 任务执行数据 | 根据任务类型不同有不同的约定。”1001”—返回拍照的图片的base64字符串 |

外部设备执行完对应的任务后，向旅业端发送任务结果，其json字符串的对象如下所示：

**cmd：TaskOver**

**data说明：**data为json字符串，对应的json对象定义如下：

{

taskid: ”23445345fdfdf56fdf5”, // 执行的任务id

tasktype: ”1001”, // 任务类型，”1001”—请求设备拍照

taskCode: ”0”, // 任务执行结果码，”0”表示成功，其他表示错误

taskMsg: ”ok”, // 任务执行失败时，返回执行失败的原因

taskdata:”…” // 任务数据，根据任务类型不同有不同的约定。”1001”—返回拍照的图片的base64字符串

}

**任务类型说明**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **任务类型(**tasktype**)** | **任务参数(**taskparam**)** | **返回数据(**taskdata**)** | **说明** |
| 1 | 1001 |  | 现场人像照片，Base64字符串 | 请求设备拍照 |
| 2 | 1002 | 证件人像照片，Base64字符串 | json字符串，livephoto:现场照base64字符串，cmpresult:比对结果，通过/未通过，similarity:相似度 | 请求设备拍照并比对 |
| …… | …… | …… | …… | …… |

**注：**任务类型说明表中的参数为接口向设备下发的任务参数（taskparam），返回数据为json字符串date中的任务数据（taskdata）。

# 第五章 签名说明

## 5.1、MD5签名函数（C#示例）

public static string GetMD5Hash(string input)

{

System.Security.Cryptography.MD5CryptoServiceProvider md5 = new System.Security.Cryptography.MD5CryptoServiceProvider();

byte[] bs = System.Text.Encoding.UTF8.GetBytes(input);

bs = md5.ComputeHash(bs);

System.Text.StringBuilder s = new System.Text.StringBuilder();

foreach (byte b in bs)

{

s.Append(b.ToString("x2").ToLower());

}

return s.ToString();

}

以上为MD5编码示例，主要是采用UTF8转换字节数组，并将生成的MD5码转为字符串。

## 5.2、加密（C#示例）

// 功能：将指定字符串加密(对称算法)

// data-未加密的字符串

// key-加密密钥字符串

// iv-加密初始向量字符串

// 返回：加密后的字符串

public static string Encrypt(string data, string key, string iv)

{

if (string.IsNullOrWhiteSpace(data)) return "";

byte[] dateBytes = System.Text.Encoding.UTF8.GetBytes(data);

byte[] keyBytes = System.Text.Encoding.UTF8.GetBytes(key);

byte[] ivBytes = System.Text.Encoding.UTF8.GetBytes(iv);

byte[] keyOks = new byte[32]; // 密钥32位

Array.Copy(keyBytes, keyOks, System.Math.Min(keyBytes.Length, keyOks.Length));

// 调用字节数组加密函数

byte[] dataOks = Encrypt(dateBytes, keyOks, ivBytes);

// 转为字符串

System.Text.StringBuilder s = new System.Text.StringBuilder();

foreach (byte b in dataOks)

{

s.Append(b.ToString("x2").ToLower());

}

return s.ToString();

}

// 功能：将指定二进制数组加密(对称算法)

public static byte[] Encrypt(byte[] data, byte[] key, byte[] iv)

{

if (data == null)

{

return null;

}

else if (data.Length == 0)

{

return new byte[0];

}

// 设置向量为16字节

byte[] ivOk = new byte[16];

Array.Copy(iv, ivOk, System.Math.Min(iv.Length, ivOk.Length));

// 采用对称加密法

var aes = new RijndaelManaged();

// 秘钥的大小，以位为单位

aes.KeySize = 256;

// 支持的块大小

aes.BlockSize = 128;

// 填充模式

aes.Padding = PaddingMode.PKCS7;

aes.Mode = CipherMode.CBC;

aes.Key = key;

aes.IV = ivOk;

var encrypt = aes.CreateEncryptor(aes.Key, aes.IV);

byte[] xBuff = new byte[0];

using (var ms = new MemoryStream())

{

using(var cs = new CryptoStream(ms, encrypt, CryptoStreamMode.Write))

{

cs.Write(data, 0, data.Length);

cs.FlushFinalBlock();

}

xBuff = ms.ToArray();

}

return xBuff;

}

以上为C#加密示例代码，由于各种语言的差异，仅作参考。客户端如果不进行数据加密，则不设置aes或设置aes=“false”即可。

# 第六章 错误码说明

## 6.1、公共错误码说明

|  |  |
| --- | --- |
| **errCode值** | **说 明** |
| 0 | 接口命令执行成功 |
| -1 | 缺少数据项，缺少cmd、account、nonce、client、timestamp、data、signatrue中的一项或多项值 |
| -2 | 错误的时间戳timestamp，时间戳超期！请检查服务所在服务器或设备时钟是否正确 |
| -3 | 服务账号不存在 |
| -4 | 服务账号无效 |
| -5 | 服务账号未注册企业 |
| -6 | 企业无效 |
| -7 | 未设置签名密钥 |
| -8 | 签名验证失败 |
| -9 | 无法处理加密数据，但未设置加密密钥 |
| -10 | 解密数据失败 |
| -11 | 不支持的接口命令 |
| -12 | 请设置服务账号的可信域名 |
| -13 | 请求地址不是服务账号的可信域名，请核查服务账号的可信域名设置 |
| -19 | 解析数据包异常 |
| -20 | 执行接口命令异常 |
| -21 | 接口数据校验错误  同时返回errItems，存放具体错误的描述 |

## 6.2、身份证人像核验服务错误码说明

|  |  |
| --- | --- |
| **errCode值** | **说 明** |
| 0 | 成功 |
| … | … |

## 6.3、身份证指纹核验服务错误码说明

|  |  |
| --- | --- |
| **errCode值** | **说 明** |
| 0 | 成功 |
| … | … |

## 6.4、未带证核验服务错误码说明

|  |  |
| --- | --- |
| **errCode值** | **说 明** |
| 0 | 认证成功 |
| 1 | 认证不符 |
| … | … |

# 第七章 示例数据格式

## 7.1、身份证人像核验数据示例

示例数据：

{

"account": "dahe",

"client": "4108831987",

"cmd": "FaceVerify",

"nonce": "ec0a4752d93541a8bebde392e376cc98",

"timestamp": "20200911142449",

"signatrue": "a4752d93541a8bebde392e376",

"aes": false,

"data": "{\"name\":\"付小明\",\"enname\":\"\",\"paperType\":\"1\",\"number\":\"410883198610151071\",\"sex\":\"女\",\"folk\":\"汉\",\"country\":\"\",\"address\":\"河南洛阳龙门大道1001号\",\"birthday\":\"19900109\",\"signOrgan\":\"\",\"valid\":\"\",\"idCard\":\"41987654321\",\"passNumber\":\"\",\"changeNum\":\"\",\"headImage\":\"Base64字符串\",\"snapshot\":\"Base64字符串\",\"similarity\":\"0.8\",\"isPass\":\"true\",\"fpFeatrue1\":\"\",\"fpFeatrue2\":\"\",\"sceneFP1\":\"\",\"sceneFP2\":\"\",\"fpImage1\":\"\",\"fpImage2\":\"\",\"deviceSN\":\"41088319871001\"}"

}

## 7.2、身份证指纹核验数据示例

示例数据：

{

"account": "dahe",

"client": "4108831987",

"cmd": "FPVerify",

"nonce": "ec0a4752d93541a8bebde392e376cc98",

"timestamp": "20200911142449",

"signatrue": "a4752d93541a8bebde392e376",

"aes": false,

"data": "{\"name\":\"付小明\",\"enname\":\"fuguiyang\",\"paperType\":\"1\",\"number\":\"410883198910151071\",\"sex\":\"女\",\"folk\":\"汉\",\"country\":\"\",\"address\":\"河南洛阳龙门大道1001号\",\"birthday\":\"19900109\",\"signOrgan\":\"\",\"valid\":\"\",\"idCard\":\"41987654321\",\"passNumber\":\"\",\"changeNum\":\"\",\"headImage\":\"Base64字符串\",\"snapshot\":\"Base64字符串\",\"similarity\":\"0.8\",\"isPass\":\"true\",\"fpFeatrue1\":\"Base64字符串\",\"fpFeatrue2\":\"Base64字符串\",\"sceneFP1\":\"Base64字符串\",\"sceneFP2\":\"\",\"fpImage1\":\"\",\"fpImage2\":\"\",\"deviceSN\":\"41088319871001\"}"

}

## 7.3、未带证核验数据示例

示例数据：

{

"account": "dahe",

"client": "4108831987",

"cmd": "NoPaper",

"nonce": "e70cb1c7eae240ed8be989fc8e435edc",

"timestamp": "20200911141232",

"signatrue": "f02abfa7296c078c36721f8514d36d6b",

"aes": false,

"data": "{\"name\":\"付小明\",\"number\":\"410883199010151071\",\"folk\":\"\",\"address\":\"洛阳市龙门大道100号\",\"snapshot\":\"Base64字符串\",\"sceneFP1\":\"\",\"sceneFP2\":\"\",\"fpImage1\":\"\",\"fpImage2\":\"\",\"deviceSN\":\"41088319871001\"}"

}