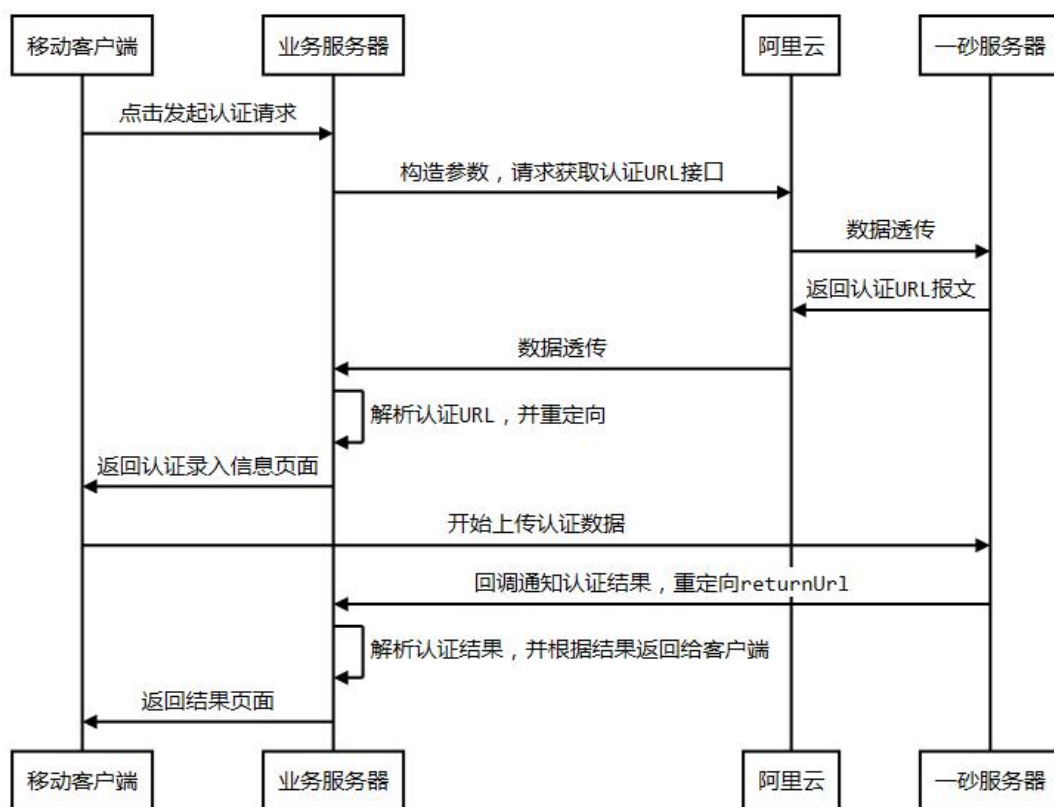


## 版本历史

日期	版本号	作者/修改者	描述	审核人
2020-3	V1.0	子陵		
2020-3	V1.1	子陵	增加可选返回返回身份证照片; 增加可选仅支持拍照模式	

# 实人认证三要素接入文档

## 1. 调用流程



## 2. 接入阿里云获取认证 url

### ➤ 请求 Url

路径	描述
<a href="https://edish5.market.alicloudapi.com/edis_ctid_id_name_video_ocr_h5">https://edish5.market.alicloudapi.com/edis_ctid_id_name_video_ocr_h5</a>	阿里云获取认证 URL

## ➤ 请求报文

字段	是否必传	类型	长度	说明
bizNo	必传	String	16	业务 id, 随机数, 需保证其唯一
returnUrl	必传	String	/	用户完成或取消验证后网页跳转的目标 URL
notifyUrl	必传	String	/	用户完成验证、取消验证、或验证超时后, 由服务器请求客户服务器的 URL
textBgColor	可选	String	/	按钮背景颜色
pageBgColor	可选	String	/	页面背景颜色
ocrOnly	可选	Boolean		是否仅支持拍照录入, 默认为 false
retIdImg	可选	Boolean		是否返回身份证照片, 默认为 false (只有拍照录入才生效)

## ➤ 响应报文

字段	类型	长度	说明
ver	String	/	协议版本
code	String	4	响应状态码
msg	String	/	异常信息描述
verifyUrl	String	/	实人认证 H5 页面 URL
expired	Ineteger	/	token 超时时间

## ➤ 响应码 CODE 说明

值	类型	返回码说明
0000	String	成功
F999	String	系统异常
9999	String	系统繁忙

## ➤ Php Demo Code

```
<?php
$host = "https://edish5.market.alicloudapi.com";
$path = "/edis_ctid_id_name_video_ocr_h5";
$method = "POST";
$appcode = "你的阿里云 code"; //你的阿里云 code
$headers = array();
array_push($headers, "Authorization:APPCODE " . $appcode);
//根据 API 的要求, 定义相对应的 Content-Type
array_push($headers,"Content-Type".":". "application/x-www-form-urlencoded;
charset=UTF-8");
$body = array();
$url = $host . $path;
$body['bizNo'] = "bizNo=1923797893247372942"; //唯一业务 id
$body['notifyUrl'] = "notifyUrl=http://faceid.dev.esandinfo.com/faceId/notifyUrl"; //必
选, post 回调通知验证结果
$body['returnUrl'] = "returnUrl=http://faceid.dev.esandinfo.com/faceId/returnUrl"; //
必选, 重定向回调 url 地址, 结果
//$body['pageBgColor'] = "pageBgColor=#eeeeee"; //可选, 背景颜色 #eeeeee
//$body['textBgColor'] = "textBgColor=#cccccc"; //可选, 按钮, 文字背景色
#cccccc
//$body['ocrOnly'] = "ocrOnly=true"; //可选, 是否值支持拍照录入, 默认为 false
//$body['retIdImg'] = "retIdImg=true"; //可选, 是否返回身份证照片,默认为 false (只
有拍照录入才生效)
$body = $body['bizNo']."&".$body['notifyUrl']."&".$body['returnUrl'];
$curl = curl_init();
curl_setopt($curl, CURLOPT_CUSTOMREQUEST, $method);
curl_setopt($curl, CURLOPT_URL, $url);
curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);
curl_setopt($curl, CURLOPT_FAILONERROR, false);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_HEADER, true);
if (1 == strpos("$".$host, "https://"))
{
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, false);
}
curl_setopt($curl, CURLOPT_POSTFIELDS, $body);
var_dump(curl_exec($curl));
?>
```

## ➤ Java Demo Code

```
public static void main(String[] args) {
    String host = "https://edish5.market.alicloudapi.com";
    String path = "/edis_ctid_id_name_video_ocr_h5";
    String method = "POST";
    String appcode = "你自己阿里云的 AppCode"; //你自己阿里云的 AppCode
    Map<String, String> headers = new HashMap<String, String>();
    //最后在 header 中的格式(中间是英文空格)为 Authorization:APPCODE
83359fd73fe94948385f570e3c139105
    headers.put("Authorization", "APPCODE " + appcode);
    //根据 API 的要求, 定义相对应的 Content-Type
    headers.put("Content-Type", "application/x-www-form-urlencoded;
charset=UTF-8");
    Map<String, String> querys = new HashMap<String, String>();
    Map<String, String> bodys = new HashMap<String, String>();
    bodys.put("bizNo", "19237978932479372942");
    bodys.put("notifyUrl", "http://faceid.dev.esandinfo.com/faceid/notifyUrl");
    bodys.put("ocrOnly", "false");
    bodys.put("pageBgColor", "#eeeeee");
    bodys.put("retIdImg", "false");
    bodys.put("returnUrl", "http://faceid.dev.esandinfo.com/faceid/returnUrl");
    bodys.put("textBgColor", "#cccccc");
    try {
        /**
         * 重要提示如下:
         * HttpUtils 请从
         *
https://github.com/aliyun/api-gateway-demo-sign-java/blob/master/src/main/java/com/aliyun/api/gateway/demo/util/HttpUtils.java
         * 下载
         * 相应的依赖请参照
         *
https://github.com/aliyun/api-gateway-demo-sign-java/blob/master/pom.xml
         */
        HttpResponse response = HttpUtils.doPost(host, path, method, headers,
querys, bodys);
//        System.out.println(response.toString());
//        //获取 response 的 body
        System.out.println(EntityUtils.toString(response.getEntity()));
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```



### 3. 请求认证 url

- 直接请求返回的 verifyUrl，跳转到身份信息录入界面（可选身份证拍照录入）
- 根据流程一步步操作，即可完成验证。

### 4. 认证完成结果说明

notidyUrl 通知回调接口 和 returnUrl 页面跳转回调接口 返回参数都一样，提供两种业务场景，可选择参考。

- 回调报文

字段	类型	说明
data	String	业务数据
sign	String	对 Data 数据的签名 (签名类型, ecc256)

- data 字段数据格式(json 字符串)

字段	类型	说明
code	String	操作结果(值参考操作结果)
msg	String	操作结果描述
bizNo	String	getVerifyUrl 传入的 bizNo
id	String	身份证号
name	String	姓名
gender	String	性别 [男/女], 拍照模式才有值
address	String	住址, 拍照模式才有值
race	String	民族,拍照模式才有值

字段	类型	说明
issuedBy	String	身份证签发机构, 拍照模式才有值
validDate	String	有效期(如 2010.11.13-2020.11.13), 拍照模式才有值
idFront	String	Base64 编码的身份证人脸面图片, 拍照模式且选择需要才有值
idBack	String	Base64 编码的身份证国徽面图片, 拍照模式且选择需要才有值

- 操作结果

值	类型	说明
PASS	String	实人认证认证成功
FAIL	String	实人认证失败
LIVE_DETECTION_FAILED	String	活体检测失败

- 验签说明

签名方式采用 ECC256, 以下是 PHP 和 JAVA 验签 Demo

```
esand_public_key:
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEWKKJoLwh6XEBkTeCfVbKSB3zkkycbld
d8SBabj2jpWynXx0pBZvdFpbb9AEiyrnM8blmhpz8YOXc2yUuN1ui/w==
```

PHP 验签方法

```
public static function verify($text,$sign){
    $sign = base64_decode($sign);
    $public_key = openssl_pkey_get_public('-----BEGIN PUBLIC KEY-----
    esand_public_key
    -----END PUBLIC KEY-----');
    if(!$public_key){
        die('error');
    }
    $flag = openssl_verify($text,$sign,$public_key,OPENSSL_ALGO_SHA256);
    if($flag == 1)
        return true;
    else {
        if($flag == 0){
            echo "验签不通过";
        }elseif ($flag == -1){
            echo "error";
        }
        return false;
    }
}
```



## JAVA 验签

```
public static boolean verify(byte[] hashData, String signatureBase64) {
    boolean verifyResult = false;
    byte[] publicKeyBuff = Base64.decodeBase64("esand_public_key");
    byte[] signatureBuff = Base64.decodeBase64(signatureBase64);
    Security.addProvider(new BouncyCastleProvider());
    X509EncodedKeySpec      x509EncodedKeySpec      =      new
X509EncodedKeySpec(publicKeyBuff);
    try {
        KeyFactory keyFactory = KeyFactory.getInstance(KEY_ALGORITHM);
        PublicKey publicKey = keyFactory.generatePublic(x509EncodedKeySpec);
        Signature signature = Signature.getInstance(SIGNATURE_ALGORITHM);
        signature.initVerify(publicKey);
        signature.update(hashData);
        verifyResult = signature.verify(signatureBuff);
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    } finally {
        return verifyResult;
    }
}
```

## 5. 收费说明

本服务收费根据最终的认证结果进行收费。

当拉起认证流程，认证中途取消、关闭退出、认证失败并不进行收费处理。只有当认证成功时才进行收费。