1 银行卡四要素验证接口[712]

1.1 接口介绍

姓名、身份证号、银行卡卡号、与银行卡绑定的手机号四要素认证，传入姓名、身份证证件号码、银行卡卡号、与银行卡绑定的手机号，返回匹配结果。

1.2 接口信息

1.2.1接口地址

名称	内容
生产地址	https://api.situdata.com/situ-identity/v2/id-verify/bank-four-elements
测试地址	https://staging-api.situdata.com/situ-identity/v2/id-verify/bank-four-elements

1.2.2接口格式

名称	内容
请求方式	POST
返回类型	Json
编码格式	UTF-8

1.2.3接口 Header

名称	类型	必填	内容
x-access-id	String	是	客户账号 ID（如无请联系商务）
x-signature	String	是	客户校签码，计算方法为

			HmacSHA256(x-access-key + x-request-send-timestamp)，服务端根据 signature 来验证请求 的合法性，校签方法示例代码请见示例代码 generateSignature 方法。
x-request-send-timestamp	Int	是	客户发送请求时间。linux epoch，单位(秒) (注意：如果与服务端收到请求的时间相差 5 分钟则被判为无效。
Content-Type	String	是	application/json

1.2.4接口 Body 请求参数

名称	类型	必须	描述
name	String	是	姓名
idCard	String	是	证件号码
bankCard	String	是	银行卡卡号
mobile	String	是	与银行卡绑定的手机号

1.2.5接口返回参数

参数名		类型	描述
globalRequestId		String	请求 ID
rtn		int	返回状态码：0 为正常（详见附录）
msg		String	附加结果说明
result	**code**	int	code=1:验证通过 code=2：验证信息不一致

			code=3：无法验证 code=4：密码错误次数超限 code=5：入参格式有误 code=6：银行卡异常
	desc	String	结果描述

1.2.6接口请求和返回示例：

内容名称	内容

请求	{
	"name":"王菲",
	"bankCard":"6214856550650133"，
	"idCard":"421127199004280042"，
	"mobile":"13828731660"
	}

返回 1	{
	"msg": "成功！ ",
	"rtn": 0,
	"globalRequestId": "600x15536005160274506Rr",
	"result": {
	"code": 1,
	"desc": "验证通过"
	}
	}

返回 2	{
	"msg": "传入参数错误！ ",
	"rtn": 1,
	"globalRequestId": "600x15536005160274506RB",
	"result": "null"
	}

1.2.7接口请求示例：Java

```
import com.alibaba.fastjson.JSONObject;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.entity.StringEntity;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.io.IOException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.util.Base64;
import java.util.HashMap;

public class bankCardThreeElementsTest {
    private static String url = "接口 url";
    private static String accessId = "您的 accessId";
    private static String accessKey = "您的 accessKey";
    public static void main(String[] args) {
        String name = "您的姓名";
        String idCard= "您的证件号";
        String bankCard="您的银行卡卡号";
        String mobile="您的手机号";
        HashMap paramMap = new HashMap<String, Object>() {
            {
                put("name", name);
                put("idCard", idCard);
                put("bankCard", bankCard);
                put("mobile", mobile);
            }
        };
        HashMap<String, String> header = new HashMap<>();
        long timeMillis = System.currentTimeMillis() / 1000L - 10;
        String signature = generateSignature(accessId, accessKey, timeMillis);
        // 设置通用的请求属性,设置 accessId 和 signature
        header.put("accept", "*/*");
        header.put("connection", "Keep-Alive");
        header.put("Content-Type", "application/json");
        header.put("x-access-id", accessId);
        header.put("x-request-send-timestamp", timeMillis    + "");
        header.put("x-signature", signature);
        String paramJson = JSONObject.toJSONString(paramMap);
        String res = null;
        try {
            res = doPost(url, paramJson, header);
        } catch (IOException e) {
```

```
            e.printStackTrace();
        }
        System.out.println(res);
    }

    private static String generateSignature(String accessId, String base64Key, long timestamp) {
        String accessKey = new String(Base64.getDecoder().decode(base64Key));
        String signKey = accessId + "-" + timestamp;
        try {
            Mac hmacSha256 = Mac.getInstance("HmacSHA256");
            SecretKeySpec secretKey = new SecretKeySpec(accessKey.getBytes(), "HmacSHA256");
            hmacSha256.init(secretKey);
            return Base64.getEncoder().encodeToString(hmacSha256.doFinal(signKey.getBytes()));
        } catch (NoSuchAlgorithmException | InvalidKeyException e) {
            e.printStackTrace();
            return null;
        }
    }

    /**
     * POST 请求，JSON 请求方式
     */
    private static String doPost(String url, String json, HashMap<String, String> header) throws IOException {
        // 创建 Httpclient 对象
        CloseableHttpClient httpClient = HttpClients.createDefault();
        CloseableHttpResponse response = null;
        String resultString = "";
        try {
            // 创建 Http Post 请求
            HttpPost httpPost = new HttpPost(url);
            // 添加请求头
            if (null != header) {
                for (String key : header.keySet()) {
                    httpPost.addHeader(key, header.get(key));
                }
            }
            // json 方式发送 post 请求
            StringEntity entity = new StringEntity(json, "utf-8");// 解决中文乱码问题
            entity.setContentEncoding("UTF-8");
            entity.setContentType("application/json");
            httpPost.setEntity(entity);
            // 执行 http 请求
            response = httpClient.execute(httpPost);
            // 判断返回状态是否为 200
            if (200 == response.getStatusLine().getStatusCode()) {
                resultString = EntityUtils.toString(response.getEntity(), "UTF-8");
            }
        } finally {
            try {
```

```
                    if (null != response) {
                        response.close();
                    }
                    httpClient.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
            return resultString;
        }

}
```