北京慧阅信息服务有限公司

银行卡验证接口

V1.0.1

目录

1、接入方需要申请的信息

验证帐号(sid)，3DES 密码，Md5 密码，IVPS:12345678（固定值）

2、接口地址

http://verifyapi.huiyuenet.com/zxbank/verifyApi.do

3、请求规则

说明	描述
传输方式	采用 HTTP
请求方式	采用 POST 方法请求
数据格式	Json 字符串
字符编码	UTF-8
算法	MD5
校验	请求和接受数据均需要校验

4、请求参数

字段	描述	Value	备注
name	姓名(UTF-8 编码)	用 3des 加密后 base64 编码成 string 字符串	
idnum	身份证号	用 3des 加密后 base64 编码成 string 字符串	只支持 18 位
phone	电话号码	用 3des 加密后 base64 编码成 string 字符串	
sid	服务账号		
vtype	02/03	02 代表三项验证，03 代表 4 项验证	
bankCard	银行卡号	用 3des 加密后 base64 编码成 string 字符串	
cpserialnum	请求方订单号	长度不超过 20 个字符	
md5num	Md5 编码	bankCard (密)+idnum(密)+cpserialnum+md5 密码字符串连接后进行 md5	

【注：vtype=02 时，phone 值必须为””3des 加密后的值，当 vtype=03，phone 为真实手机号 3des 加密后值】

5、返回数据 JSON 参数说明

字段	描述	Value	描述
cpserialnum	公司流水号		

result	验证结果	**BANKCONSISTENT**：一致（收费） **BANKINCONSISTENT**：不一致（收费） **BANKNOLIB**：库无（不支持该银行卡验证）（不收费） **FAIL**：验证失败（见 errmsg 描述）	**FAIL 收费见附录**
sysserialnum	系统验证流水号	长度不超过 32 个字符	**FAIL 无**
md5num	Md5 编码	result+cpserialnum+sysserialnum+ md5 密码字符串连接后进行 md5	**FAIL 无**
errmsg	错误代码	见附录	**FAIL 有**

【注：当 result 为 FAIL 时{"errmsg":"ERR2014","result":"FAIL"}，否则每个字段都有】

6、附录

错误码	描述	是否收费
ERR2022	姓名/身份证号/手机号不匹配	是
ERR2012	数据MD5不正确	否
ERR2013	无效身份证号码	是
ERR2014	姓名为空或长度不正确	否
ERR2015	无效手机号码	是
ERR2016	无效银行账号	是
ERR2017	参数不能为空	否
ERR2018	其他错误（银行返回）	是
ERR2019	未找到用户	否
ERR2020	账号已停用	否
ERR2021	银行卡内部接口错误	否
ERR9999	其他错误	否

7、POST 示例

```
public static String sendPost(String url, String paramJson,
        int connectionTimeout, int soTimeOut) throws IOException {
    PrintWriter out = null;
    BufferedReader in = null;
    try {
        URL realUrl = new URL(url);
        // 打开和URL之间的连接
        HttpURLConnection conn = (HttpURLConnection)
realUrl.openConnection();
```

```
            conn.setConnectTimeout(connectionTimeout);
            // 读取数据超时，毫秒
            conn.setReadTimeout(soTimeOut);
            // 设置通用的请求属性
            conn.setRequestProperty("accept", "*/*");
            conn.setRequestProperty("connection", "Keep-Alive");
            conn.setRequestProperty("user-agent", "Mozilla/4.0 (compatible;
MSIE 6.0; Windows NT 5.1;SV1)");
            // 发送POST请求必须设置如下两行
            conn.setDoOutput(true); conn.setDoInput(true);
            conn.setRequestMethod("POST");
            conn.setUseCaches(false);
            conn.setInstanceFollowRedirects(true);
            conn.setRequestProperty("Content-Type", application/json");
            // 获取URLConnection对象对应的输出流
            out = new PrintWriter(conn.getOutputStream());
            // 发送请求参数
            out.print(paramJson);
            // flush输出流的缓冲
            out.flush();
            // 定义BufferedReader输入流来读取URL的响应
            in = new BufferedReader(
                    new InputStreamReader(conn.getInputStream()));
            StringBuffer stringBuffer = new StringBuffer();
            String str = StringUtils.EMPTY;
            while ((str = in.readLine()) != null) {
                stringBuffer.append(str);
            }
            return stringBuffer.toString();
        } catch (IOException e) {
            throw e;
        } finally {// 使用finally块来关闭输出流、输入流
            try {
                if (out != null) {
                    out.close();
                }
                if (in != null) {
                    in.close();
                }
            } catch (IOException ex) {
                ex.printStackTrace();
            }
        }
    }
```