

# 全网手机三要素验证（尊享版）

贵州数据宝网络科技有限公司专有

2018年07月16日

数据宝技术部、产品加工部

## 更新日志

| 时间 | 说明 |
|----|----|
|----|----|

贵州数据宝网络科技有限公司专有

## 1. 接口信息

### 1.1 全网手机三要素验证（尊享版）

URL: /communication/personal/1979

REQUEST TYPE : post

REQUEST PARAM :

| Param name | Param type | Required | Desc         |
|------------|------------|----------|--------------|
| key        | string     | 是        | 您购买的API的key值 |
| name       | string     | 是        | 姓名           |
| idcard     | string     | 是        | 身份证号         |
| mobile     | string     | 是        | 手机号          |

RESPONSE PARAM :

| Param name | Param type | Desc                 |
|------------|------------|----------------------|
| code       | string     | 请求code码              |
| message    | string     | code码说明              |
| data       | string     | 接口返回数据体              |
| seqNo      | string     | 调用唯一标识（如有接口问题，请提供此值） |

RESPONSE PARAM data :

| Param name | Param type | Desc                    |
|------------|------------|-------------------------|
| state      | string     | （1:验证一致，2:验证不一致，3:异常情况） |

SUCCESS RESPONSE :

```
{
  "code": "10000",
  "message": "成功",
  "data": {
    "state": "1"
  },
  "seqNo": "4XU29Z4D1704061618"
}
```

ERROR CODE :

| Code value | Desc      |
|------------|-----------|
| 10000      | 成功        |
| 10001      | 请求报文错误    |
| 10002      | 系统错误      |
| 10003      | 请求报文错误    |
| 10004      | 无权限       |
| 10005      | 用户验证失败    |
| 10006      | 系统繁忙      |
| 10007      | 调用接口失败    |
| 10008      | 加密失败      |
| 10009      | 请求报文格式错误  |
| 10010      | 发送的报文不能为空 |
| 10011      | 请求地址错误    |
| 10012      | 系统错误      |

|       |                         |
|-------|-------------------------|
| 10013 | 数据加密失败                  |
| 10014 | 接口调用失败                  |
| 10015 | 未查询到数据                  |
| 10016 | 非法参数                    |
| 10017 | 系统超时                    |
| 10018 | 查询无记录                   |
| 10019 | 无记录                     |
| 10020 | 调用服务接口错误                |
| 10022 | 获取通道信息出错                |
| 10024 | 机构交易无效                  |
| 10025 | 交易流水号重复                 |
| 10027 | 交易类型无效                  |
| 10028 | 不正确的访问                  |
| 10029 | 请求参数不合法                 |
| 10030 | 请求处理超时                  |
| 10031 | 服务地址获取失败                |
| 10032 | 构造响应消息失败                |
| 10033 | 绑定不存在                   |
| 10034 | 非本网号码                   |
| 10035 | 本网号段，但非用户号码（比如是设备码，漫游号） |
| 10036 | 未能获取到地址                 |
| 10037 | 无可用的通道                  |
| 10038 | 请求失败                    |
| 10039 | 未知错误                    |
| 10040 | 系统超时                    |
| 10041 | 系统异常                    |
| 10042 | 接口调用异常                  |
| 10043 | 接口请求失败                  |
| 10043 | 接口请求失败                  |
| 10044 | 查询无结果                   |

## 2. 全系统错误码

| Code value | Desc   |
|------------|--|
| SYSTEM_900 | IP 不合法   |
| SYSTEM_999 | 接口处理异常   |
| SYSTEM_000 | key 参数不能为空   |
| SYSTEM_001 | 找不到这个 key  |
| SYSTEM_002 | 调用次数已用完  |
| SYSTEM_003 | 用户该接口状态为不可用  |
| SYSTEM_004 | 接口信息不存在  |
| SYSTEM_005 | 你没有认证信息  |
| SYSTEM_008 | 当前接口只允许“企业认证”通过的账户进行调用，请在数据宝官网个人中心进行企业认证后再进行调用，谢谢！ |
| SYSTEM_009 | 必须认证审核通过才可以使用                                      |
| SYSTEM_011 | 接口缺少参数   |
| SYSTEM_012 | 没有 ip 访问权限   |
| SYSTEM_013 | 接口模板不存在  |
| SYSTEM_014 | 接口模板没开启  |
| SYSTEM_015 | 该接口已下架   |
| SYSTEM_017 | 模板配置的平台参数与请求的参数不一致                                 |
| SYSTEM_019 | 调用第三方协议配置错误  |
| SYSTEM_020 | 调用第三方产生异常  |
| SYSTEM_022 | 调用第三方返回的数据格式错误                                     |
| SYSTEM_025 | 你没有购买此接口   |
| SYSTEM_026 | 用户信息不存在  |
| SYSTEM_027 | 请求第三方地址超时，请稍后再试                                    |
| SYSTEM_028 | 请求第三方地址被拒绝，请稍后再试                                   |
| SYSTEM_029 | 返回示例错误   |
| SYSTEM_034 | 签名不合法  |
| SYSTEM_035 | 请求参数加密有误   |
| SYSTEM_036 | 验签失败   |
| SYSTEM_037 | timestamp 不能为空                                     |
| SYSTEM_038 | 请求繁忙，请稍候再试   |
| SYSTEM_039 | 请在个人中心接口设置加密状态                                     |
| SYSTEM_040 | timestamp 不合法                                      |
| SYSTEM_041 | timestamp 过期                                       |
| SYSTEM_042 | 身份证手机号等不符合规则                                       |
| SYSTEM_043 | 当前您的接口覆盖范围不能满足本次数据验证，请升级 API 接口套餐获取更丰富的服务          |

### 3. 接口对接示例代码

#### 3.1 sample code

```
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.config.RequestConfig;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class HttpUtil {

    public static void main(String[] args) {
        //接口地址
        String url = "http://api.chinadatipay.com/trade/user/1985";
        //请求参数
        Map<String, Object> params = new HashMap<>();
        //输入数据宝提供的 key
        params.put("key", "");
        //输入局被查询手机号码
        params.put("mobile", "");

        String result = null;
        try {
            result = post(url, params);
        } catch (Exception e) {
            e.printStackTrace();
        }
        System.out.println("result:\n" + result);
    }

    public static String post(String url, Map<String, Object> params) throws Exception {
        ArrayList<NameValuePair> pairs = covertParams2NVPS(params);
        return PostHttpRequest(url, pairs);
    }

    public static String PostHttpRequest(String Url, List<NameValuePair> params) throws Exception {
        CloseableHttpClient client = HttpClients.createDefault();
        //超时时间
        RequestConfig requestConfig = RequestConfig.custom()
            .setSocketTimeout(300000)
            .setConnectTimeout(300000)
```

```

        .build();
String result = null;
try {
    HttpPost request = new HttpPost(Uri);
    request.setConfig(requestConfig);
    request.setEntity(new UrlEncodedFormEntity(params, "UTF-8"));
    HttpResponse responses = client.execute(request);
    if (responses.getStatusLine().getStatusCode() == 200) {
        result = EntityUtils.toString(responses.getEntity(), "UTF-8");
    }
} catch (Exception e) {
    e.printStackTrace();
} finally {
    client.close();
}
return result;
}

private static ArrayList<NameValuePair> covertParams2NVPS(Map<String, Object> params) {
    ArrayList<NameValuePair> pairs = new ArrayList<>();
    if (params == null || params.size() == 0) {
        return pairs;
    }
    for (Map.Entry<String, Object> param : params.entrySet()) {
        Object value = param.getValue();
        if (value instanceof String[]) {
            String[] values = (String[]) value;
            for (String v : values) {
                pairs.add(new BasicNameValuePair(param.getKey(), v));
            }
        } else {
            if (value instanceof Integer) {
                value = Integer.toString((Integer) value);
            } else if (value instanceof Long) {
                value = Long.toString((Long) value);
            }
            pairs.add(new BasicNameValuePair(param.getKey(), (String) value));
        }
    }
    return pairs;
}
}
}

```

## 4. 加密对接说明

我们的加密方式使用 MD5 进行 sign 签名验证，以此来杜绝信息篡改的发生，同时针对入参和出参内容进行对称加密（base64 编码）。以下具体讲解对接方式。

对接步骤如下：

## 4.1 设置加密

第一步：登陆 <https://www.chinadatapay.com>, 首页 > 个人中心 > 接口管理 > 我的数据宝 点击”获取密钥”，如下图，举例 secretKey:lgJKiiakuplMXy4s



第二步：首页 > 个人中心 > 接口管理 > 我的数据 选择某个接口操作栏中的”数据服务”按钮，弹框中选择”加密对接”



## 4.2 技术对接

接口的入参和出参参数名无需使用 base64 编码，只是针对入参值和出参值使用密码加解密。案例如下：

Url : <http://api.chinadatapay.com/communication/personal/1896>

RequestWay : Post

SecretKey : lgJKiiakuplMXy4s

Request Param:

| 名称     | 类型     | 必填 | 说明               |
|--------|--------|----|------------------|
| key    | string | 是  | 您购买的 API 的 key 值 |
| name   | string | 是  | 姓名               |
| idcard | string | 是  | 身份证号             |
| mobile | string | 是  | 手机号              |

e. g

key=您购买的 API 的 key 值

name=谢天浩&idcard=342623198610025939&mobile=15000639994

timestamp=1505352152882

开始处理如下：

第一步、BASE64AES 加密入参：

备注：key 字段不参与加密



```
name=M1H0rGak3gAnL7aBv/qUyw==&idcard=rBsi6bNM04IKFrPylaoPU4y9HUFPUv7L+3SLgsMFcMo=&mobile=yn30yI+FuN96GD6WT10ATg==
```

```
timestamp=g1/T7VMDawD7mxhE3uAk6A==
```

## 第二步、sign 令牌获取：

规则为入参按照 ACS 码排序, 结果如下:

```
idcard=rBsi6bNM04IKFrPylaoPU4y9HUFPUv7L+3SLgsMFcMo=&mobile=yn30yI+FuN96GD6WT10ATg==&name=M1H0rGak3gAnL7aBv/qUyw==&timestamp=g1/T7VMDawD7mxhE3uAk6A==
```

备注: key 字段不参与签名

```
sign=new BASE64Encoder().encode(md5(入参按照 ACS 码排序结果))
```

```
获得入参 sign=ISzmFWZQeJUFqnKe9cdrFA==
```

## 第三步、开始发送请求：

```
url: http://api.chinadatipay.com/communication/personal/1896
```

```
requestWay:post
```

Request Param:

key=您购买的 API 的 key 值

```
name=M1H0rGak3gAnL7aBv/qUyw==&idcard=rBsi6bNM04IKFrPylaoPU4y9HUFPUv7L+3SLgsMFcMo=&mobile=yn30yI+FuN96GD6WT10ATg==&timestamp=g1/T7VMDawD7mxhE3uAk6A==
```

```
sign=ISzmFWZQeJUFqnKe9cdrFA==
```

Response result:

```
{
  "code": "10000",
  "message": "成功",
  "data": "81hwJ3Fzj4De9fNJccustQ==",
  "seqNo": "C03SX59Z1709071444"
}
```

data 值秘钥进行 BASE64AES 解密, 解密结果如下: {"state": "1"}

## 4.3 java 代码示例

### Maven pom Repositories:

```
<dependency>
  <groupId>com.cdp.product</groupId>
  <artifactId>cdp-common-security</artifactId>
  <version>3.5.0</version>
</dependency>
```

### BASE64AES 加密工具类和方法：

```
com.cdp.product.security.encode.CdpEncryptUtil.aesEncrypt(明文,秘钥)
```

### BASE64AES 解密工具类和方法：

```
com.cdp.product.security.decode.CdpDecryptUtil.aesDecrypt(密文,秘钥)
```

```
Sign 签名工具类和方法：CdpSignUtil.sign(Map<String, String> param)
```

加密解密代码示例如下：

```
package com.cdp.product.security.test;

import com.cdp.product.security.decode.CdpDecryptUtil;
import com.cdp.product.security.encode.CdpEncryptUtil;
import com.cdp.product.security.exception.DecryptFailureException;
import com.cdp.product.security.exception.EncryptFailureException;
import com.cdp.product.security.exception.SignFailureException;
import com.cdp.product.security.sign.CdpSignUtil;

import java.util.HashMap;
import java.util.Map;

/**
 * 测试加解密以及签名
 */
public class CommonTest {
    public static void main(String[] args) throws EncryptFailureException, SignFailureException, DecryptFailureException {
        //密钥
        String secretKey = "lgJKiakuplMXy4s";
        //入参集合 针对入参 value 值进行加密
        Map<String, String> param = new HashMap<>();
        param.put("name", CdpEncryptUtil.aesEncrypt("谢天浩", secretKey));
        param.put("idcard", CdpEncryptUtil.aesEncrypt("342623198610025939", secretKey));
        param.put("mobile", CdpEncryptUtil.aesEncrypt("15000639994", secretKey));
        param.put("timestamp", CdpEncryptUtil.aesEncrypt(System.currentTimeMillis() + "", secretKey));
        //获取 sign 签名
        String sign = CdpSignUtil.sign(param);
        //返回各种加解密签名结果
        System.out.println(String.format("入参集合:\n%s", param));
        System.out.println(String.format("获取 sign 值:\n%s", sign));
        System.out.print(String.format("返回结果解密:\n%s",
            CdpDecryptUtil.aesDecrypt("81hwJ3Fzj4De9fNjccustQ==", secretKey)));
    }
}
```

结果示例：

```
入参集合:
{timestamp=gl/T7VMDawD7mxhE3uAk6A==, name=M1H0rGak3gAnL7aBv/qUyw==,
idcard=rBsi6bNM04IKFrPy1aoPU4y9HUFPUv7L+3SLgsMFcMo=, mobile=yn30yI+FuN96GD6WT10ATg==}
获取 sign 值:
ISzmFWZQeJUFqnKe9cdrFA==
返回结果解密:
{"state": "1"}
```