

信息核验-身份核验接口文档

日期	版本	编写人	说明
2019/11/19	V1.0	大白梨	创建文档

一、接口介绍

信息核验-身份核验接口，该 API 的功能是验证身份信息（身份证号和姓名）是否匹配；

二、通用说明

1、请求地址：

<https://api.apetime.cn/api/verify/identity>

2、请求方式：

POST

3、时间戳：

timestamp 取 GMT1970 年 1 月 1 日 00:00:00 到现在所经历的毫秒数

4、签名规则：

1、非必填字段不参与签名

2、将签名字段的字段值按字母序排列，然后将他们对应的值拼接在一起

3、将 app_secret 拼接到上一步产生的字符串后面（app_secret 由平台提供）

4、利用 md5 得到签名字段

5、具体拼接规则如下：

app_key+id_number+name+request_id+timestamp+app_secret

将各参数对应的值拼接在一起

举例：

app_key = aserd2345kiutt

request_id = sdkluf132345ddd

name = 张三

id_number = 321234232123421231

timestamp = 1575975345623

该客户的 app_secret 为 2231234124ssss

则进行 md5 签名的字符串为

aserd2345kiutt321234232123421231 张三

sdkluf132345ddd15759753456232231234124ssss

加签方式

Md5Util.createMD5ByStr(“aserd2345kiutt321234232123421231 张三

sdkluf132345ddd15759753456232231234124ssss”)

三、接口及参数说明

1、请求参数：

字段	类型	必需	描述
app_key	string	是	客户应用标识，由平台提供

字段	类型	必需	描述
request_id	<i>string</i>	是	客户请求流水号，建议唯一性
name	<i>string</i>	是	姓名
id_number	<i>string</i>	是	身份证号
timestamp	<i>long</i>	是	时间戳
sign	<i>string</i>	是	签名

所有中文和特殊字符必需以 UTF-8 编码转义。

2、返回参数

字段	类型	说明
trace_id	<i>string</i>	本次请求的唯一流水号
code	<i>string</i>	业务响应码
msg	<i>string</i>	说明
data	object	消息体（只有 0000 调用成功时返回）
charge	<i>int</i>	是否收费，1 收费，2 不收费

data 详情：

字段	类型	说明
----	----	----

字段	类型	说明
checkresult	int	1 身份证号和姓名一致 2 身份证号和姓名不一致

返回样例

```
{
  "trace_id": "TID2af751b4bff24be781d60af10bf84101",
  "code": "0000",
  "msg": "调用成功",
  "data": {
    "checkresult":1
  }
  "charge": 1
}
{
  "trace_id": "TID2af751b4bff24be781d60af10bf84101",
  "code": "1000",
  "msg": "验签失败",
  "charge": 2
}
```

3、响应码

code	msg

0000	调用成功
1000	验签失败
1001	参数非 UTF-8 编码
1002	请求参数错误
1003	商户冻结
1004	商户停用
1005	商户账号过期
1006	调用频率过高
1007	免费次数用尽
1008	服务无权限
1009	App_key 验证失败
1010	请求过期
2000	供应商服务异常
9999	其他错误

4、收费标准：

以返回字段中的 charge 为准，当 charge 为 1 时收费，为 2 不收费。

四、代码示例

```
import org.apache.commons.codec.binary.Hex;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.UUID;

public class IdentityVerifyDemo {

    private static String CHARSET = "UTF-8";
    private static String URL = "https://api.apetime.cn/api/verify/identity";
    //连接超时 1000ms
    private static int CONNECT_TIMEOUT = 1000;
    //读超时 2000ms
    private static int READ_TIMEOUT = 2000;

    private static String APP_KEY = "";

    private static String APP_SECRET = "";

    private static String name = "";

    private static String id_number = "";

    public static void main(String[] args) throws IOException,
    NoSuchAlgorithmException {
        Long timestamp = System.currentTimeMillis();
        String requestId = UUID.randomUUID().toString().replaceAll("-", "");

        StringBuilder requestBuilder = new StringBuilder();
```

```

requestBuilder.append("app_key=").append(APP_KEY);
requestBuilder.append("&request_id=").append(requestId);
requestBuilder.append("&name=").append(name);
requestBuilder.append("&id_number=").append(id_number);
requestBuilder.append("&timestamp=").append(timestamp);

StringBuilder builder = new StringBuilder();
builder.append(APP_KEY);
builder.append(id_number);
builder.append(name);
builder.append(requestId);
builder.append(timestamp);
builder.append(APP_SECRET);
MessageDigest digest = MessageDigest.getInstance("MD5");
String sign =
Hex.encodeHexString(digest.digest(builder.toString().getBytes(CHARSET)));

requestBuilder.append("&sign=").append(sign);

BufferedReader br = null;
URLConnection conn = null;
String responseStr = null;

String requestStr = requestBuilder.toString();
try {
    URL obj = new URL(URL);
    conn = (URLConnection) obj.openConnection();
    conn.setRequestMethod("POST");
    conn.setConnectTimeout(CONNECT_TIMEOUT);
    conn.setReadTimeout(READ_TIMEOUT);
    conn.setDoOutput(true);
    conn.setDoInput(true);
    conn.getOutputStream().write(requestStr.getBytes(CHARSET));
    conn.connect();
    br = new BufferedReader(new
InputStreamReader(conn.getInputStream(), CHARSET));
    StringBuffer response = new StringBuffer();

    String line;
    while ((line = br.readLine()) != null) {
        response.append(line);
    }
    responseStr = response.toString();
}

```



```
} finally {  
    if (br != null) {  
        br.close();  
    }  
    if (conn != null) {  
        conn.disconnect();  
    }  
}  
System.out.println(responseStr);  
}  
  
}
```