思图场景 OpenApi

身份证真伪接口文档

版本修订历史纪录				
序号	修订内容	修订日期	修订人	版本
1	基本接口	2017-06-05	李利雪	V1.1.1.2

普通身份证真伪接口

接口介绍

普通身份证真伪接口，回传一张身份证的照片，返回真伪结果。

接口信息

调用地址：http://openapi.situdata.com/id/distinguish
测试调用地址：http://test-openapi.situdata.com/id/distinguish

请求方式：POST

返回类型：JSON

编码格式：utf-8

请求参数：Header

名称	类型	是否必须	描述
X-Auth-Token	String	是	每个账号都会唯一指定一个 token，请妥善保管，token 是接口认证的唯一方法

请求参数：Body

名称	类型	必须	描述
idcard	String	是	身份证图片转 base64 编码以后的字符串（json）

返回参数：

参数名			类型	描述
analyzeId			int	证件分析 ID
code			int	结果： 成功 = 1， 查询失败= 105
message			String	结果说明
result	distinguish	Code	String	成功 = 1， 校验信息错误 = 8，

				疑似假证 = 9， 无法识别 = 10， 参数错误 = 100， 文件编码后大小超限 = 103， 查询失败= 105
		Message	String	结果说明

请求示例：Java

```
import sun.misc.BASE64Encoder;
import java.io.*;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class DrivingLicence1 {

    private static  final String  strUrl = "http://test-openapi.situdata.com/id/distinguish";

    public static void main(String[] args) {

        String filePath = "身份证图片文件目录";
        File file = new File(filePath);

        String[] fileName=file.list();
        for(String vfile:fileName) {
            String imageToBase64 = imageToBase64(filePath+vfile);
            System.out.println(filePath+vfile);
            String replaceBlank = replaceBlank(imageToBase64);
            String params="{\"idcard\":\""+replaceBlank+"\"}";

            try {
                String readByGet = readByGet(strUrl, params);
                System.out.println(readByGet);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }

    }
```

```
    private static String readByGet(String inUrl,String params) throws IOException {
        StringBuffer sbf = new StringBuffer();
        String strRead = null;
        URL url = new URL(inUrl);

        HttpURLConnection connection = (HttpURLConnection) url.openConnection();

        connection.setRequestMethod("POST");// ðúģ

        connection.setDoInput(true);
        connection.setDoOutput(true);
        connection.setRequestProperty("Content-Type", "application/json");
        connection.setRequestProperty("X-Auth-Token", "");

        connection.connect();

        OutputStreamWriter writer = new
OutputStreamWriter(connection.getOutputStream(),"UTF-8");

        writer.write(params);

        writer.flush();

        InputStream is = connection.getInputStream();

        BufferedReader reader = new BufferedReader(new InputStreamReader(is,
                "UTF-8"));
        while ((strRead = reader.readLine()) != null) {
            sbf.append(strRead);
            sbf.append("\r\n");
        }
        reader.close();

        connection.disconnect();
        return sbf.toString();
    }

    public static String imageToBase64(String path) {
        byte[] data = null;

        try {
            InputStream in = new FileInputStream(path);
            data = new byte[in.available()];
```

```
            in.read(data);
            in.close();
        } catch (IOException e) {
            e.printStackTrace();
        }

        BASE64Encoder encoder = new BASE64Encoder();
        return encoder.encode(data);
    }

    public static String replaceBlank(String str) {
        String dest = "";
        if (str!=null) {
            Pattern p = Pattern.compile("\\s*|\t|\r|\n|");
            Matcher m = p.matcher(str);
            dest = m.replaceAll("");
        }
        return dest;
    }
}
```

正常返回示例：

```
样例 1
{
    "result": {
        "distinguish": {
            "code": "1",
            "message": "Ok"
        }
    },
    "code": "1",
    "message": "Ok",
    "analyzeId": 2574
}

样例 2
{
    "analyzeId": 2590
    "code": "105",
    "message": "查询失败",
    "result":{
    }
}
```

样例 3

```
{
    "result": {
        "distinguish": {
            "code": "9",
            "message": "疑似假证"
        }
    },
    "code": "1",
    "message": "Ok",
    "analyzeId": 2638
}
```

样例 4

```
{
    "result": {
        "distinguish": {
            "code": "10",
            "message": "无法识别"
        }
    },
    "code": "1",
    "message": "Ok",
    "analyzeId": 2639
}
```