具体文档：[http://ai.baidu.com/docs#/OCR-API/e1bd77f3](http://ai.baidu.com/docs" \l "/OCR-API/e1bd77f3" \t "_blank)

PS：   
基于Java 1.8   
版本控制：maven   
使用之前需要获取对应的项目API\_KEY，SECRET\_KEY，这些参数在使用API的时候必须用到，用于生成access\_token。   
如何获取这些参数：在百度开发者中心申请一个“通用文字识别”项目，然后就可以获取到这些参数。   
准备条件都完成了，现在开始进行图像识别了。

**test4j图片文字识别教程：[http://blog.csdn.net/wsk1103/article/details/54173282](http://blog.csdn.net/wsk1103/article/details/54173282" \t "_blank)**

**1. 准备pom文件**

<!-- https://mvnrepository.com/artifact/com.alibaba/fastjson -->

<dependency>

<groupId>com.alibaba</groupId>

<artifactId>fastjson</artifactId>

<version>1.2.46</version>

</dependency>

<**dependency**>  
 <**groupId**>org.json</**groupId**>  
 <**artifactId**>json</**artifactId**>  
 <**version**>20140107</**version**>  
</**dependency**>

**2. 获取access\_token**

package com.wsk.netty.check;

import org.json.JSONObject;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

import java.util.List;

import java.util.Map;

/\*\*

\* 获取token类

\*

\* @Author : WuShukai

\* @Date :2018/2/12 10:04

\*/

public class AuthService {

/\*\*

\* 获取权限token

\* @return 返回示例：

\* {

\* "access\_token": "24.460da4889caad24cccdb1fea17221975.2592000.1491995545.282335-1234567",

\* "expires\_in": 2592000

\* }

\*/

public static String getAuth() {

// 官网获取的 API Key 更新为你注册的

String clientId = "\*\*";

// 官网获取的 Secret Key 更新为你注册的

String clientSecret = "\*\*";

return getAuth(clientId, clientSecret);

}

/\*\*

\* 获取API访问token

\* 该token有一定的有效期，需要自行管理，当失效时需重新获取.

\* @param ak - 百度云官网获取的 API Key

\* @param sk - 百度云官网获取的 Securet Key

\* @return assess\_token 示例：

\* "24.460da4889caad24cccdb1fea17221975.2592000.1491995545.282335-1234567"

\*/

private static String getAuth(String ak, String sk) {

// 获取token地址

String authHost = "https://aip.baidubce.com/oauth/2.0/token?";

String getAccessTokenUrl = authHost

// 1. grant\_type为固定参数

+ "grant\_type=client\_credentials"

// 2. 官网获取的 API Key

+ "&client\_id=" + ak

// 3. 官网获取的 Secret Key

+ "&client\_secret=" + sk;

try {

URL realUrl = new URL(getAccessTokenUrl);

// 打开和URL之间的连接

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

connection.setRequestMethod("GET");

connection.connect();

// 获取所有响应头字段

Map<String, List<String>> map = connection.getHeaderFields();

// 遍历所有的响应头字段

for (String key : map.keySet()) {

System.err.println(key + "--->" + map.get(key));

}

// 定义 BufferedReader输入流来读取URL的响应

BufferedReader in = new BufferedReader(new InputStreamReader(connection.getInputStream()));

StringBuilder result = new StringBuilder();

String line;

while ((line = in.readLine()) != null) {

result.append(line);

}

/\*\*

\* 返回结果示例

\*/

System.err.println("result:" + result);

JSONObject jsonObject = new JSONObject(result.toString());

return jsonObject.getString("access\_token");

} catch (Exception e) {

System.err.printf("获取token失败！");

e.printStackTrace(System.err);

}

return null;

}

public static void main(String[] args) {

getAuth();

}

}

**3. 编写将图片转化成base64后再转化成urlencode的工具类**

package com.wsk.netty.check;

import sun.misc.BASE64Encoder;

import java.io.FileInputStream;

import java.io.IOException;

import java.io.InputStream;

import java.net.URLEncoder;

/\*\*

\* 图片转化base64后再UrlEncode结果

\* @Author : WuShukai

\* @Date :2018/2/12 10:43

\*/

public class BaseImg64 {

/\*\*

\* 将一张本地图片转化成Base64字符串

\* @param imgPath 本地图片地址

\* @return 图片转化base64后再UrlEncode结果

\*/

public static String getImageStrFromPath(String imgPath) {

InputStream in;

byte[] data = null;

// 读取图片字节数组

try {

in = new FileInputStream(imgPath);

data = new byte[in.available()];

in.read(data);

in.close();

} catch (IOException e) {

e.printStackTrace();

}

// 对字节数组Base64编码

BASE64Encoder encoder = new BASE64Encoder();

// 返回Base64编码过再URLEncode的字节数组字符串

return URLEncoder.encode(encoder.encode(data));

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41

.

**4. 编写调用百度API接口的方法，获取识别结果**

package com.wsk.netty.check;

import org.apache.http.HttpResponse;

import org.apache.http.client.HttpClient;

import org.apache.http.client.methods.HttpPost;

import org.apache.http.entity.StringEntity;

import org.apache.http.impl.client.DefaultHttpClient;

import org.apache.http.util.EntityUtils;

import java.io.File;

import java.io.IOException;

import java.net.URI;

import java.net.URISyntaxException;

/\*\*

\* 图像文字识别

\*

\* @Author : WuShukai

\* @Date :2018/2/12 10:25

\*/

public class Check {

private static final String POST\_URL = "https://aip.baidubce.com/rest/2.0/ocr/v1/general\_basic?access\_token=" + AuthService.getAuth();

/\*\*

\* 识别本地图片的文字

\*

\* @param path 本地图片地址

\* @return 识别结果，为json格式

\* @throws URISyntaxException URI打开异常

\* @throws IOException io流异常

\*/

public static String checkFile(String path) throws URISyntaxException, IOException {

File file = new File(path);

if (!file.exists()) {

throw new NullPointerException("图片不存在");

}

String image = BaseImg64.getImageStrFromPath(path);

String param = "image=" + image;

return post(param);

}

/\*\*

\* @param url 图片url

\* @return 识别结果，为json格式

\*/

public static String checkUrl(String url) throws IOException, URISyntaxException {

String param = "url=" + url;

return post(param);

}

/\*\*

\* 通过传递参数：url和image进行文字识别

\*

\* @param param 区分是url还是image识别

\* @return 识别结果

\* @throws URISyntaxException URI打开异常

\* @throws IOException IO流异常

\*/

private static String post(String param) throws URISyntaxException, IOException {

//开始搭建post请求

HttpClient httpClient = new DefaultHttpClient();

HttpPost post = new HttpPost();

URI url = new URI(POST\_URL);

post.setURI(url);

//设置请求头，请求头必须为application/x-www-form-urlencoded，因为是传递一个很长的字符串，不能分段发送

post.setHeader("Content-Type", "application/x-www-form-urlencoded");

StringEntity entity = new StringEntity(param);

post.setEntity(entity);

HttpResponse response = httpClient.execute(post);

System.out.println(response.toString());

if (response.getStatusLine().getStatusCode() == 200) {

String str;

try {

/\*读取服务器返回过来的json字符串数据\*/

str = EntityUtils.toString(response.getEntity());

System.out.println(str);

return str;

} catch (Exception e) {

e.printStackTrace();

return null;

}

}

return null;

}

public static void main(String[] args) {

String path = "E:\\find.png";

try {

long now = System.currentTimeMillis();

checkFile(path);

checkUrl("https://gss3.bdstatic.com/-Po3dSag\_xI4khGkpoWK1HF6hhy/baike/c0%3Dbaike80%2C5%2C5%2C80%2C26/sign=08c05c0e8444ebf8797c6c6db890bc4f/fc1f4134970a304e46bfc5f7d2c8a786c9175c19.jpg");

System.out.println("耗时：" + (System.currentTimeMillis() - now) / 1000 + "s");

} catch (URISyntaxException | IOException e) {

e.printStackTrace();

}

}

}

.

**5. 识别结果（仅测试本地图片识别）**

**中文**



**结果：**

**结论**

这里是使用了Postman进行测试的，用IDEA控制台的话，返回的json不易读。   
从这里可以看出，耗时是1s，虽然识别率高，但是结果还是有那么的一些差距，例如识别结果的第五列，只返回了“我是逊尼”，而原图片的很大串没有识别出来。

**英文：**

**结果：**

**结论**

单识别英文的图片，效果还是比较满意的，耗时短，精准率高。

**中英文结合：**

**结果：**