第一步：获取ACCESS\_TOKEN

//获取token的接口地址

public final static String access\_token\_url = "https://api.weixin.qq.com/cgi-bin/token?" + "grant\_type=client\_credential&appid=APPID&secret=APPSECRET";

/\*\*

\* 获取access\_token

\*

\* @param appid

\* @param appsecret

\* @return

\*/

public String getAccess\_token(String appid, String appsecret) {

try {

String requestUrl = access\_token\_url.replace("APPID", appid).replace("APPSECRET", appsecret);

JSONObject jsonObject = httpRequst(requestUrl, "GET", null);

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

} catch (Exception e) {

return "errer";

}

}

/\*\*

\*

\* @Title: httpsRequest

\* @Description: 发送请求,返回JSONObject对象

\* @param requestUrl

\* @param requestMethod

\* @param outputStr

\* @return

\* @throws Exception

\*/

public static JSONObject httpsRequest(String requestUrl, String requestMethod, String outputStr) throws Exception {

JSONObject jsonObject = null;

try {

// 使用自定义的信任管理器

TrustManager[] tm = { new X509TrustManager() {

/\*\*

\* 检查客户端证书

\*/

@Override

public void checkClientTrusted(X509Certificate[] arg0, String arg1) throws CertificateException {

}

/\*\*

\* 检查服务器端证书

\*/

@Override

public void checkServerTrusted(X509Certificate[] arg0, String arg1) throws CertificateException {

}

/\*\*

\* 返回受信任的X509证书数组

\*/

@Override

public X509Certificate[] getAcceptedIssuers() {

return null;

}

} };

SSLContext sslContext = SSLContext.getInstance("SSL", "SunJSSE");

sslContext.init(null, tm, new java.security.SecureRandom());

SSLSocketFactory ssf = sslContext.getSocketFactory();

// 建立连接

URL url = new URL(requestUrl);

HttpsURLConnection conn = (HttpsURLConnection) url.openConnection();

conn.setSSLSocketFactory(ssf);

conn.setDoOutput(true);

conn.setDoInput(true);

// 设置请求方式

conn.setRequestMethod(requestMethod);

// 当outputStr不为null时,向输出流写数据

if (null != outputStr) {

OutputStream outputStream = conn.getOutputStream();

outputStream.write(outputStr.getBytes("UTF-8"));

outputStream.close();

}

// 获取输入流

InputStream inputStream = conn.getInputStream();

InputStreamReader inputStreamReader = new InputStreamReader(inputStream, "utf-8");

BufferedReader bufferedReader = new BufferedReader(inputStreamReader);

// 读取响应内容

StringBuffer buffer = new StringBuffer();

String str = null;

while ((str = bufferedReader.readLine()) != null) {

buffer.append(str);

}

// 关闭资源

bufferedReader.close();

inputStreamReader.close();

inputStream.close();

conn.disconnect();

jsonObject = JSONObject.fromObject(buffer.toString());

} catch (Exception e) {

throw new Exception("请求/解析失败");

}

return jsonObject;

}

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

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

第二步：生成小程序二维码工具类

import java.io.File;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStream;

import org.apache.http.HttpEntity;

import org.apache.http.HttpResponse;

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

import org.apache.http.entity.StringEntity;

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

import org.apache.http.message.BasicHeader;

import org.apache.http.protocol.HTTP;

/\*\*

\*

\* @ClassName: CreateImgUtil

\* @Description: 生成小程序二维码工具类

\* @author cheng

\* @date 2017年9月13日 上午10:34:59

\*/

@SuppressWarnings("deprecation")

public class CreateImgUtil {

/\*\*

\* 私有化构造函数，防止创建本工具类的实例

\*/

private CreateImgUtil() {

}

@SuppressWarnings({ "resource" })

public static String httpPostWithJSON(String url, String json, String id,String dir) throws Exception {

String result = null;

DefaultHttpClient httpClient = new DefaultHttpClient();

HttpPost httpPost = new HttpPost(url);

httpPost.addHeader(HTTP.CONTENT\_TYPE, "application/json");

StringEntity se = new StringEntity(json);

se.setContentType("application/json");

se.setContentEncoding(new BasicHeader(HTTP.CONTENT\_TYPE, "UTF-8"));

httpPost.setEntity(se);

HttpResponse response = httpClient.execute(httpPost);

if (response != null) {

HttpEntity resEntity = response.getEntity();

if (resEntity != null) {

InputStream instreams = resEntity.getContent();

File saveFile = new File(dir + id + ".png");

// 判断这个文件（saveFile）是否存在

if (!saveFile.getParentFile().exists()) {

// 如果不存在就创建这个文件夹

saveFile.getParentFile().mkdirs();

}

saveToImgByInputStream(instreams, dir, id + ".png");

}

}

httpPost.abort();

return result;

}

/\*

\* @param instreams 二进制流

\*

\* @param imgPath 图片的保存路径

\*

\* @param imgName 图片的名称

\*

\* @return 1：保存正常 0：保存失败

\*/

private static int saveToImgByInputStream(InputStream instreams, String imgPath, String imgName) {

int stateInt = 1;

if (instreams != null) {

try {

File file = new File(imgPath + imgName);// 可以是任何图片格式.jpg,.png等

FileOutputStream fos = new FileOutputStream(file);

byte[] b = new byte[1024];

int nRead = 0;

while ((nRead = instreams.read(b)) != -1) {

fos.write(b, 0, nRead);

}

fos.flush();

fos.close();

} catch (Exception e) {

stateInt = 0;

e.printStackTrace();

} finally {

try {

instreams.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

return stateInt;

}

}

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

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

第三步：生成携带参数的小程序码

// 依据公司信息生成个性化小程序码,返回小程序码名称

private String createImgByCompanyId(String companyId,String dir) {

// 获取token

String appid = "wx8b3f3454sdfsdfsdgdfgdf4tr3402c2d7728";

String appsecret = "c35b5a0b94ea1ece9c0rwerwerwer34wetwetre390468b3efed7";

String token = getAccess\_token(appid, appsecret);

// 生成小程序码接口url

String url = "https://api.weixin.qq.com/wxa/getwxacodeunlimit?access\_token=ACCESS\_TOKEN".replace("ACCESS\_TOKEN",

token);

// 二维码中信息

Map<String, Object> map = new HashMap<String, Object>();

map.put("path", "pages/index/index");// 你二维码中跳向的页面

map.put("scene", companyId);// 携带参数

String json = JSONUtils.toJSONString(map);

// 生成二维码

try {

CreateImgUtil.httpPostWithJSON(url, json, companyId,dir);

return companyId;

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return null;

}

---------------------

作者：大白能

来源：CSDN

原文：https://blog.csdn.net/qq\_28988969/article/details/77969365

版权声明：本文为博主原创文章，转载请附上博文链接！