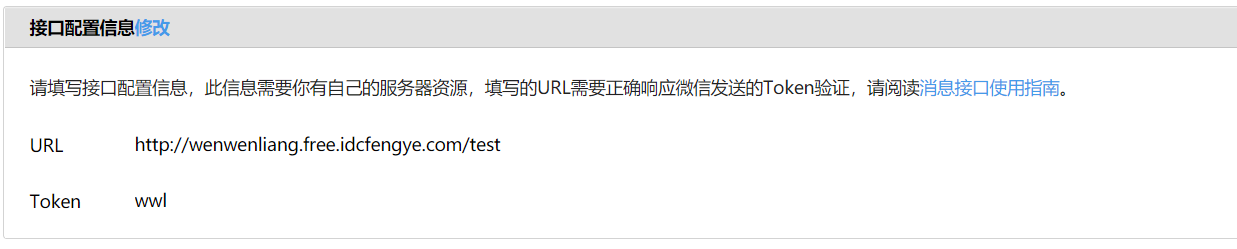
# 基本步骤

## 公众号接口配置信息配置(绑定接口)

说明 先验证绑定自己工程的接口；它会通过get方式发送信息进行验证，后续关注、接收用户发送来的事件都是通过此接口的post进行接收

url：<http://wenwenliang.free.idcfengye.com/test> 配置自己的工程servlet访问地址

Token ：wwl 验证



下面是代码：用来验证绑定用

下面有三步。进行绑定接口用的

public static final String TOKEN="wwl";//这里的常量token不能乱写要和微信公众号的token一致

/\*\*

\* 验证微信签名：当然嫌麻烦可以直接就返回echostr 即可和微信公众号进行绑定

\* @param signature 跟token、timestamp、nonce合起来进行解密后的结果一致，用于最后判断是否是微信发送过来的请求

\* @param timestamp 用于解密用

\* @param nonce 用于解密用

\* @param echostr 最后用于返回给微信，微信就会提示绑定成功

\* @return

\*/

@RequestMapping(value = "test")

public String test(String signature,String timestamp,String nonce,String echostr) {

//1) 将token、timestamp、nonce三个参数进行字典排序

String[] strs = {TOKEN,timestamp,nonce};

Arrays.sort(strs);

//2) 将三个参数字符串拼接成一个字符串再进行sho1加密

String str=strs[0]+strs[1]+strs[2];

//进行sha1加密

String mysig=sha1(str);

System.out.println("mysig:"+mysig); 返回来的就和signature一致

System.out.println(signature);

//3)前两步的动作都是把加密后的字符串跟同时传过来的signature进行对比，对比成功返回echostr即可成功绑定

if(signature.equals(mysig)) {

//验证成功了直接返回这个echostr，就会和微信公众号上的接口配置绑定成功

return echostr;

}else {

return "";

}

}

/\*\*

\* 进行sha1加密

\* @param str

\* @return

\*/

private String sha1(String src) {

//获取一个加密对象

try {//也可以传入字符串md5就是md5加密了

MessageDigest md= MessageDigest.getInstance("sha1");

//加密

byte[] digest = md.digest(src.getBytes());

char[] chars= {'0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f'};

StringBuilder sb = new StringBuilder();

//处理加密结果:一个byte有8位，我们把它分成两部分：前4位(1111刚好最大15可以转16进制)和后四位转成16进制

for(byte b:digest) {

//先处理前4位，防止有负数进行&

sb.append(chars[(b>>4)&15]);

sb.append(chars[b&15]);

}

return sb.toString();

} catch (NoSuchAlgorithmException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return null;

}

## 2、接收用户发送的信息

需要用到公众号配置的url，post请求方式，是以流的方式发送过来(事件的推送也是这样的，如关注等)

绑定成功后就可以通过公众号配置的url用port请求接收普通用户在公众号上发送过来的信息了

注意：使用springmvc方式的话可以再写一个和绑定微信公众号的url一致的地址，但是要变成post请求方式，后续入口就可以从这里接受微信信息

下面是接受用户信息的步骤方式

\* 接收的格式内容

\* <xml>

\* <ToUserName><![CDATA[gh\_f7b518bc167]]></ToUserName>

\* <FromUserName><![CDATA[oPJbu5\_MnohBqjnjxx6KgxXm9Iyw]]></FromUserName>

\* <CreateTime>1348831860</CreateTime>

\* <MsgType><![CDATA[text]]></MsgType>

\* <Content><![CDATA[this is a test]]></Content>

\* <MsgId>1234567890123456</MsgId>

\* </xml>

\* 说明

\* ToUserName 开发者微信号 ：就是自己开发的账号名称gh\_f7b518bc167

\* FromUserName 发送方帐号(OpenID)：oPJbu5\_MnohBqjnjxx6KgxXm9Iyw

\* CreateTime 消息创建时间： （整型）：默认是秒：1555823019

\* MsgType 消息类型，文本为text： 事件类型 可以用来进行判断以不同方式回复用户

\* Content 文本消息内容 ： 用户发送过来的信息

\* MsgId 消息id，64位整型 没啥用的

/\*\*

\* 接收用户发送过来的消息,

\* 是post方式以流(使用request接收)的方式推送过来，springmvc的那种方式不起作用：因为没有名字

\* @return

\*/

@PostMapping("test") //在公众号配置的url 用post请求接收发送信息

public void message(HttpServletRequest request) {

//使用Dom4j解析xml

SAXReader sr=new SAXReader();

try {

Document document = sr.read(request.getInputStream());

//获取根标签：xml

Element element = document.getRootElement();

List<Element> list = element.elements();

for (Element e : list) {

//节点名称

String string = e.getName();

//节点value

String value = e.getStringValue();

System.out.println(string+":"+value); 最后打印的内容就是：上面接收格式的内容，以字符串方式

}

} catch (DocumentException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

## 3、给用户回信息

前提：还是在第一条配置的url下进行操作

1、格式内容必须一致才行：多一个空格都不行，ToUserName和FromUserName跟接收用户信息是相反的

格式内容如下

<xml>

<ToUserName><![CDATA[toUser]]></ToUserName> 要回复用户的OpenId

<FromUserName><![CDATA[fromUser]]></FromUserName> 开发者的id

<CreateTime>12345678</CreateTime> 时间默认是秒，java是毫秒所以还要处理

<MsgType><![CDATA[text]]></MsgType> 发送的类型

<Content><![CDATA[你好]]></Content> 要回复用户的内容

</xml>

开发步骤

try {

long currentTimeMillis = System.currentTimeMillis()/1000;

System.out.println(currentTimeMillis);

response.setCharacterEncoding("utf-8");

//要发送的信息字符串

String str=

"<xml><ToUserName><![CDATA[oPJbu52RA5ikoxOp2lPllfu0u-\_U]]></ToUserName><FromUserName><![CDATA[gh\_f7b518bc167e]]></FromUserName><CreateTime>"+currentTimeMillis+"</CreateTime><MsgType><![CDATA[text]]></MsgType><Content><![CDATA[你好]]></Content></xml>";

使用response发送

PrintWriter writer = response.getWriter();

writer.write(str);

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

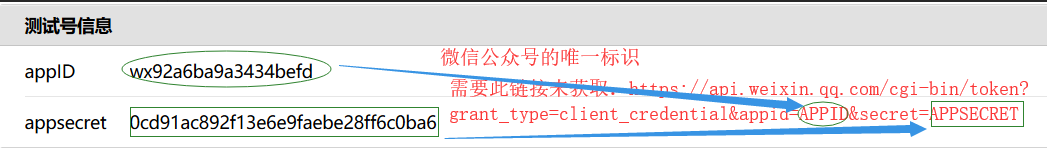
}

# 使用微信开放接口操作

## 第一步 需要access\_token来验证连接接口

### 获取access\_token

第一步：需要微信公众号的appID和appsecret



第二步 介绍获取注意事项

1、获取的是有时间限制的：获取的是一个数字：7200秒（2小时）

2、获取access\_token每天微信是有次数限制的

3、获取的access\_token每次值都不同，每一次获取的，上一次的就会失效，所以需要自己进行保存此值

4、使用java的类java.net.URL方式进行请求： https://api.weixin.qq.com/cgi-bin/token?grant\_type=client\_credential&appid=APPID&secret=APPSECRET

获取的内容是json格式：{"access\_token":"20\_3EYUBL6GMt36Op","expires\_in":7200}

第三步：获取步骤

这里使用springboot方式操作的

步骤一 使用java.net.URL来请求地址获取微信端发送过来的json串内容

get请求

public static String get(String urlPath) { //这个是要请求的地址

try {

URL url = new URL(urlPath);

URLConnection urlConnection = url.openConnection();

InputStream inputStream = urlConnection.getInputStream();

BufferedReader br = new BufferedReader(new InputStreamReader(inputStream));

String readLine = "";

StringBuilder sb = new StringBuilder();

while ((readLine = br.readLine()) != null) {

sb.append(readLine);

}

return sb.toString();

} catch (MalformedURLException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

return null;

}

步骤二 进行获取操作和进行封装成实体类

封装实体类：因为返回来的有两个属性：access\_token和expires\_in，还有一个进行计算有效时间

public class AccessToken {

@JsonProperty("access\_token") //此注解是jackson的注解类：要把json内容封装，名称要统一

private String accessToken; //token值

@JsonProperty("expires\_in")

private long expiresIn;

private long expiresTime; //有效期间内

保存微信提供的access\_token：微信说明：需要此凭证来调用接口：每天有限制的,需自己进行保存值(不然再次去取又token值又会变化)，两小时有效

// 获取token凭证的url地址

private static final String GET\_TOKEN\_URL = "https://api.weixin.qq.com/cgi-bin/token?grant\_type=client\_credential&appid=APPID&secret=APPSECRET";

// token对象

private static AccessToken accessToken; //此实体类3个属性：access\_token、expires\_in、还有一个算出的有效时间

private static void getToken() {

try {

// 根据微信说明获取access\_token凭证需要公众号的id和APPSECRE

// url地址：https://api.weixin.qq.com/cgi-bin/token?grant\_type=client\_credential&appid=APPID&secret=APPSECRET

// 进行替换成自己公众号的appid和appsecret，使用字符串替换方法replace()

String tokenUrl = GET\_TOKEN\_URL.replace("APPID", APP\_ID).replace("APPSECRET", APPSECRET);

String jsonToken = get(tokenUrl); //调用步骤一的方法请求微信地址 获取String：内容：{"access\_token":"20\_3EYUBL6GMt36Op","expires\_in":7200}

ObjectMapper om = new ObjectMapper(); //使用jackson对象

accessToken = om.readValue(jsonToken, AccessToken.class);//把json内容封装实体类中

accessToken.setExpiresTime(System.currentTimeMillis()+accessToken.getExpiresIn()\*1000); //算出有效时间：保存到实体类中

} catch (Exception e) {

e.printStackTrace();

}

}

步骤三 获取操作

获取ACCESSTOKEN ：因为有效时期两个小时，如果为null调用get()方法获取，有直接取自己设置保存的ACCESSTOKEN

public static String getAccessToken() {

//如果为null或空

if(accessToken ==null || isExpiresIn() ) { // isExpiresIn()

getToken();

}

return accessToken.getAccessToken();

}

判断accessToken是否还在有效期内

private static boolean isExpiresIn() {

long timeMillis = System.currentTimeMillis();

long time = accessToken.getExpiresTime();

System.out.println("当前时间调用时间timeMillis:"+timeMillis);

System.out.println("有效时间time:"+time);

return timeMillis>accessToken.getExpiresTime();

}

步骤三 测试

在公众号配置的url中用post请求调用getAccessToken()即可 ：这里怕自己不会调用要清楚：每天调用有限的，在调用微信提供接口才去调用它

## 第二步 使用微信提供的接口

### 1、获取用户信息步骤

#### 1.1获取用户详细内容

{

"subscribe":1, 值为0时，代表此用户没有关注该公众号，拉取不到其余信息。

"openid":"oPJbu52RA5ikoxOp2lPllfu0u-\_U", openid 用户的标识，对当前公众号唯一

"nickname":"简单枫", nickname 用户的昵称

"sex":1, 用户的性别，值为1时是男性，值为2时是女性，值为0时是未知

"language":"zh\_CN","city":"广州","province":"广东", "country":"中国", 用户的语言，简体中文为zh\_CN \所在省份\用户所在城市\用户所在国家

"headimgurl":"http:\/\/thirdwx.......",用户头像，最后一个数值代表正方形头像大小（有0、46、64、96、132数值可选，0代表640\*640正方形头像），用户没有头像时该项为空。

"subscribe\_time":1555824379, 用户关注时间，为时间戳。如果用户曾多次关注，则取最后关注时间

"remark":"",公众号运营者对粉丝的备注，公众号运营者可在微信公众平台用户管理界面对粉丝添加备注

"unionid": 测试号没有此字段：只有在用户将公众号绑定到微信开放平台帐号后，才会出现该字段。

"groupid":0, 用户所在的分组ID（兼容旧的用户分组接口）

"tagid\_list":[],用户被打上的标签ID列表

返回用户关注的渠道来源，ADD\_SCENE\_SEARCH 公众号搜索，ADD\_SCENE\_ACCOUNT\_MIGRATION 公众号迁移，ADD\_SCENE\_PROFILE\_CARD 名片分享，ADD\_SCENE\_QR\_CODE 扫描二维码，ADD\_SCENEPROFILE LINK 图文页内名称点击，ADD\_SCENE\_PROFILE\_ITEM 图文页右上角菜单，ADD\_SCENE\_PAID 支付后关注，ADD\_SCENE\_OTHERS 其他

"subscribe\_scene":"ADD\_SCENE\_PROFILE\_CARD",

"qr\_scene":0, 二维码扫码场景（开发者自定义）

"qr\_scene\_str":"" 二维码扫码场景描述（开发者自定义）

}

#### 1.2获取操作

用户关注了微信公众号获取用户信息操作：

要求

1、需要access\_token 获取方法：[跳转](#_获取access_token)

2、需要用户的openid

3、get方式调用url：https://api.weixin.qq.com/cgi-bin/user/info?access\_token=ACCESS\_TOKEN&openid=OPENID&lang=zh\_CN就会返回xml格式用户信息

private String getMessageinfo(HttpServletRequest request) {

String url="https://api.weixin.qq.com/cgi-bin/user/info?access\_token=ACCESS\_TOKEN&openid=OPENID&lang=zh\_CN";

String resultUrl = url.replace("ACCESS\_TOKEN", Utils.getAccessToken()).replace("OPENID", openId);//使用字符串方法替换内容

String str= Utils.get(resultUrl);

System.out.println(str);//返回来的内容看上面

}

用户未关注授权进入自己的网页获取操作：

需求：

1、需要access\_token 此时的access\_token是不一样的，需要获取网页授权的access\_token

2、需要用户的openid

3、公众号唯一标识appID：wx92a6ba9a3434befd appsecret：0cd91ac892f13e6e9faebe28ff6c0ba6

官方步骤：

1 第一步：用户同意授权，获取code

2 第二步：通过code换取网页授权access\_token

3 第三步：刷新access\_token（如果需要）

4 第四步：拉取用户信息(需scope为 snsapi\_userinfo)

5 附：检验授权凭证（access\_token）是否有效

详细步骤

第一步 引导用户跳转到此页面：如用户发送消息：’登录’两个字，这里给他发送一个a标签链接给他：

<a>

https://open.weixin.qq.com/connect/oauth2/authorize?appid=APPID&redirect\_uri=REDIRECT\_URI&response\_type=code&scope=SCOPE&state=STATE#wechat\_redirect

</a>

url参数说明：

appid 公众号的唯一标识

redirect\_uri 授权后重定向的回调链接地址， 请使用 urlEncode 对链接进行处理

response\_type 返回类型，请填写code

scope 应用授权作用域，两个值：snsapi\_base （不弹出授权页面，直接跳转，只能获取用户openid）

snsapi\_userinfo （弹出授权页面，可通过openid拿到昵称、性别、所在地。并且， 即使在未关注的情况下，只要用户授权，也能获取其信息 ）

state 不是必须的可以删掉 重定向后会带上state参数，开发者可以填写a-zA-Z0-9的参数值，最多128字节

#wechat\_redirect 是 无论直接打开还是做页面302重定向时候，必须带此参数 不用给值

第二步 需要在后台进行配置你需要跳转的网页也就是参数redirect\_uri的值

这里要注意：提示“已停止访问该网页”证明是连接成功的，只是此域名被微信封掉了



第三步 用户点击跳转成功后会有参数code返回(redirect\_uri/?code=CODE&state=STATE)，获取此返回的参数

String code=request.getParameter(“code”);

System.out.println(code);//内容就是英文字母和数字组合的内容

第四步 通过code值获取access\_token(注意和之前公众号的是不一样的，也是2小时有效，当然也需要自己保存操作)

//获取授权网页access\_token请求地址：使用get方式请求以下url

String ulr=” https://api.weixin.qq.com/sns/oauth2/access\_token?appid=APPID&secret=SECRET&code=CODE&grant\_type=authorization\_code”;

参数说明：

appID：wx92a6ba9a3434befd 公众号唯一标识

secret 就是appsecret：0cd91ac892f13e6e9faebe28ff6c0ba6 公众号的appsecret

code 第三步 获取的code值

grant\_type：默认不变

其实呢步骤和获取公众号access\_token一致只是url变了 请看[获取access\_token](#_获取access_token)

获取到json内容如下：

{

"access\_token":"ACCESS\_TOKEN", 网页授权接口调用凭证,注意：此access\_token与基础支持的access\_token不同

"expires\_in":7200, access\_token接口调用凭证超时时间，单位（秒）

"refresh\_token":"REFRESH\_TOKEN", 用户刷新access\_token

"openid":"OPENID", 用户唯一标识，请注意，在未关注公众号时，用户访问公众号的网页，也会产生一个用户和公众号唯一的OpenID

"scope":"SCOPE" 用户授权的作用域，使用逗号（,）分隔

}

第五步 (不是必须的) 刷新access\_token（如果需要）

由于access\_token拥有较短的有效期，当access\_token超时后，可以使用refresh\_token进行刷新，refresh\_token有效期为30天，当refresh\_token失效之后，需要用户重新授权。

https://api.weixin.qq.com/sns/oauth2/refresh\_token?appid=APPID&grant\_type=refresh\_token&refresh\_token=REFRESH\_TOKEN

第六步 拉取用户信息(需scope为 snsapi\_userinfo)

http：GET（请使用https协议） https://api.weixin.qq.com/sns/userinfo?access\_token=ACCESS\_TOKEN&openid=OPENID&lang=zh\_CN

access\_token 网页授权接口调用凭证,注意：此access\_token与基础支持的access\_token不同

openid 用户的唯一标识

lang 返回国家地区语言版本，zh\_CN 简体，zh\_TW 繁体，en 英语

返回的内容

就是用户的信息：看关注公众号获取信息内容：[跳转](#_1.1获取用户详细内容)