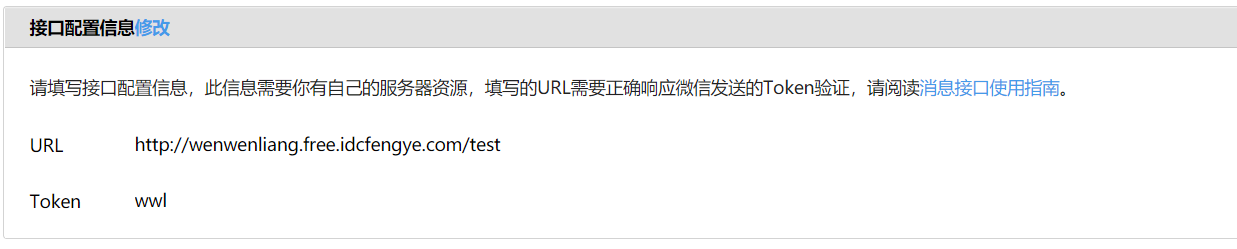
# 开发步骤

## 公众号接口配置信息配置(绑定接口)

说明 只是用来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

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，一次性的，用完就可以扔掉

下面有三步。进行绑定接口用的

//这里的常量token不能乱写要和微信公众号的token一致

public static final String TOKEN="wwl";

/\*\*

\* 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

\* @param timestamp

\* @param nonce

\* @param echostr

\* @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、接受用户发送的信息

绑定成功后就可以通过此接口接收普通用户在公众号上发送过来的信息了(post方式)HttpServlet的getPost进入

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

/\*\*

\* 接收用户在公众号上发送过来的消息

\* @return

\*/

@PostMapping("test") //springmvc只要请求不同，路径可以和get的test名称并存

public void message() {

System.out.println("用户发信息了");

}