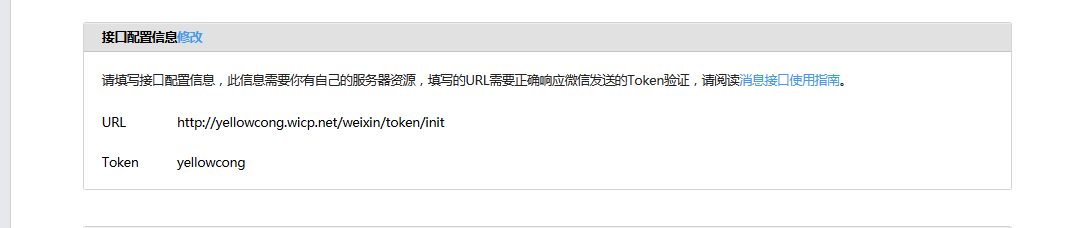
# 微信开发

## 微信接口的接入

首先要验证服务器的有效性，需要给服务器返回数据





服务器发送过来的数据

|  |
| --- |
| {"signature":"fea055de794b99a6b7cad667be3896d0cf9fed24","timestamp":"1458956459","nonce":"485291044","echostr":"3686687078633963515"} |

服务器验证

|  |
| --- |
| **package** com.yellowcong.weixin.utils;  **import** java.util.Arrays;  **import** com.yellowcong.action.TokenController;  **import** com.yellowcong.utils.StringUtil;  **import** com.yellowcong.weixin.model.Signature;  **public** **class** TokenUtils {  /\*\*  \* 验证服务器是否有效  \* **@param** signature 威信服务器发来的验证数据  \* **@param** token 我们自己的token  \* **@return**  \*/  **public** **static** **boolean** signature(Signature signature,String token){  //获取到了数据进行排序  String [] arrs = {token,signature.getNonce(),signature.getTimestamp()};  //排序  Arrays.*sort*(arrs);  //然后加密  StringBuffer sb = **new** StringBuffer();  **for**(String str:arrs){  sb.append(str);  }  //加密字符串  String sha1 =StringUtil.*sha1*(sb.toString());    //加密字符串  //如果验证正确，就返回他们的微信加密签名  **if**(signature.getSignature().equals(sha1)){  **return** **true**;  }**else**{  **return** **false**;  }  }  } |

## 获取Token

在测试中，这个appid和appsecret直接就给出来了



通过get请求来发送数据



发送get请求来获取我们的token

## 自定义菜单

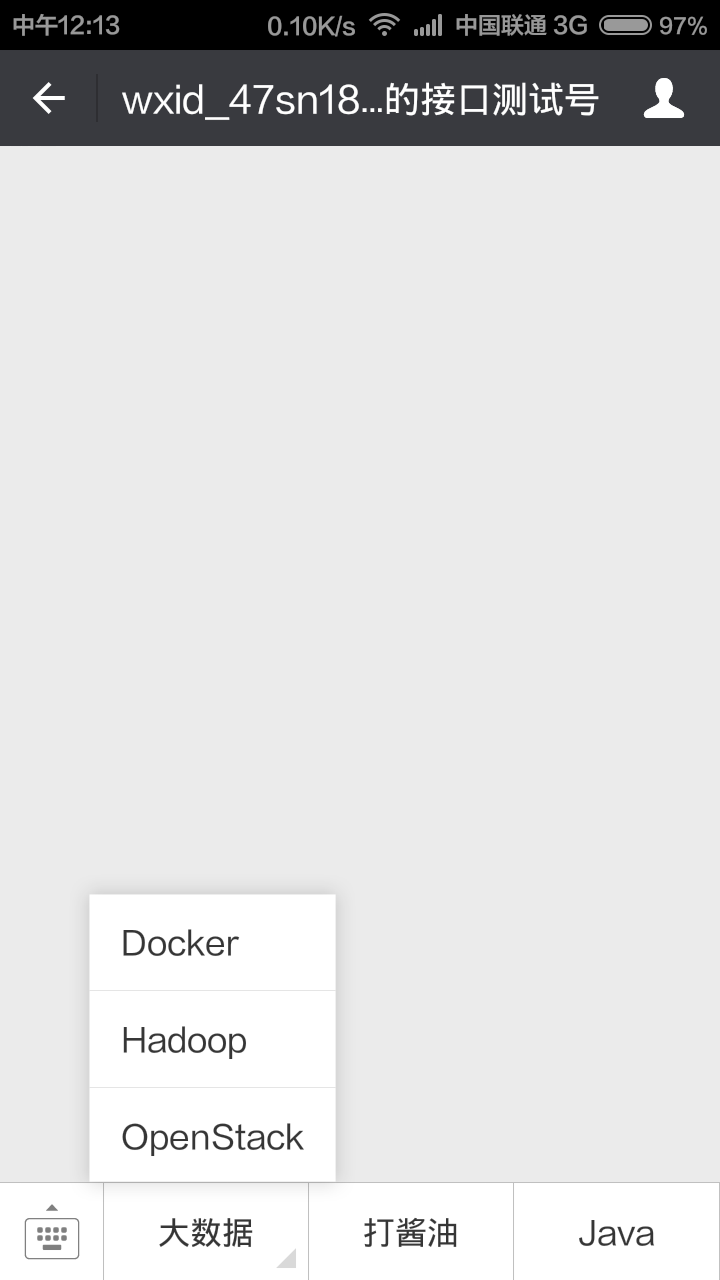
自定义菜单需要post发送json数据到服务器上



在自定义猜中中，click和view两个事件比较的多

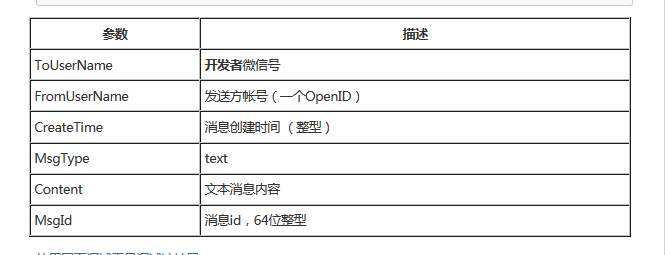
|  |
| --- |
| **1、click：点击推事件**  用户点击click类型按钮后，微信服务器会通过消息接口推送消息类型为event 的结构给开发者（参考消息接口指南），并且带上按钮中开发者填写的key值，开发者可以通过自定义的key值与用户进行交互；  **2、view：跳转URL**  用户点击view类型按钮后，微信客户端将会打开开发者在按钮中填写的网页URL，可与网页授权获取用户基本信息接口结合，获得用户基本信息。  **3、scancode\_push：扫码推事件**  用户点击按钮后，微信客户端将调起扫一扫工具，完成扫码操作后显示扫描结果（如果是URL，将进入URL），且会将扫码的结果传给开发者，开发者可以下发消息。  **4、scancode\_waitmsg：扫码推事件且弹出“消息接收中”提示框**  用户点击按钮后，微信客户端将调起扫一扫工具，完成扫码操作后，将扫码的结果传给开发者，同时收起扫一扫工具，然后弹出“消息接收中”提示框，随后可能会收到开发者下发的消息。  **5、pic\_sysphoto：弹出系统拍照发图**  用户点击按钮后，微信客户端将调起系统相机，完成拍照操作后，会将拍摄的相片发送给开发者，并推送事件给开发者，同时收起系统相机，随后可能会收到开发者下发的消息。  **6、pic\_photo\_or\_album：弹出拍照或者相册发图**  用户点击按钮后，微信客户端将弹出选择器供用户选择“拍照”或者“从手机相册选择”。用户选择后即走其他两种流程。  **7、pic\_weixin：弹出微信相册发图器**  用户点击按钮后，微信客户端将调起微信相册，完成选择操作后，将选择的相片发送给开发者的服务器，并推送事件给开发者，同时收起相册，随后可能会收到开发者下发的消息。  **8、location\_select：弹出地理位置选择器**  用户点击按钮后，微信客户端将调起地理位置选择工具，完成选择操作后，将选择的地理位置发送给开发者的服务器，同时收起位置选择工具，随后可能会收到开发者下发的消息。  **9、media\_id：下发消息（除文本消息）**  用户点击media\_id类型按钮后，微信服务器会将开发者填写的永久素材id对应的素材下发给用户，永久素材类型可以是图片、音频、视频、图文消息。请注意：永久素材id必须是在“素材管理/新增永久素材”接口上传后获得的合法id。**10、view\_limited：跳转图文消息URL**  用户点击view\_limited类型按钮后，微信客户端将打开开发者在按钮中填写的永久素材id对应的图文消息URL，永久素材类型只支持图文消息。请注意：永久素材id必须是在“素材管理/新增永久素材”接口上传后获得的合法id。 |

我们自己生成的数据



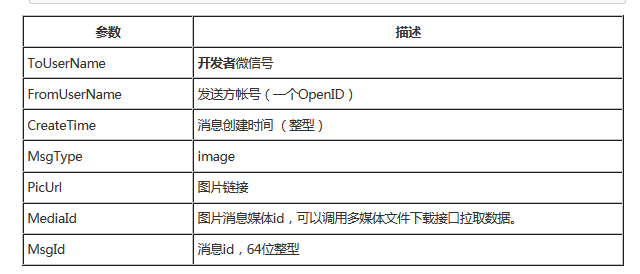
## 消息处理

### 文字消息



### 图片消息

接受的图片消息



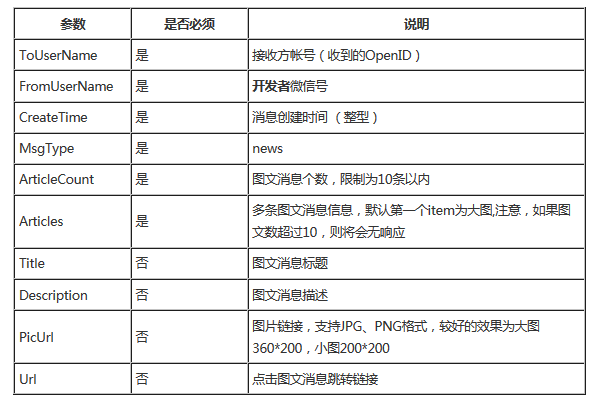
回复的图片消息

消息是Mediaid是，比发来的时候少 MsgId,PicUrl





### 图文消息



一张小图，两张大图的情况

大图 360x200

小图 200x200

下面的小图如果是360x200 会自动裁剪



只有一条消息的情况下，描述才会显示

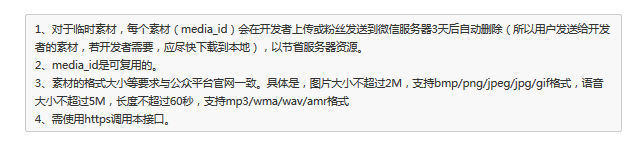


三个小图的效果，很不好

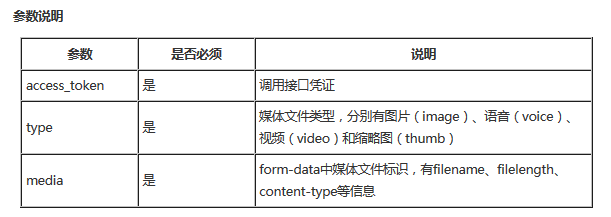


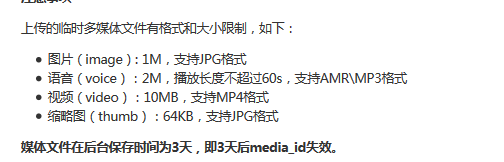
## 图灵机器人

## 素材管理



上传的类型有4中文件类型





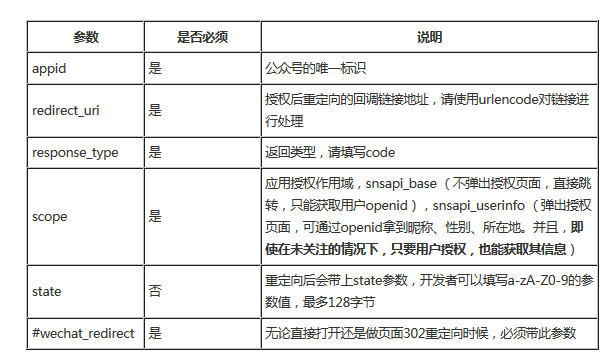
## 模板消息

## 微信组管理

## 用户关注和取消事件

# 微信二维码

## 用户授权



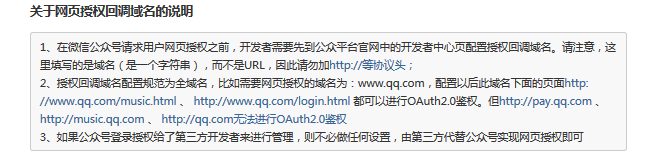


具体实现思路，是通过获取agent 来看是否是手机登录，然后获取openId



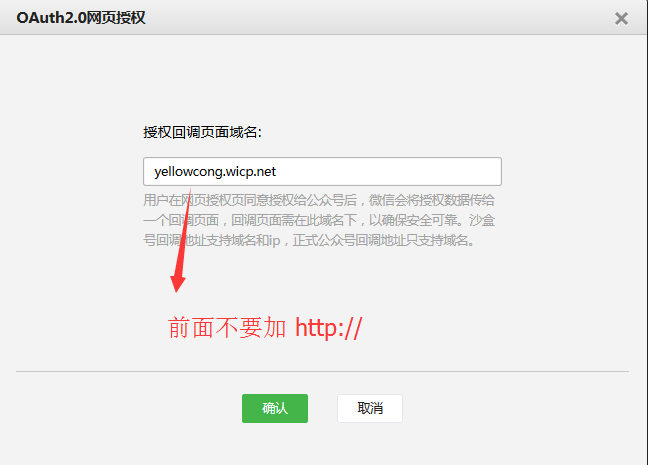
需要认证了，才会有权限操作

到开发者中心配置回调域名



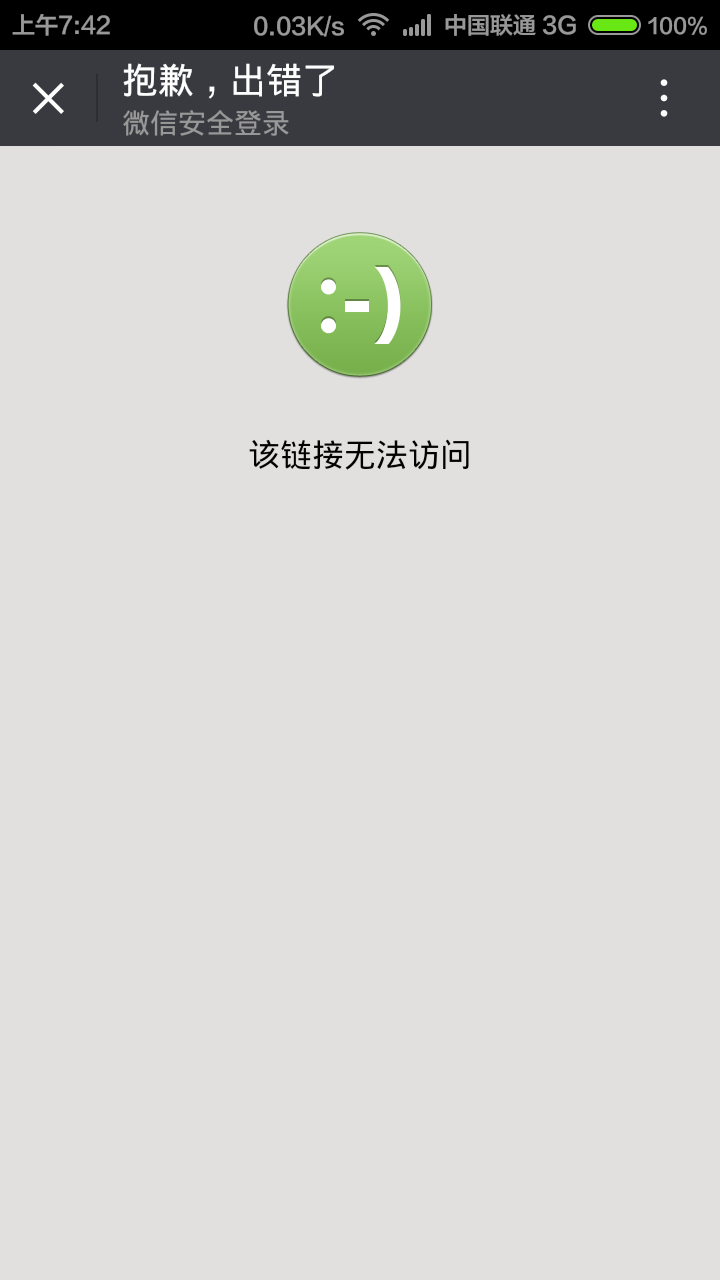
需要先设定好我们的域名





当配置好了域名后，我们的结果变了，我发现是配置的时候，配置了http://导致的错误，所以解决方法就是去掉http://





## 二维码绑定

二维码绑定的机理是

1. 获取是否是微信客户端
2. 发送请求到服务器，获取授权（需要到web上写上授权域名，不需要加上http://，而且只有认证用户在开发环境 中，才可以使用）
3. 获取到认证后，得到code,
4. 通过code,获取到用户的openid,这是一个唯一的id

5.将openid存入到用户的数据库中，就可以授权了

## 微信扫描登录

1. 生成二维码图片
2. 获取到ticket的url
3. 获取到扫描的事件
4. 在html中通过ajax 定时向我们的中间服务发送请求
5. 服务是扫描的，我们就可以将事件 forward 到 中间的服务中，然后获取到数据
6. 将用户数据存储到Session中
7. 通过href.localtion 来加载 数据