

微信开发入门 -- 第一步：介绍

本人学习了网上相关的视频教程，然后将老师讲的代码记录了下来。

微信公众号有 2 中模式，编辑模式和开发模式，这 2 种模式互斥。

注意：微信公众号的开发不是微信小程序的开发。

微信开发入门 -- 第二步：公众号配置

1.注册登陆

进入微信公众平台官网进行注册，然后登陆。官网地址: <https://mp.weixin.qq.com/>

2.账号设置

由于个人订阅号权限不够并且未认证，所以无法进行更多的微信接口的开发。这时进入左侧菜单的开发者工具，然后选择公共平台测试账号，用微信扫码登录进去。如下图所示：



3.开发设置

记住自己的 appID 和 appsecret 信息，这个信息在开发中会用到。

设置 URL 和 Token ， URL 是根据你的接口地址来填写的，格式为 : `http://域名/工程名/接口名`，就是自己接口的访问地址，工程名有的也可以不要，本工程就可以不用写工程名。Token 是自己随意输入的，在代码中会进行匹配来进行验证，这一步在当前阶段配置不了，需要你的程序启动后才能配置成功。这一步之后在进行配置。

4.外网映射工具

花生壳、Ngrok 或 Natapp，具体使用方法看百度。

我用的是 natapp，进入 natapp 目录下命令行执行 `natapp -authtoken=你的 xxx`。

与微信对接的 url 具备条件：

- 在公网上能访问
- http 端口为 80 端口

微信开发入门 -- 第三步：开发模式接入

1.工程搭建

本人比较懒，直接从 YouMeek 的 github 上 down 下 SSM 脚手架代码。你也可以自己搭建一个 SSM 工程。

2.接入接口

新建一个 `WeiXinController` 类，在类中添加与微信的对接接口 `wxrest`，写好 `get` 方法和 `post` 方法。`get` 方法为验证消息来自微信服务器，`post` 方法为验证通过后与微信的消息交互。
如图：

```

/**
 * 微信开发入门 一个人学习总结
 * Created by ZhaJunYong on 2018/4/3.
 */
@Controller
@RequestMapping("/weiXinController")
public class WeiXinController {
    private static final Logger logger = LoggerFactory.getLogger(WeiXinController.class);

    @RequestMapping(value = "/wxRest", method = RequestMethod.GET)
    @ResponseBody
    public String wxGet(@Context HttpServletRequest request, @Context HttpServletResponse response) {
        return "";
    }

    @RequestMapping(value = "/wxRest", method = RequestMethod.POST)
    @ResponseBody
    public void wxPost(@Context HttpServletRequest request, @Context HttpServletResponse response) {
        wxGet(request, response);
    }
}

```

3.get 方法编写

官方文档说的很详细，开发者通过检验 signature 对请求进行校验。若确认此次 GET 请求来自微信服务器，请原样返回 echostr 参数内容，则接入生效，成为开发者成功，否则接入失败。校验方法: 1) 将 token、timestamp、nonce 三个参数进行字典序排序 2) 将三个参数字符串拼接成一个字符串进行 sha1 加密 3) 开发者获得加密后的字符串可与 signature 对比，标识该请求来源于微信。

```

@RequestMapping(value = "/wxRest", method = RequestMethod.GET)
@ResponseBody
public String wxGet(@Context HttpServletRequest request, @Context HttpServletResponse response) {
    String signature = request.getParameter("signature");
    String timestamp = request.getParameter("timestamp");
    String nonce = request.getParameter("nonce");
    String echostr = request.getParameter("echostr");

    //校验参数
    logger.debug("| 开始校验参数 |");
    if (CheckUtil.checkSignature(signature, timestamp, nonce)) {
        logger.debug("| 校验成功 |");
        //将字符串返回给微信
        return echostr;
    }
    logger.debug("| 校验失败 |");
    return "";
}

```

校验工具类看源码。此时可以启动服务，映射外网地址，将地址写入 URL 中，将 CheckUtil 中的 Token 也写进去，确认成功。

4.post 方法编写

先获取微信服务器传过来的信息，如下图所示：

```
Map<String, String> map = MessageUtil.xmlToMap(request);
String fromUserName = map.get("FromUserName"); //发送方帐号（用户OpenID）
String toUserName = map.get("ToUserName"); //开发者微信号原始ID
String msgType = map.get("MsgType"); //text
String content = map.get("Content"); //文本消息内容
```

然后判断消息类型，根据消息类型进行相应的处理，保存数据内容或者是回复。以文本消息为例子，先判断消息类型为“text”类型，然后根据用户输入的内容进行相应回复，截图如下，回复消息的封装可以看源码。

```
String message = null;
if("text".equals(msgType)){
    //发送的消息为文本消息
    if("1".equals(content)){
        StringBuffer sb = new StringBuffer();
        sb.append("这是一个测试项目");
        message = MessageUtil.sendText(toUserName, fromUserName, sb.toString());
    }else if ("2".equals(content)){
        StringBuffer sb = new StringBuffer();
        sb.append("www.baidu.com");
        message = MessageUtil.sendText(toUserName, fromUserName, sb.toString());
    }else if("3".equals(content)){
        StringBuffer sb = new StringBuffer();
        sb.append("用于学习微信项目的开发");
        message = MessageUtil.sendText(toUserName, fromUserName, sb.toString());
    }else if("?".equals(content) || "? ".equals(content)){
        StringBuffer sb = new StringBuffer();
        sb.append("欢迎您的关注,请按照菜单提示操作:\n");
        sb.append("1:介绍\n");
        sb.append("2:百度网址\n");
        sb.append("3:关于本项目\n");
        sb.append("?帮助");
        message = MessageUtil.sendText(toUserName, fromUserName, sb.toString());
    }else {
        StringBuffer sb = new StringBuffer();
        sb.append("您发送的消息是:" + content);
        message = MessageUtil.sendText(toUserName, fromUserName, sb.toString());
    }
}
```

微信开发入门 -- 第四步：部分代码编写

1. 获取 access_token

这一步在官网文档说的很详细，通过 `get` 方式调用接口拿到返回数据即可。接口：

`https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid=APPID&secret=APPSECRET`。替换掉 APPID 和 APPSECRET 即可。具体代码在 `WXUtil` 类中。

2. 创建菜单

调用接口：https://api.weixin.qq.com/cgi-bin/menu/create?access_token=ACCESS_TOKEN。

先获取 `access_token`，然后替换掉上述网址的 `ACCESS_TOKEN`。

然后创建菜单，菜单格式在官方文档里面有，可以照着上面的格式创建自己的菜单。

用 `POST` 方式调用接口，把菜单数据传进去。菜单即可创建成功。代码中菜单初始化在 `menu` 包中，菜单按钮在 `pojo` 中 `button`、`ClickButton`、`ViewButton`、`Menu`。下图为创建菜单的方法，只需要传入 `token` 和 `menu` 即可。

```
/**
 * 创建菜单
 */
public static int createMenu(String token, String menu) {
    int result = 0;
    String url = CREATE_MENU_URL.replace("ACCESS_TOKEN", token);
    JSONObject jsonObject = doPostStr(url, menu);
    if (null != jsonObject) {
        result = jsonObject.getInt("errcode");
    }
    return result;
}
```

3. 网页授权

第一步：用户同意授权，获取 `code`

第二步：通过 `code` 换取网页授权 `access_token`

第三步：刷新 `access_token`（如果需要）

第四步：拉取用户信息(需 `scope` 为 `snsapi_userinfo`)

在本代码中，先写授权接口，如下图：

```

/**
 * 网页授权地址
 */
@RequestMapping(value = "/authUrl", method = RequestMethod.GET)
@ResponseBody
public void getAuthUrl(@Context HttpServletRequest req, @Context HttpServletResponse resp) throws IOException {
    //回调地址
    String backUrl = WXUtil.DOMAIN + "/auth/authCallback?pageUrl=/info.jsp";
    //授权地址
    String url = "https://open.weixin.qq.com/connect/oauth2/authorize"
        + "?appid=" + WXUtil.APPID
        + "&redirect_uri=" + backUrl
        + "&response_type=code"
        + "&scope=snsapi_userinfo" //snsapi_userinfo
        + "#wechat_redirect";
    resp.sendRedirect(url);
}

```

然后再写回调接口，如下图：

```

/**
 * 微信授权回调接口
 */
@RequestMapping(value = "/authCallback", method = RequestMethod.GET)
@ResponseBody
public void authCallback(@Context HttpServletRequest req, @Context HttpServletResponse resp) throws IOException, ServletException {
    //访问地址后调用回调函数,并且传递code
    String code = req.getParameter("code");
    String url = "https://api.weixin.qq.com/sns/oauth2/access_token"
        + "?appid=" + WXUtil.APPID
        + "&secret=" + WXUtil.APPSECRET
        + "&code=" + code
        + "&grant_type=authorization_code";
    JSONObject jsonObject = WXUtil.doGetStr(url);
    String openid = jsonObject.getString("openid");
    String token = jsonObject.getString("access_token");
    String infoUrl = "https://api.weixin.qq.com/sns/userinfo" + "?access_token=" + token + "&openid=" + openid + "&lang=zh_CN";

    //获取用户信息
    JSONObject basicInfo = WXUtil.doGetStr(infoUrl);
    String info = basicInfo.toString();

    String pageUrl = req.getParameter("pageUrl");
    System.out.println("pageUrl:" + pageUrl);
    //将用户信息保留在session中
    req.setAttribute("info", info);
    req.getRequestDispatcher(pageUrl).forward(req, resp);
}

```

在 index.jsp 中点击回调接口，就会执行页面授权过程，授权成功跳转到 info.jsp。