
NF
19
携手互联
无限可能

Niagara与微信的交互集成

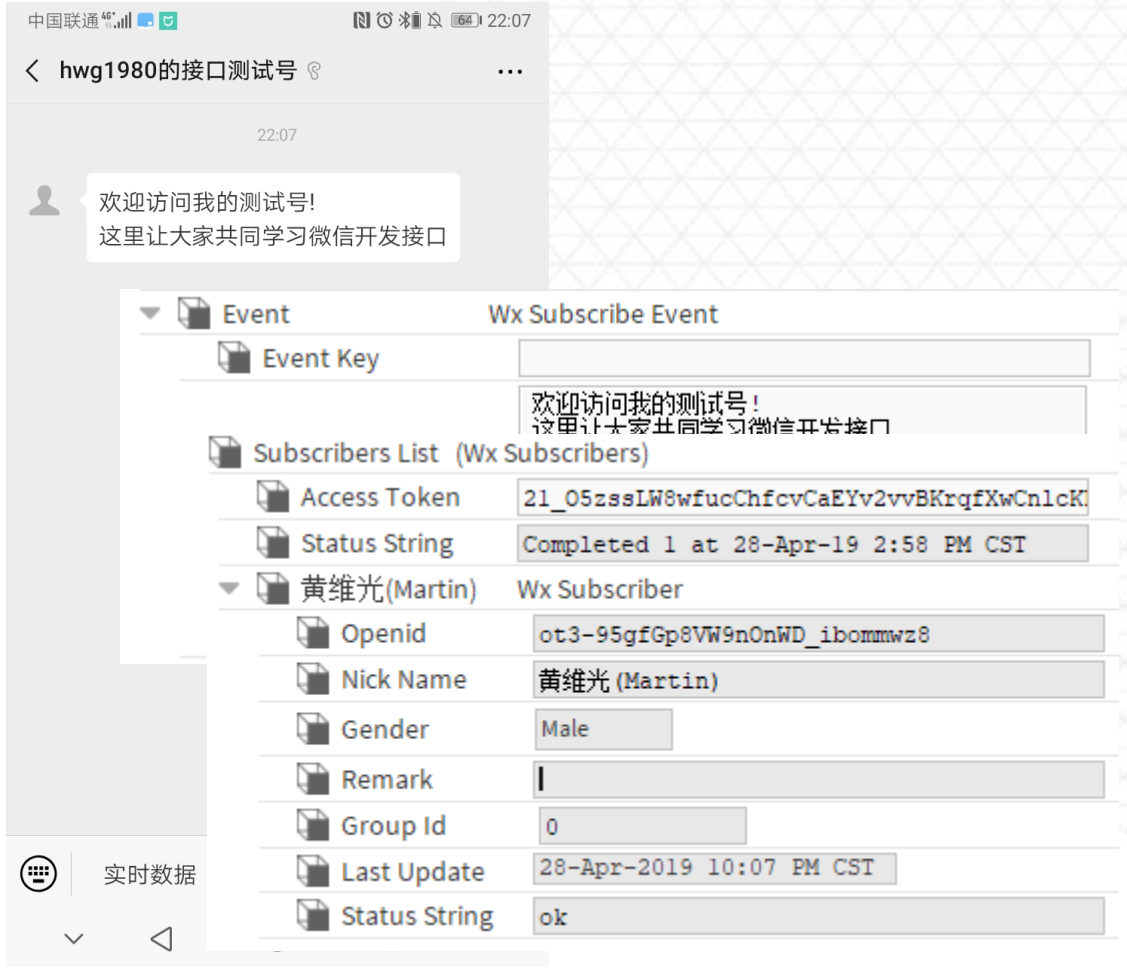
黄维光 (Martin)
Tridium

TRIDIUM

- 微信 + Niagara 能做什么?
- 开发需要什么?
- 实战

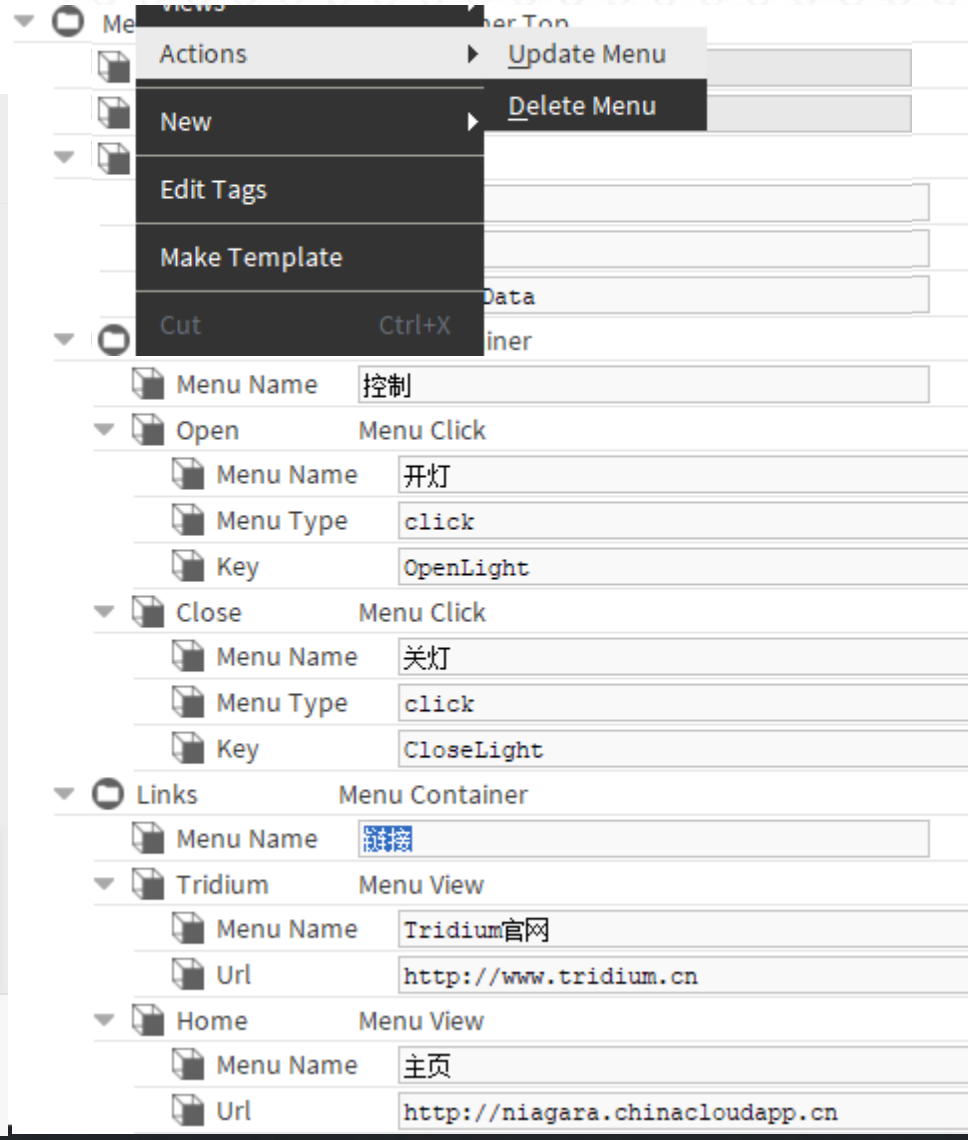
欢迎订阅

- 配置欢迎接口，内容可以定制
- 后台可以得到订阅者信息

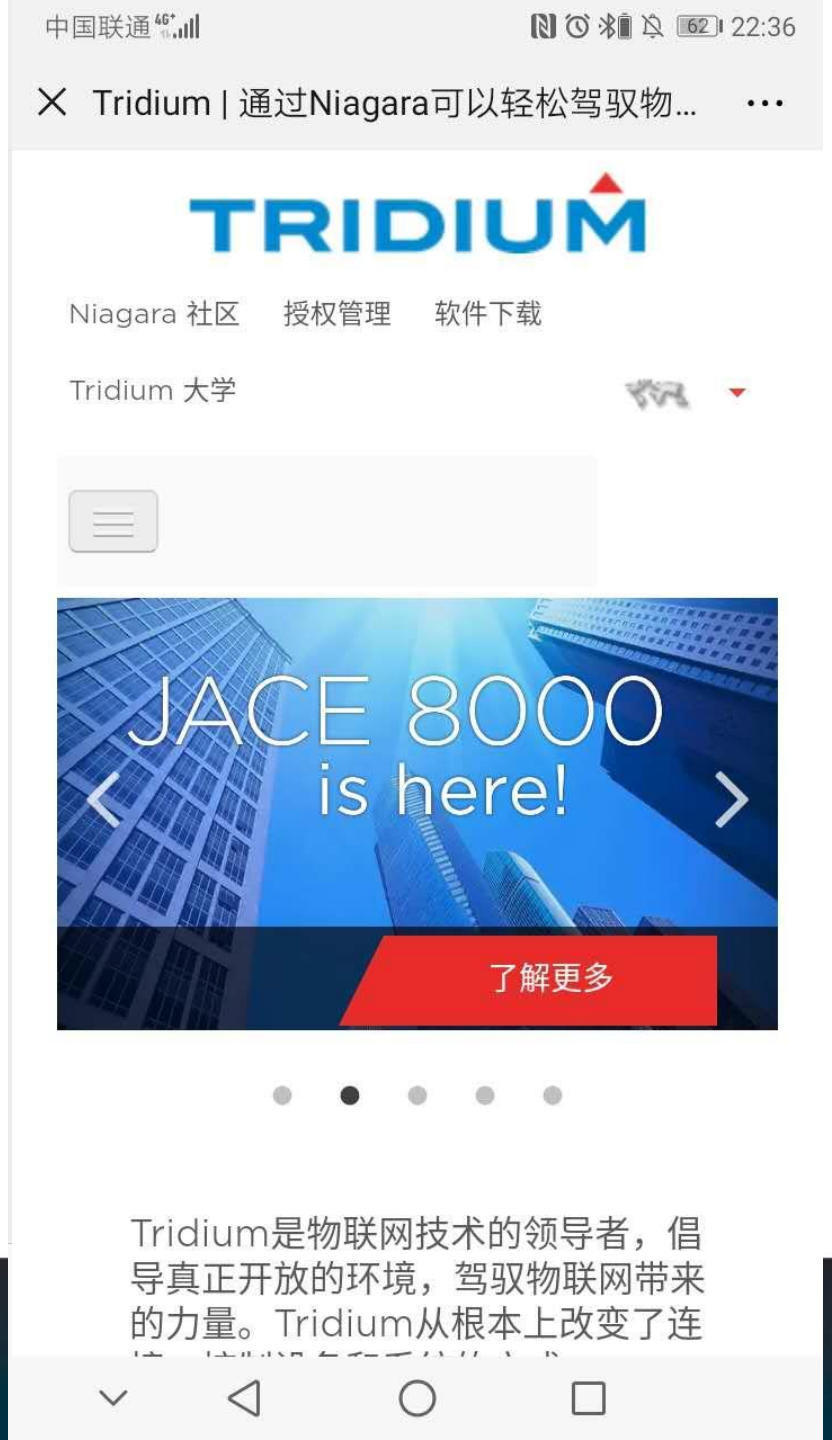


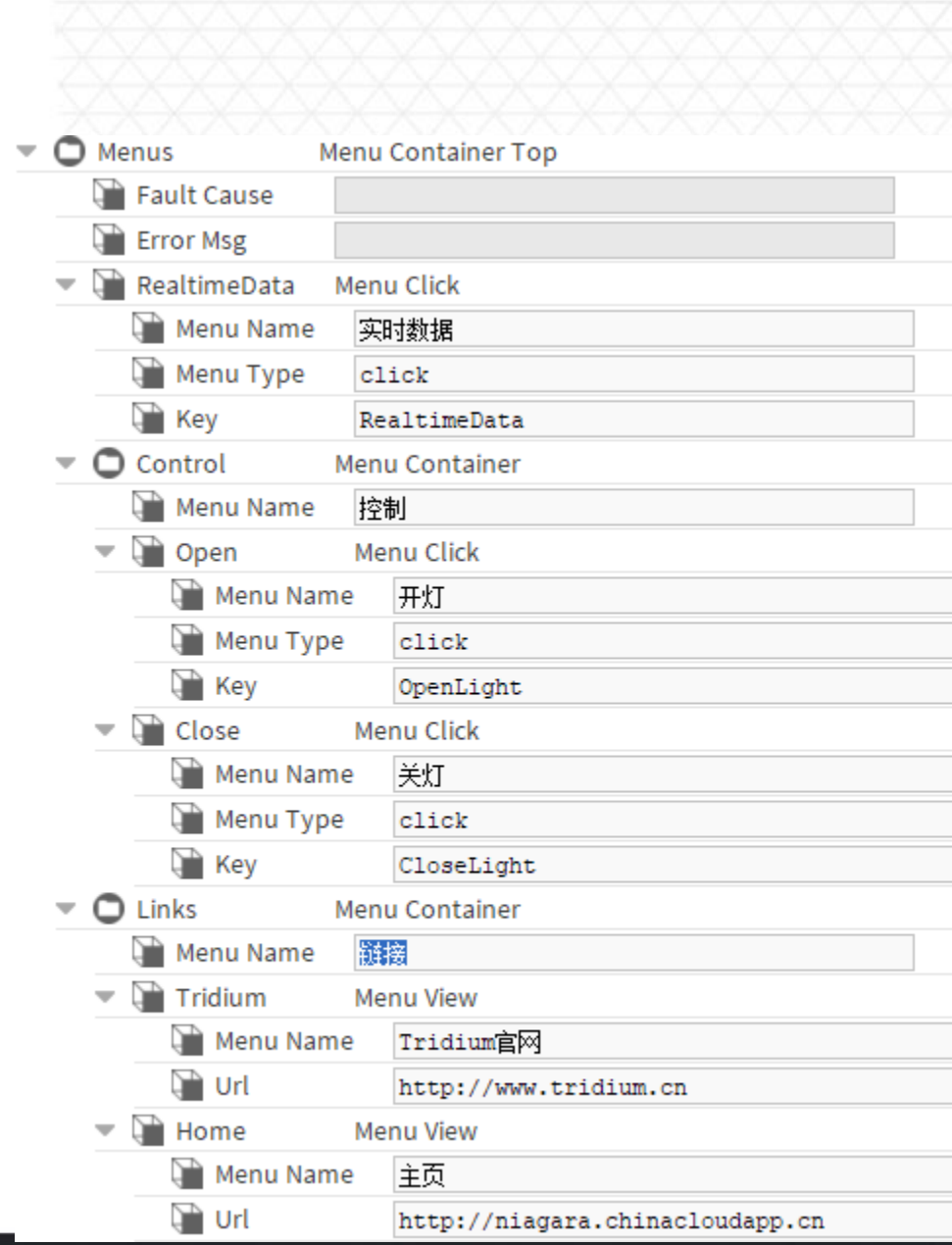
定制菜单

- Workbench中定制菜单



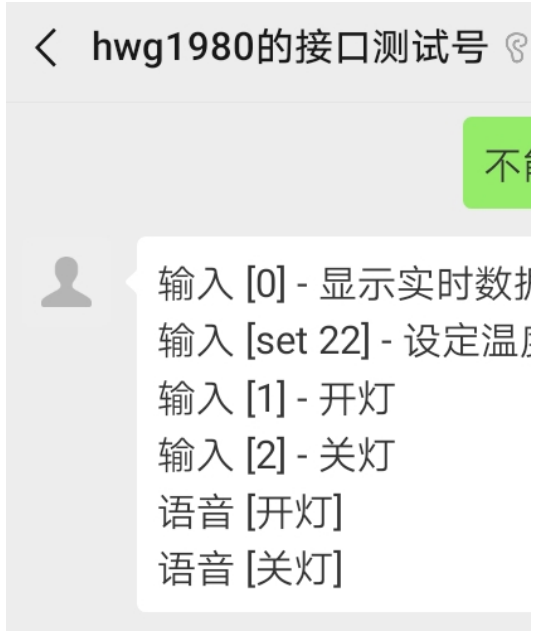
菜单交互

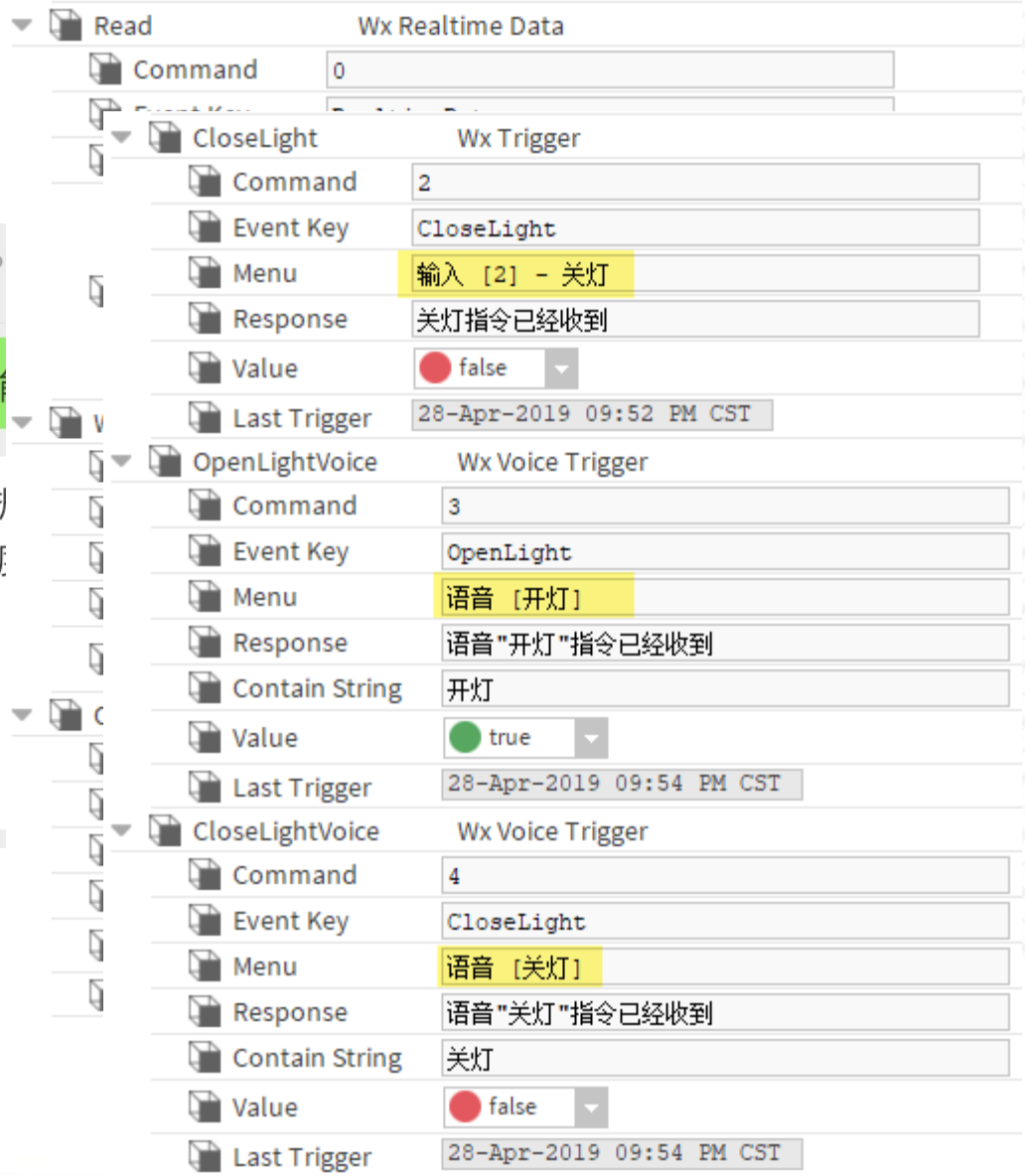

中国联通
22:36
Tridium | 通过Niagara可以轻松驾驭物...
TRIDIUM
Niagara 社区
授权管理
软件下载
Tridium 大学
JACE 8000
is here!
了解更多
Tridium是物联网技术的领导者，倡导真正开放的环境，驾驭物联网带来的力量。Tridium从根本上改变了连


Menus Menu Container Top
Fault Cause
Error Msg
RealtimeData Menu Click
Menu Name 实时数据
Menu Type click
Key RealtimeData
Control Menu Container
Menu Name 控制
Open Menu Click
Menu Name 开灯
Menu Type click
Key OpenLight
Close Menu Click
Menu Name 关灯
Menu Type click
Key CloseLight
Links Menu Container
Menu Name 链接
Tridium Menu View
Menu Name Tridium官网
Url http://www.tridium.cn
Home Menu View
Menu Name 主页
Url http://niagara.chinacloudapp.cn

文字交互 - 提示菜单

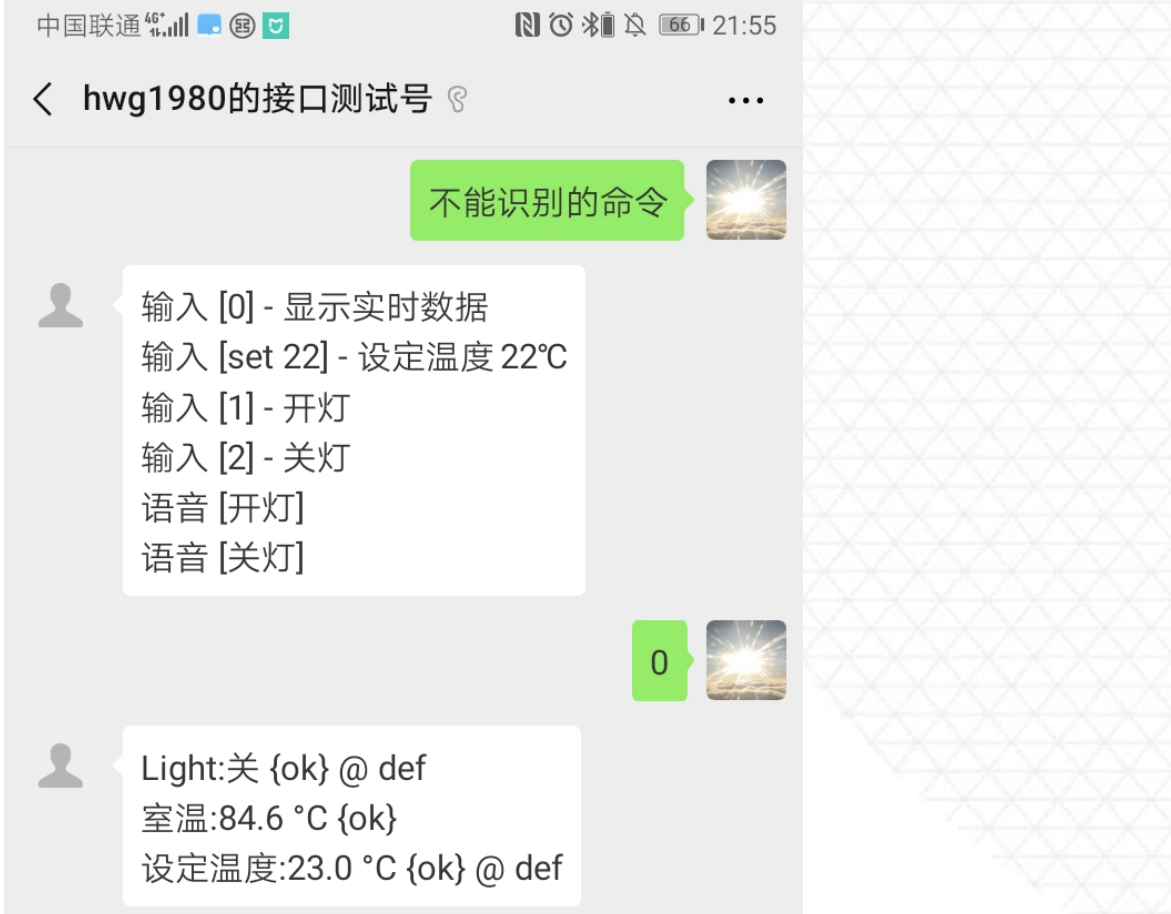
- 用户输入一个未配置的信息时，回复提示

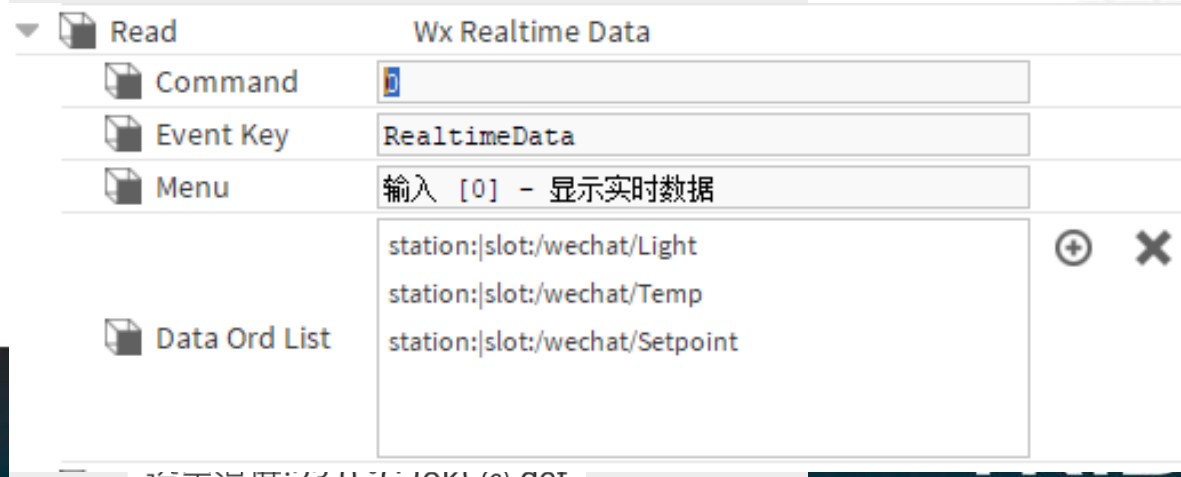




文字交互 - 实时数据

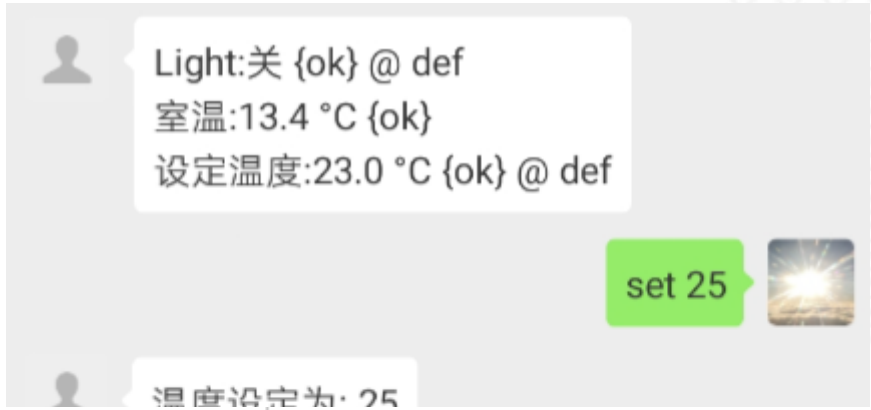
- 命令可配置
- 与菜单关联
- 数据可配置
- 数据订阅

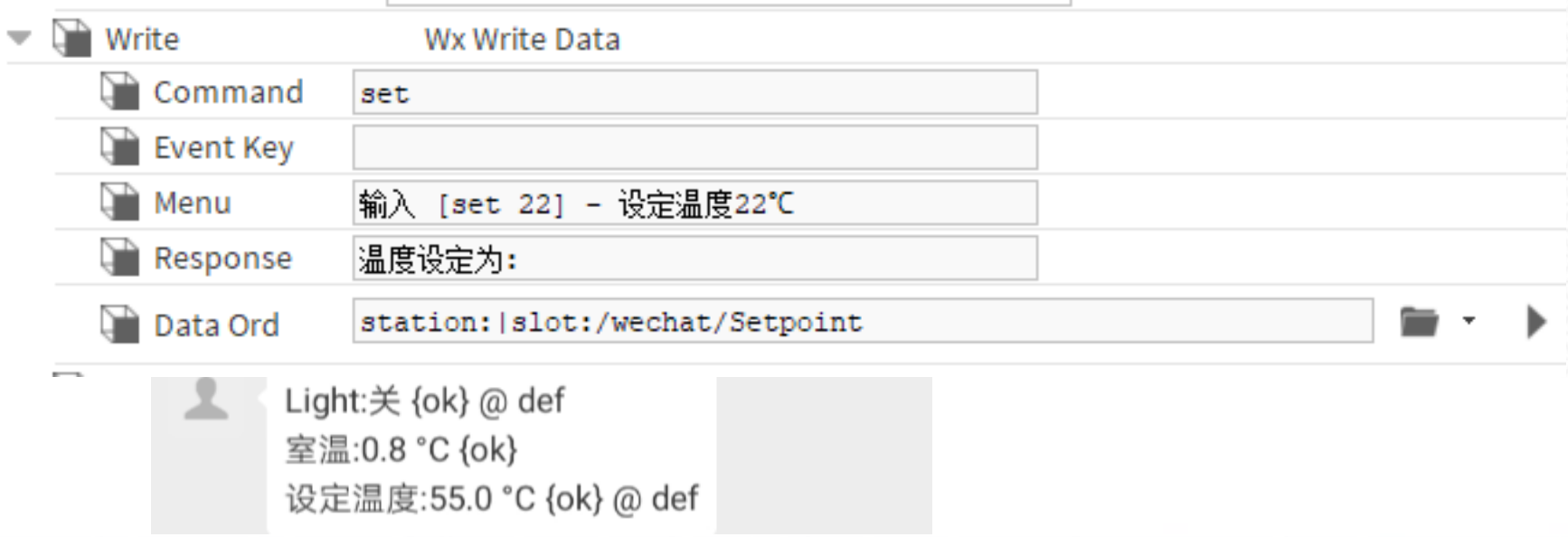




文字交互 - 数据设定

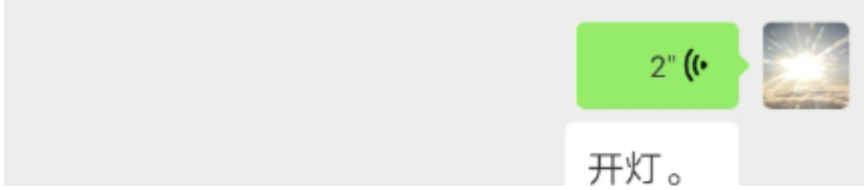
- 命令可配置
- 反馈信息可配置
- 数据可配置

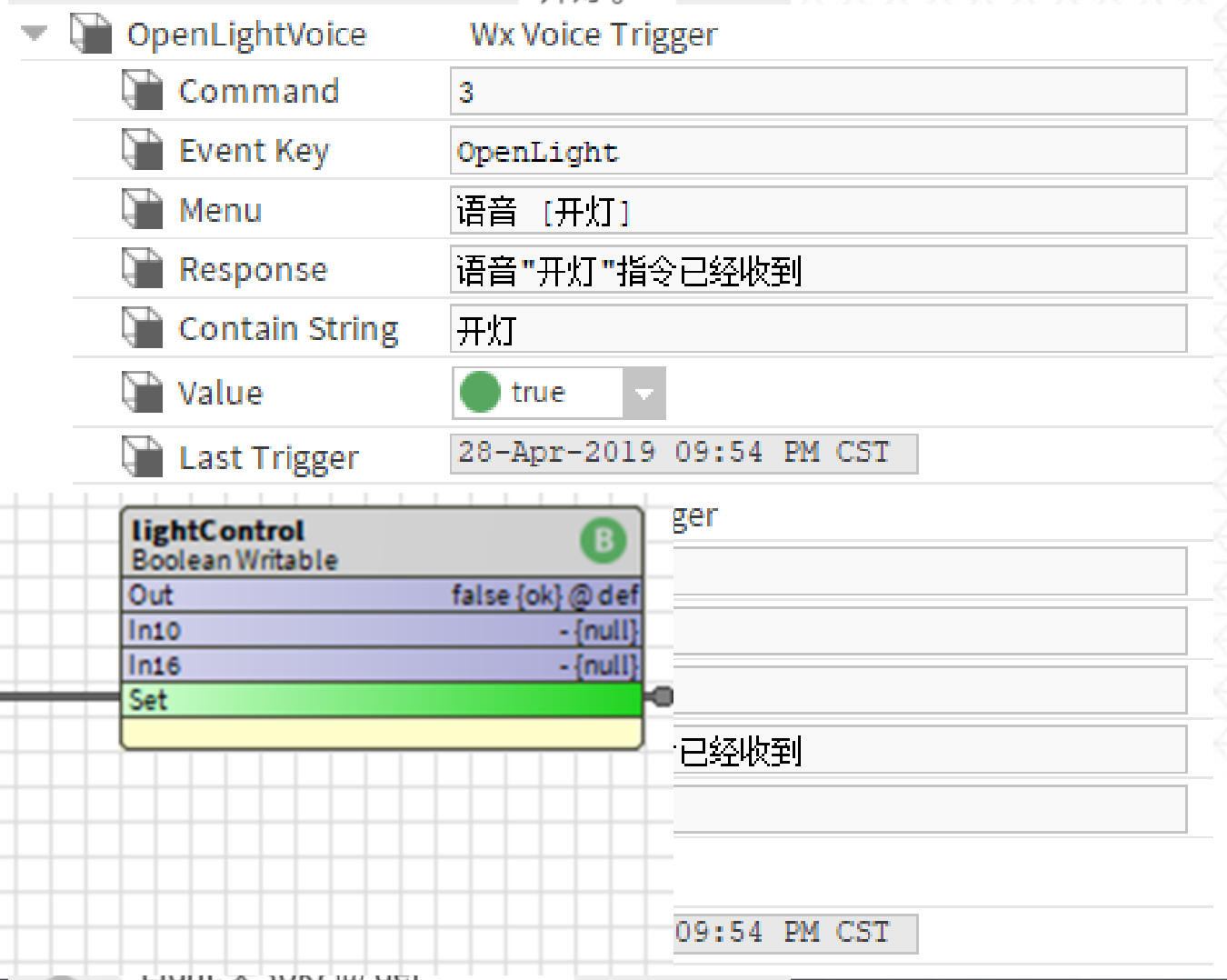




语音控制

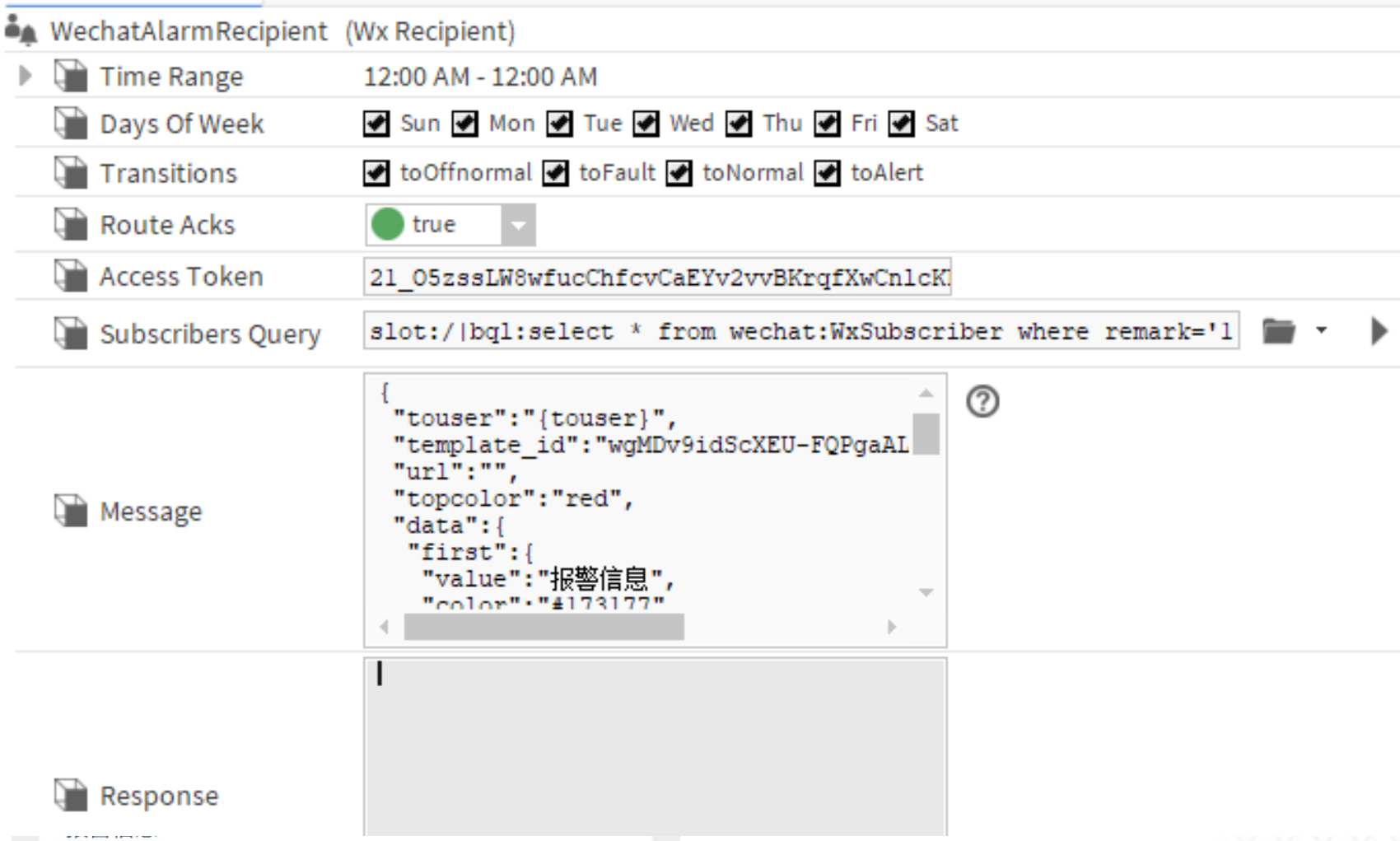
- 命令可配置
- 可与菜单关联
- 应答可配置
- 语音内容可配置





报警推送

- 报警格式可配置
- 报警模板可配置
- 接收人可选择



- 微信 + Niagara 能做什么?
- 开发需要什么?
- 实战

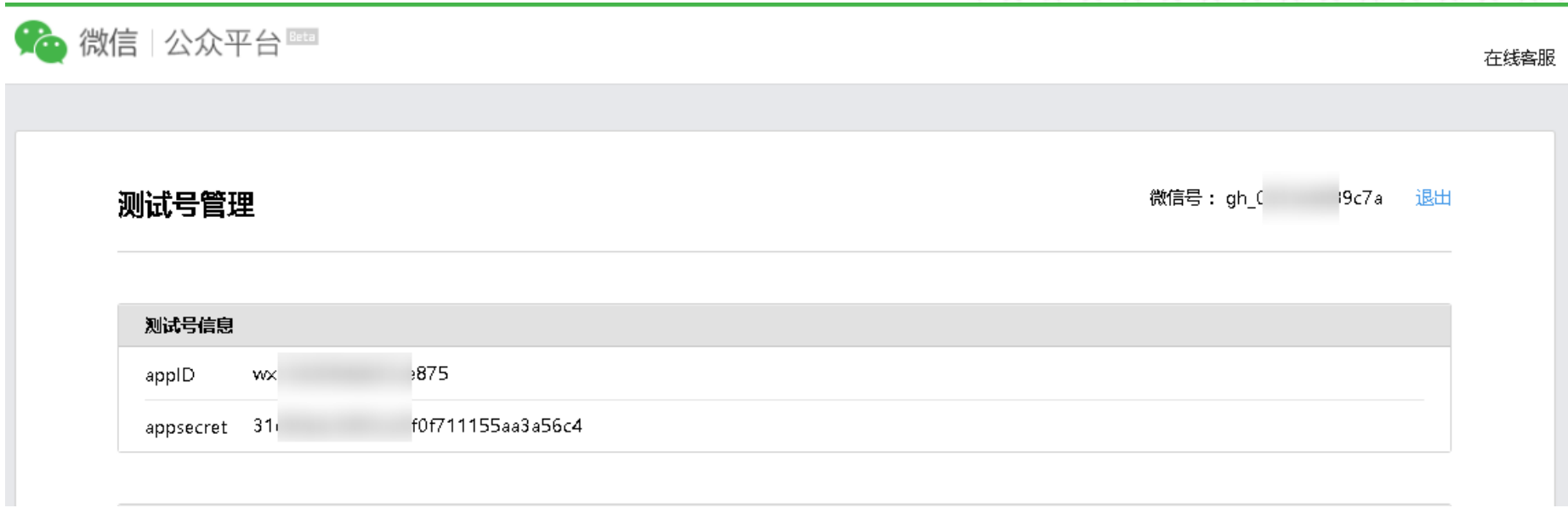
了解 Niagara 应用与开发

- 熟悉 Niagara 基本应用
- 熟悉 JAVA 开发
- 熟悉 Niagara 面向组件的开发
- 了解 Servlet 编程
- 了解 REST 接口开发
- 了解 JSON 的编程
- 学习微信公众平台开发接口
  https://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1421135319

- 微信 + Niagara 能做什么?
- 开发需要什么?
- **实战**

申请微信企业号

- 具体步骤网上搜索，这里截图是开发用的测试号
- 记住申请的appID和appsecret，用于获得动态密钥（Access Token），所有发给微信服务器的请求都需要校验动态密钥



申请微信企业号

- 指定Niagara服务器，开启80 web端口作为验证，/webchat/webchat为在Niagara上开发的免登陆Servlet
- Token需要和Niagara服务器上配置一致
- 注意：服务器需要先准备好，提交时就会进行验证

项目配置

- 设置项目依赖
- 增加Servlet配置

```
dependencies {
    compile "Tridium:nre:4.4"
    compile "Tridium:baja:4.4"
    compile "Tridium:alarm-rt:4.4"
    compile "Tridium:control-rt:4.4"
    compile "Tridium:web-rt:4.4"
    compile "libraries:javax.servlet-api-3.1.0:"
}

jar {
    from('src') {
        include 'WEB-INF/*.xml'
    }
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
    <display-name>Wechat interface</display-name>
    <servlet>
        <servlet-name>wechat</servlet-name>
        <servlet-class>com.tridiumap.wechat.WxServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>wechat</servlet-name>
        <url-pattern>/wechat</url-pattern>
    </servlet-mapping>
</web-app>
```

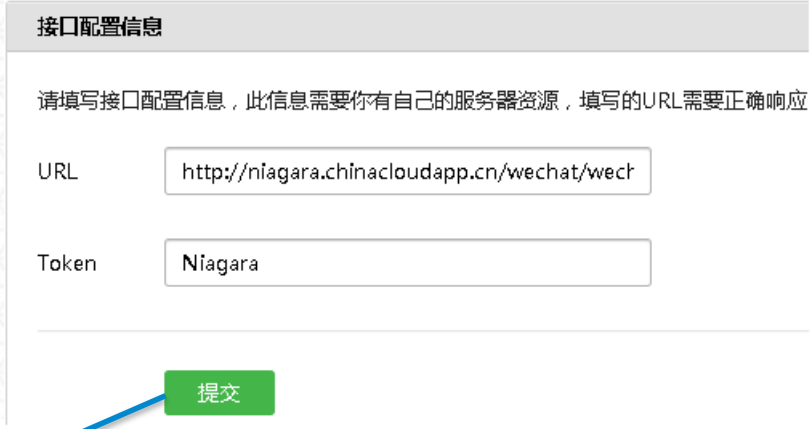
权限申明

- 申明需要访问微信服务器
- 申请需要免登陆Servlet

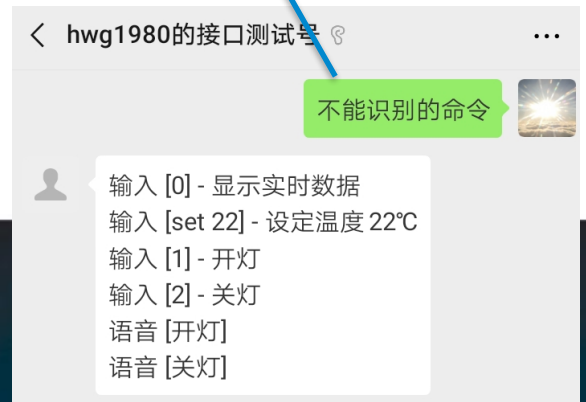
```
<permissions>
  <niagara-permission-groups type="all">
    <!-- Insert any global permissions here. -->
  </niagara-permission-groups>
  <niagara-permission-groups type="workbench">
    <!-- Insert any workbench specific permissions here. -->
  </niagara-permission-groups>
  <niagara-permission-groups type="station">
    <req-permission>
    <name>NETWORK_COMMUNICATION</name>
    <purposeKey>Need communicate with WeChat servers.</purposeKey>
    <parameters>
      <parameter name="hosts" value="api.weixin.qq.com"/>
      <parameter name="ports" value="443"/>
      <parameter name="type" value="client"/>
    </parameters>
    </req-permission>
    <req-permission>
      <name>UNAUTHENTICATED_ACCESS</name>
      <purposeKey>WeChat interface needs unauthenticated servlet</purposeKey>
      <parameters/>
    </req-permission>
  </niagara-permission-groups>
</permissions>
```

创建无需登录Servlet

- 继承UnauthenticatedServlet
- 重写doGet
  - 处理微信服务器对Niagara认证
- 重写doPost
  - 处理订阅者发给公众号的请求

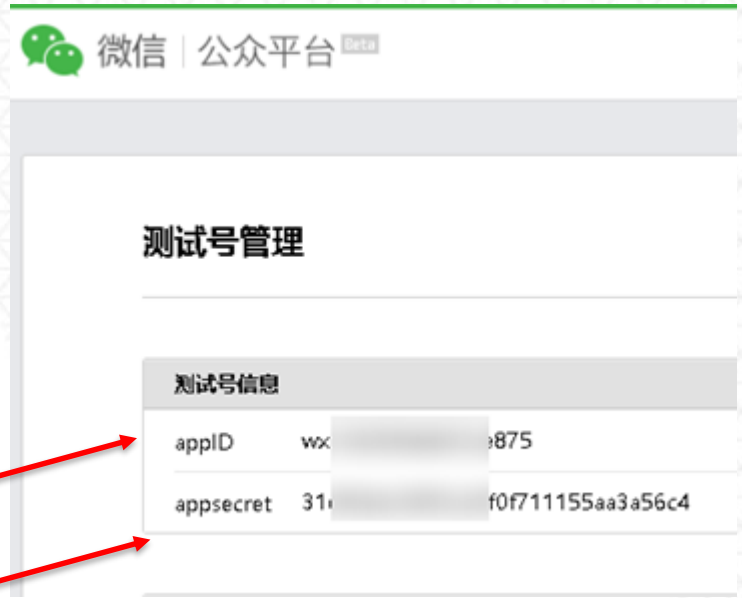



```
public class WxServlet extends UnauthenticatedServlet {
  @Override
  protected void doGet(HttpServletRequest req, HttpServletResponse res)
    throws ServletException, IOException {
    // 处理微信认证

}

@Override
protected void doPost(HttpServletRequest req, HttpServletResponse res)
  throws ServletException, IOException {
    // 处理订阅者发给公众号的请求
    // 此处可能根据您的业务需求做得相当复杂
}
```



获取Access Token

- 微信服务器的Access Token是Niagara向微信服务器申请的验证码，每两个小时必须重新申请一次。




```
public void doExecute()
{
  try{
    String sUrl = "https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid=" +
      getAppid() +
      "&secret=" + getSecret();
    String sRes = HttpClient.Send(HttpClient.ACTION_GET,sUrl,"", null, null);
    setResponse(sRes);

    parse(sRes);
  }catch(Exception e){}
}
```

niagara4

```
{
  “access_token”:“当前申请的ACCESS_TOKEN",
  "expires_in":7200
}
```

推送报警示例

- Niagara发给微信服务器
- 微信服务器根据信息发给用户
- 该代码在自定义的报警接收器里被调用

```
public void doAsyncRoute(BAlarmRecord balarmrecord){
  String sMessage;
  sMessage = getMessage().format(balarmrecord); // 使用format, 这样可以自由将报警记录中的信息组织成 JSON

  BWxSubscriber[] array = getToArray();
  setResponse("");
  String sRes = "";
  // 请求需要带上 access_token
  String sUrl = "https://api.weixin.qq.com/cgi-bin/message/template/send?access_token=" + getAccessToken();
  for(int i=0;i<array.length;i++){
    String sSend = sMessage.replace("{touser}", array[i].getOpenid());
    sRes += HttpClient.Send(HttpClient.ACTION_POST,sUrl,sSend, null, null);
  }
  setResponse(sRes);
}
```

更多高级功能

- 根据Session控制
- 微信人工智能接口
- 结合其他资源接口
  - 例如语音转发到智能音箱
  - 使用智能音箱回复微信请求
- 网站安全接口

开灯

开什么地方的灯?(三层走廊/Tridium办公室/Niagara会议室)

三层走廊

准备开启三层走廊的灯,是否确认?

确认

三层走廊的灯已经开启

更多高级功能

- 还有更多接口，发挥您的想象，结合Niagara的优势，还有更多应用可以开发 https://mp.weixin.qq.com/wiki

总结

开发要求

功能

微信→Niagara服务器：认证服务器、命令与控制

Niagara服务器→微信：服务请求、报警推送