MQTT协议通信

简述：

使用MQTT协议实现后台推送、及时通信等功能。本案例实现了web-js端、微信小程序端、Java client端、Java server端

软件版本：

Nginx: openresty-1.13.6.1-win32

<http://openresty.org/cn/>

Activemq: apache-activemq-5.15.2

<http://activemq.apache.org/download.html>

小程序端Mqtt.js:github上牛人在原eclipse开源的mqtt31.js上修改符合小程序使用的

<https://github.com/tennessine/paho.mqtt.wxapp>

https://unpkg.com/paho-wxapp@1.0.3/target/paho-mqtt.js

微信小程序：github上的微信客户端：

<https://github.com/sqwsummerwind/mqtt-wechat-applet-client>

JS端、Java client 端、Java Server 端、安卓端等：

<http://www.eclipse.org/paho/downloads.php>

案例运行环境：

Windows 7、jdk 8、小程序工具V1.02.1804120、Springboot-1.5.10

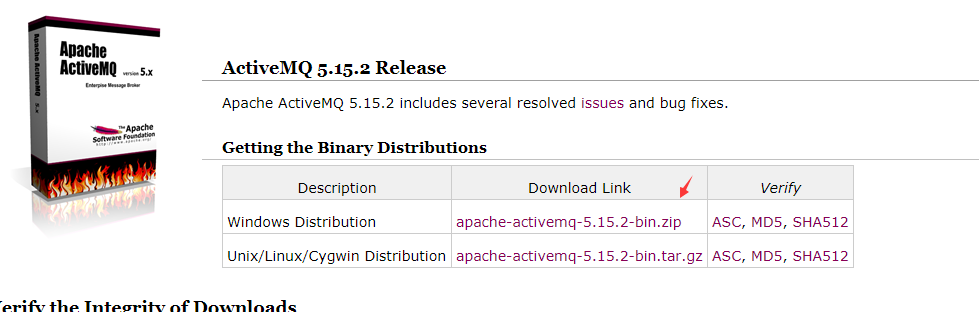
环境安装：

一、Jdk8 安装

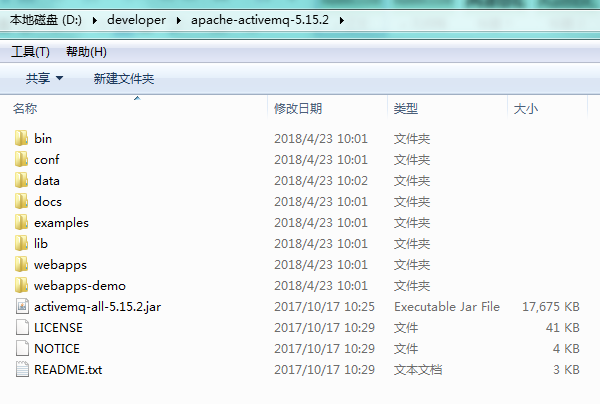
省略，自行安装

二、apache-activemq-5.15.2

1.下载windows版：<http://activemq.apache.org/activemq-5152-release.html>

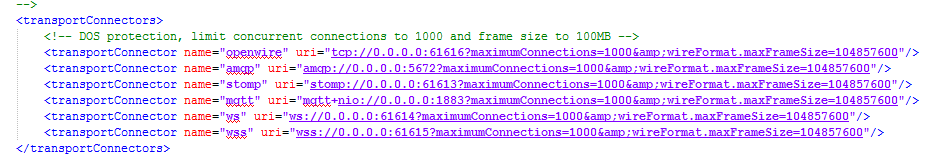


2.解压



3.配置修改(建议不做修改)

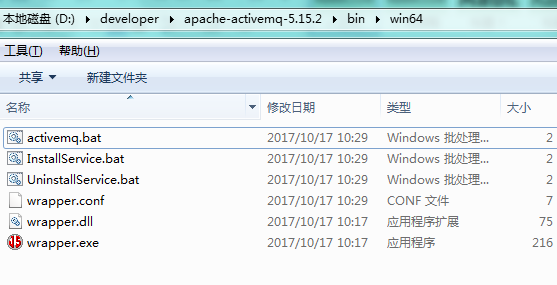
${active-home}\conf\activemq.xml支持的协议：



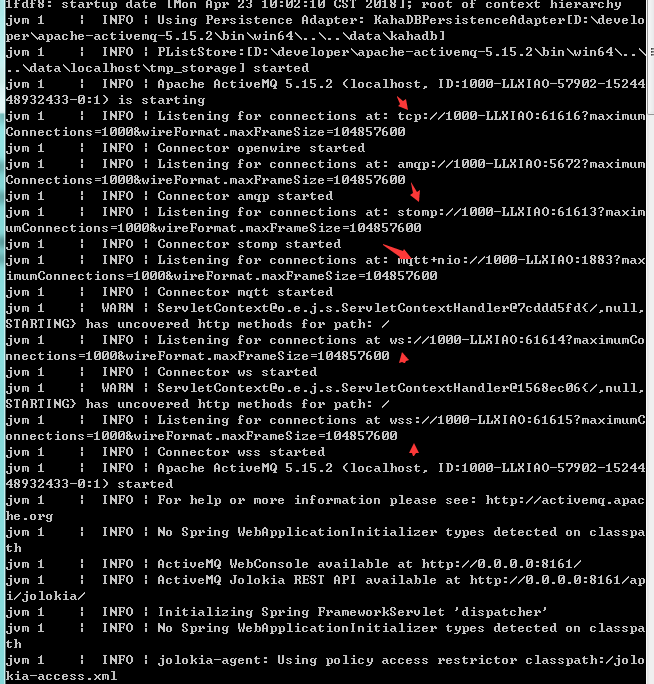
*tips:wss为后来验证所加，内部nginx跳转仅连接ws即可*

4.启动

${active-home}\bin\win64目录执行activemq.bat脚本即可



启动信息：



*tips:可以看到启动支持的一些协议连接url*

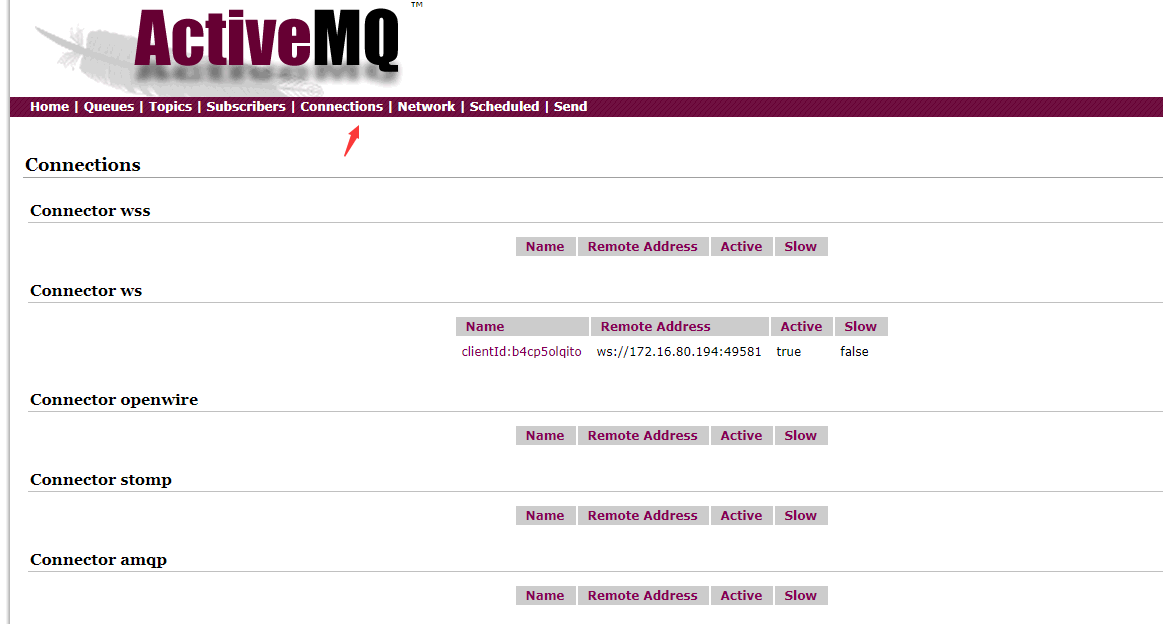
5.网页端请求，默认端口8161：

<http://localhost:8161/>



登录：admin/admin

查看连接信息：



至此，ACTIVEMQ完成

三、Nginx安装（主要适应微信小程序使用mqtt连接，js和java端可无需该支持）

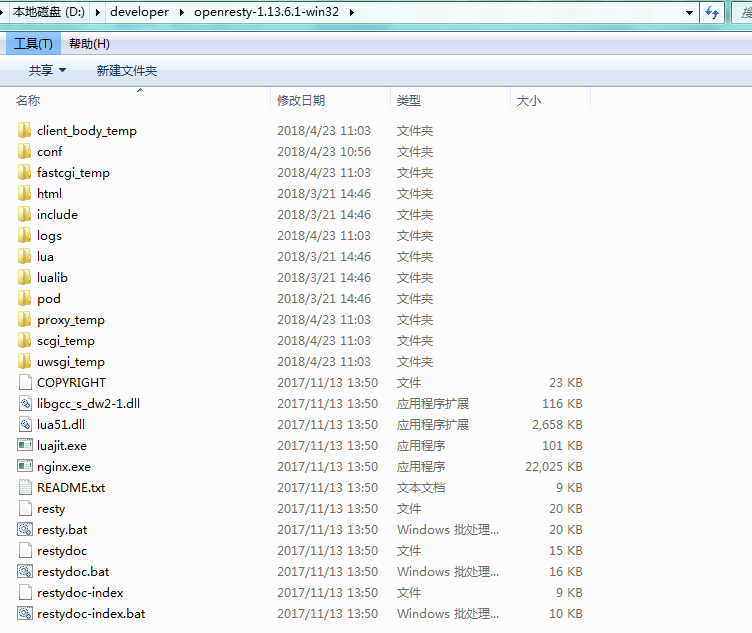
本案例使用openresty，集成了各种插件，特别是对于Lua的支持。普通的nginx不在本文考虑范围内

1.下载，选择win版本

<http://openresty.org/cn/download.html>



2.解压到目录：

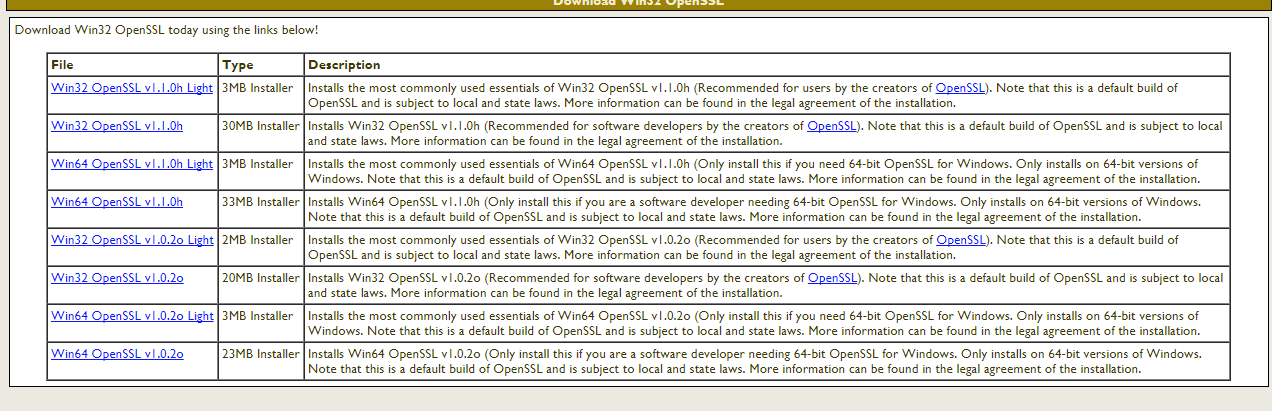


可以看到扩展了很多东东，集成了openssl(本案例重点)，还支持了lua脚本(接触重点)

3.证书生成(有证书忽略本节)

3.1 windows openssl 下载，选择对应的win版本

<http://slproweb.com/products/Win32OpenSSL.html>



3.2环境变量配置：

变量名：OPENSSL\_HOME

变量值：openssl安装地址 如：C:\OpenSSL-Win64\bin

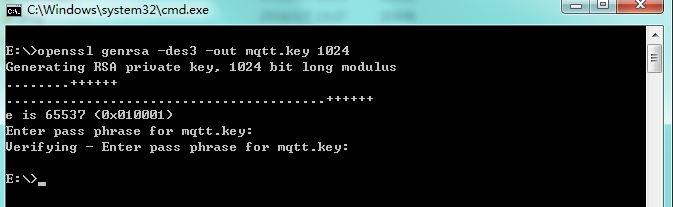
Path添加配置：%OPENSSL\_HOME%

3.3生成证书（证书名称自定义）

进入任何目录，如${openresty}\conf，进入CMD命令窗口

3.3.1 创建私钥

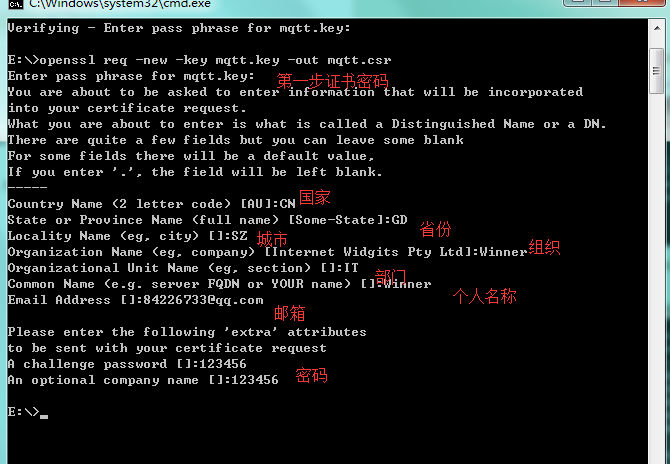
openssl genrsa -des3 -out mqtt.key 1024



输入两次密码：123456

3.3.2 创建csr证书

openssl req -new -key mqtt.key -out mqtt.csr



执行完以上步骤后，生成文件如下：



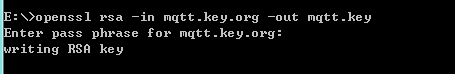
3.3.3 免密操作

*tips: 在加载SSL支持的Nginx并使用上述私钥时除去必须的口令，否则会在启动nginx的时候需要输入密码*

重命名mqtt.key文件为mqtt.key.org

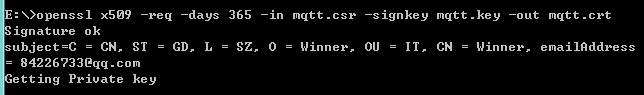
去除口令：

openssl rsa -in mqtt.key.org -out mqtt.key

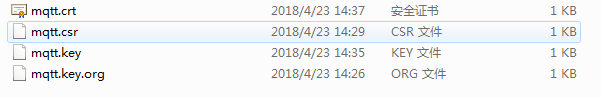


3.3.4 生成crt证书

openssl x509 -req -days 365 -in mqtt.csr -signkey mqtt.key -out mqtt.crt



最终生成文件如下：



4.nginx ssl配置

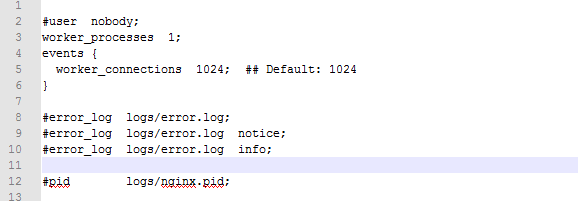
拷贝上一步生成的证书生成的crt和key文件到 ${openresty-home}\conf目录下

4.1 修改${openresty-home}\conf\nginx.conf文件，添加如下内容：

events {

worker\_connections 1024; ## Default: 1024

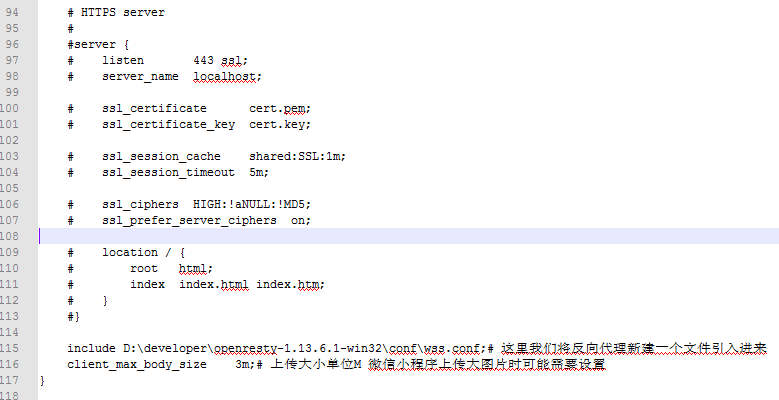
}



尾部添加

include ${openresty-home}\conf\wss.conf;# 这里我们将反向代理新建一个文件引入进来

client\_max\_body\_size 3m;# 上传大小单位M 微信小程序上传大图片时可能需要设置



4.2 在${openresty-home}\conf\wss.conf目录下新增wss.conf配置文件内容如下：

#主要是配置原来的ws 和 http 接口

upstream websocket {

# 远程websocket服务器地址，此处配置连接activemq ws连接

server 172.16.80.194:61614;

}

#upstream web{

# server www.xxx.com;# 远程http接口

#}

# 通过下面的反向代理到上面的接口去

server {

listen 443;#默认https和wss协议端口

ssl on;

ssl\_certificate D:\developer\openresty-1.13.6.1-win32\conf\mqtt.crt;#你的上传到服务器的证书位置

ssl\_certificate\_key D:\developer\openresty-1.13.6.1-win32\conf\mqtt.key;#你的上传到服务器的证书位置

ssl\_session\_timeout 5m;

ssl\_session\_cache shared:SSL:50m;

ssl\_protocols SSLv3 SSLv2 TLSv1 TLSv1.1 TLSv1.2;

ssl\_ciphers ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP;

underscores\_in\_headers on;#开启自定义头信息的下划线

#wss协议转发 小程序里面要访问的链接

location /mqtt {

proxy\_pass http://websocket;#代理到上面的地址去

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "Upgrade";

proxy\_redirect off;

#重要配置，解决小程序连接 Sec-WebSocket-Protocol错误提示

proxy\_set\_header Sec-WebSocket-Protocol mqtt;

more\_clear\_headers Sec-WebSocket-Protocol;

}

#https协议转发 小程序里面要访问的链接

#location /{

# proxy\_pass http://web;#代理到原有的http的地址去

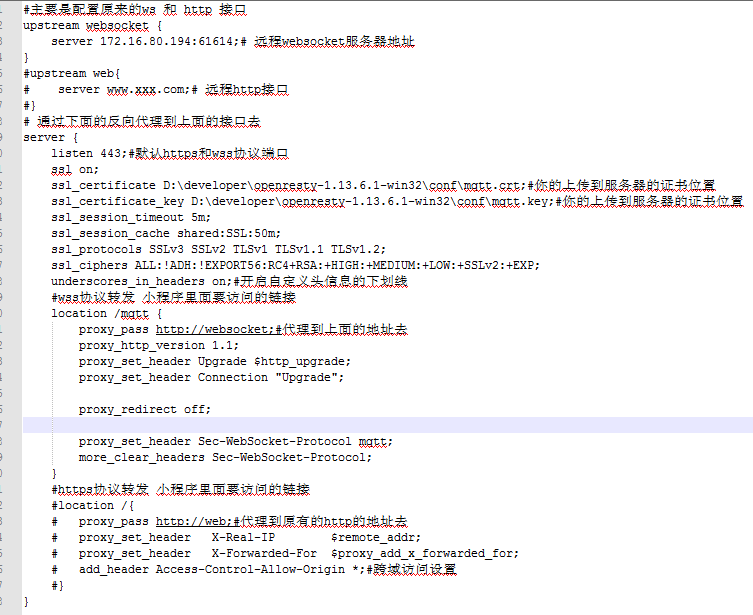
# proxy\_set\_header X-Real-IP $remote\_addr;

# proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

# add\_header Access-Control-Allow-Origin \*;#跨域访问设置

#}

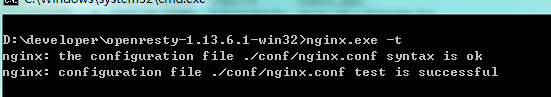
}



5.测试&启动

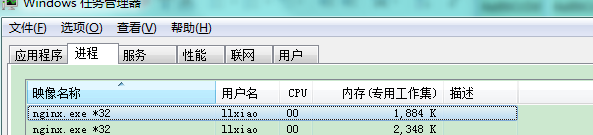
Cmd命令，进入openresty根目录

执行 nginx.exe -t 测试配置文件是否正确



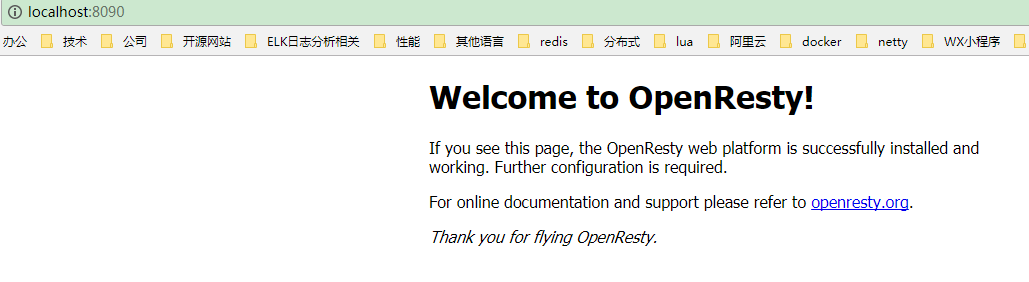
*tips:如有错误请做相应的修改*

测试通过后直接双击nginx.exe启动nginx



访问：

<http://localhost:8090/>



至此nginx ssl成功。

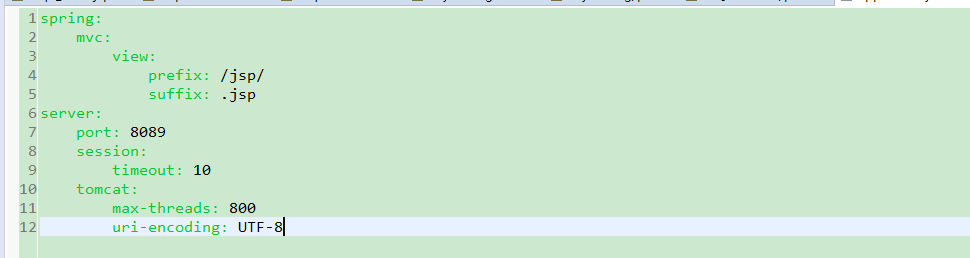
四、web服务js端，java服务端，java Client端操作：

Maven+Springboot+jsp实现

1.pom配置：



2. application.yml配置(使用web工程，使用JSP作为视图端)



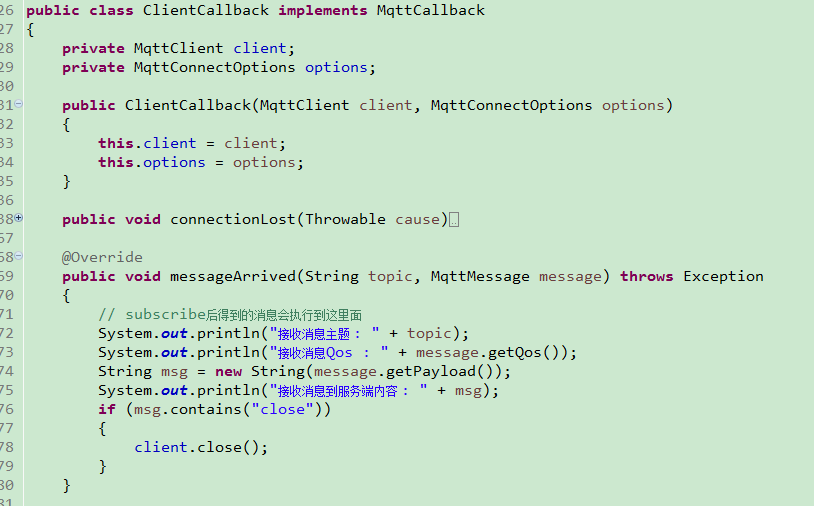
3.JAVA服务端普通JMS发送



4.JAVA mqtt client连接

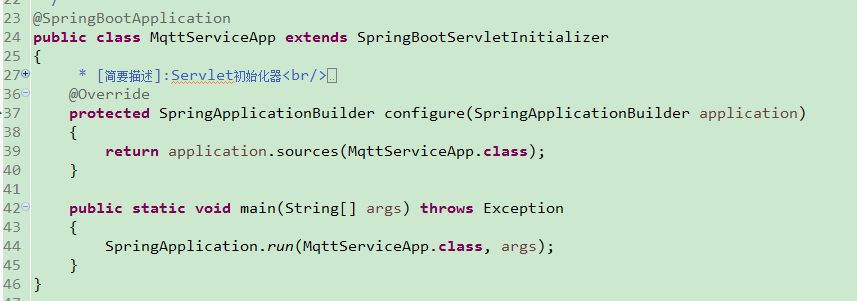


Callback

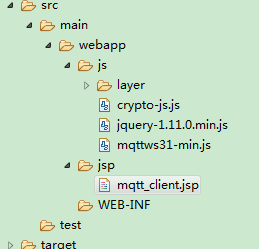


5.springboot web jsp服务器

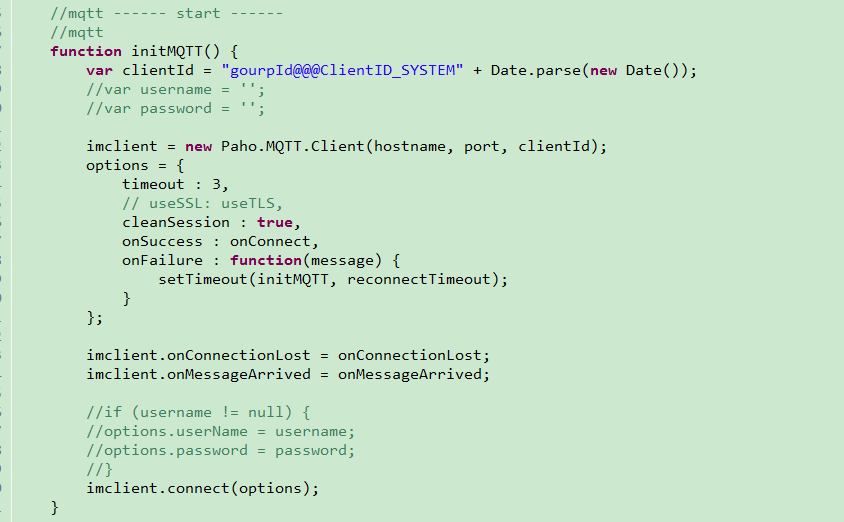
5.1 主入口



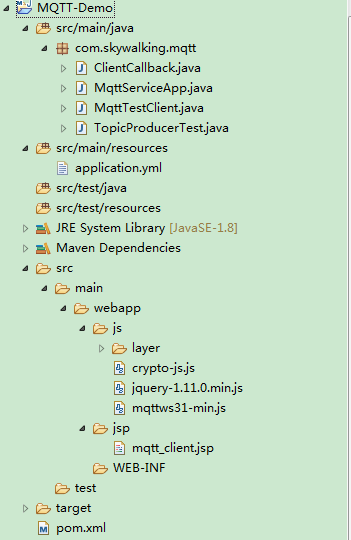
5.2 JSP相关配置



5.3 Jsp页面JS连接mqtt



以上为部分伪代码，案列DEMO工程目录结构：



源码：



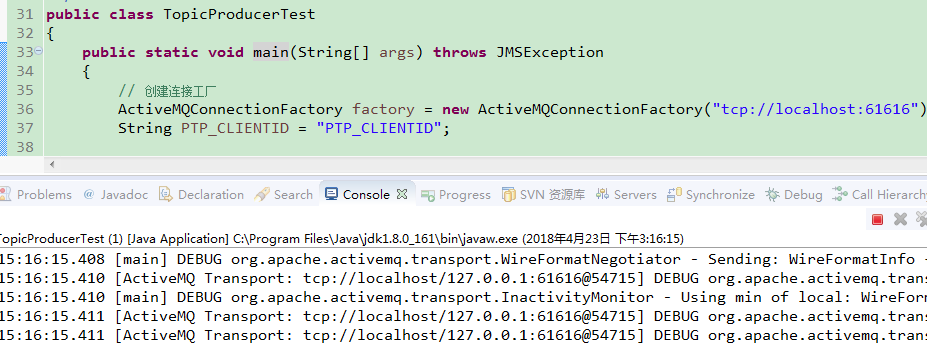
五、微信小程序端mqtt

参考工程：



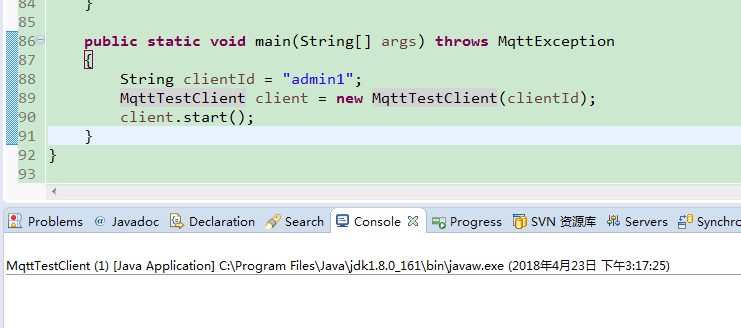
六、效果演示：

6.1 启动服务端TopicProducerTest：



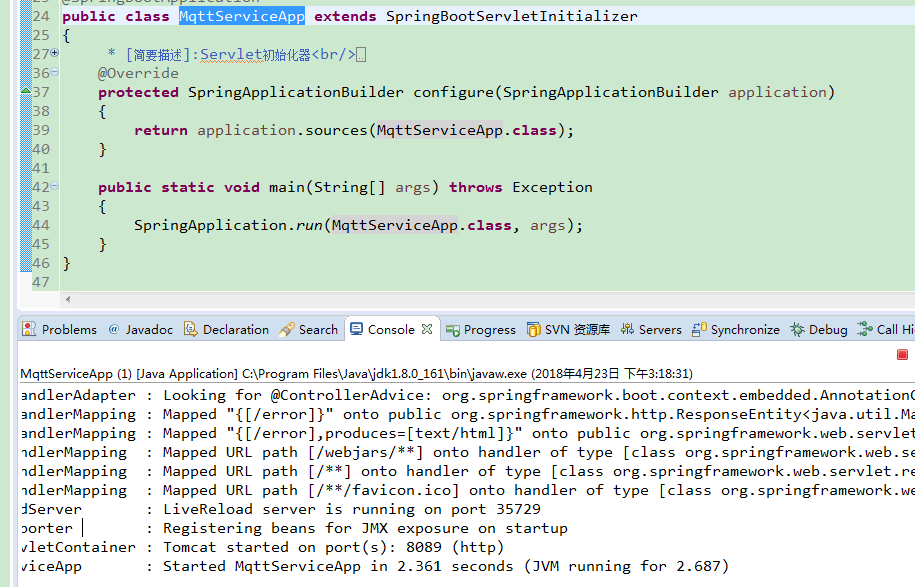
连接的是tcp 61616端口 activemq openwire协议，往主题PTP.test 发送消息

6.2 启动java client端 MqttTestClient

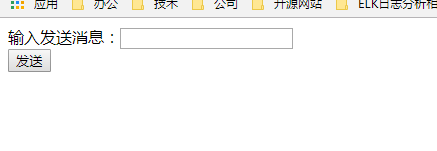


连接的是：tcp://localhost:1883 activemq matt协议，订阅PTP/test主题

6.3 启动web服务 MqttServiceApp



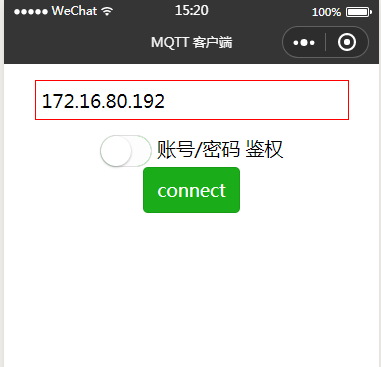
访问jsp: <http://localhost:8089/jsp/mqtt_client.jsp>



JS端连接的是 ws协议端口即：

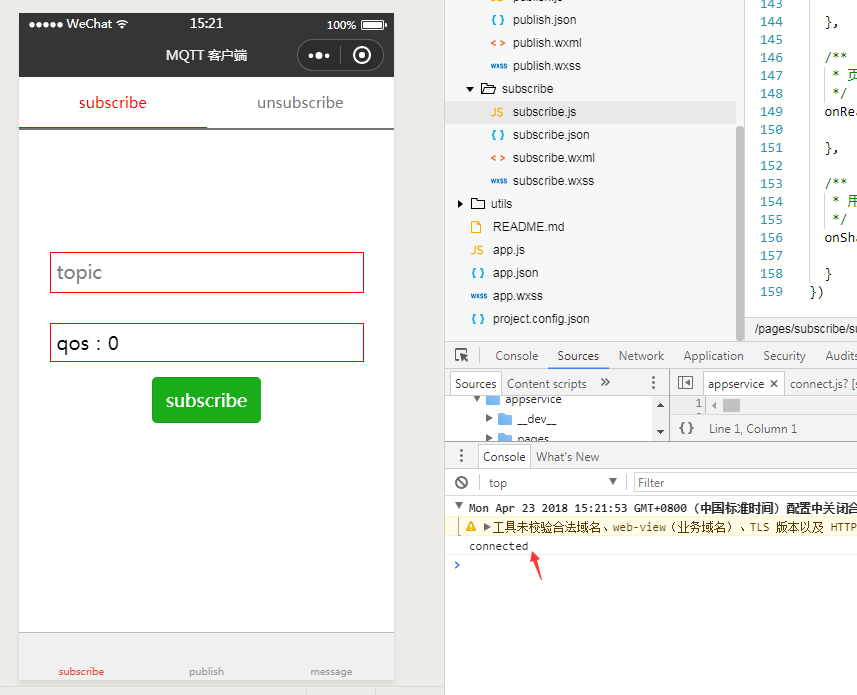
ws://localhost:61614/mqtt。使用mqtt31.js仅需指定端口Ip 即可

6.4 连接微信小程序端：

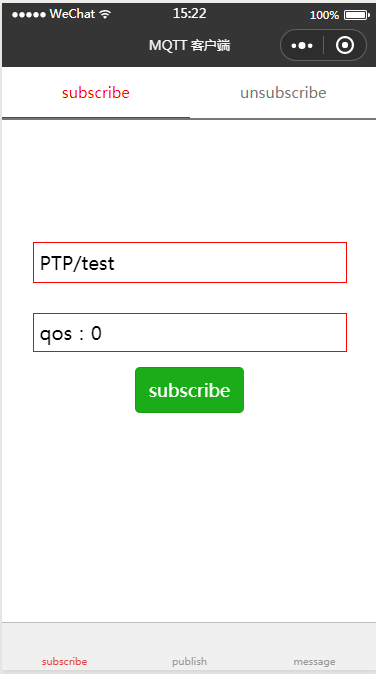


tips:此处连接的是nginx 443端口，微信默认使用wss协议连接，通过nginx跳转ws到 activemq上，微信小程序目前还暂不支持mqtt协议，开发的仅是websocket协议

连接成功后：

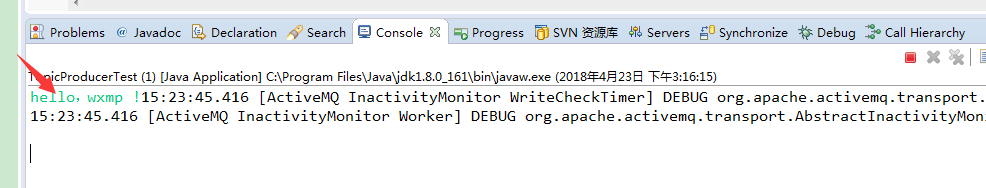


订阅主题：PTP/test

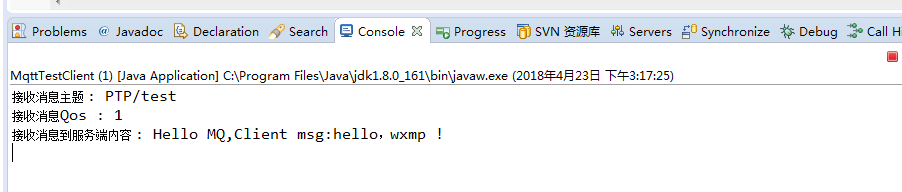




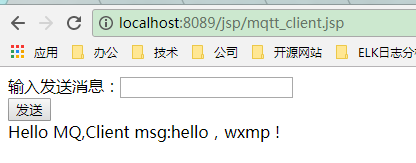
6.5 服务端发送消息



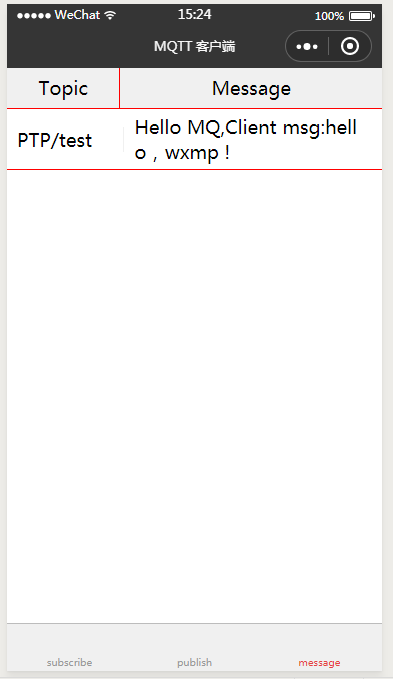
Java client 接受消息：



网页端：



小程序端：



Tips案例的message需要手动点击才会看到消息，暂为做自动显示

6.6 activemq连接情况



7、参考连接：

<http://www.wxappclub.com/topic/842>

<https://github.com/sqwsummerwind/mqtt-wechat-applet-client>

<https://blog.csdn.net/oufua/article/details/54375609>

<https://zhuanlan.zhihu.com/p/24823848>