



基于 Pulsar 获取消息推送

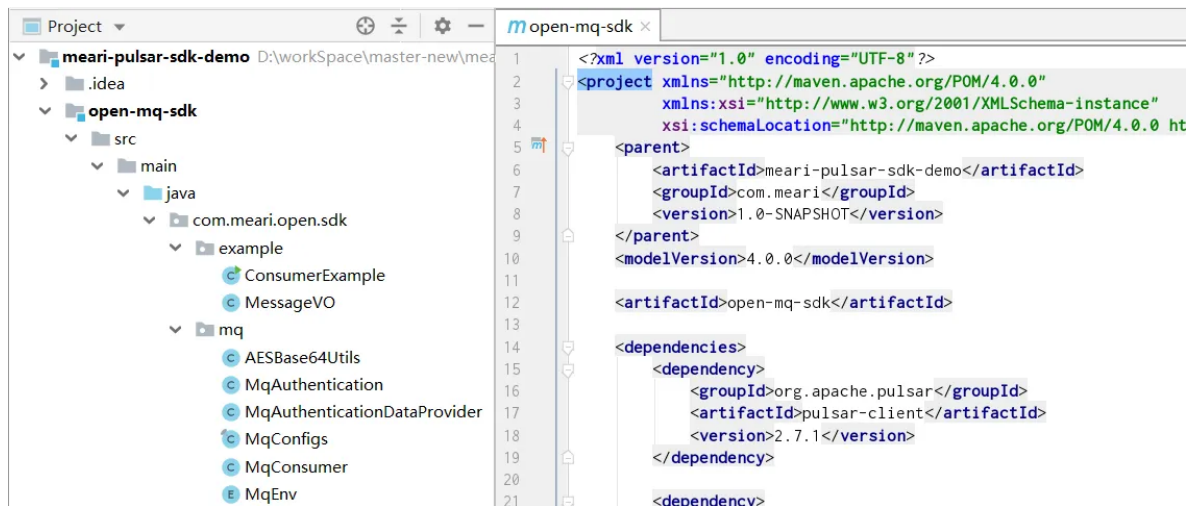
觅睿基于开源的 Pulsar 进行了定制改进，当触发事件报警或其它消息时，觅睿服务器会通过Pulsar进行转发，方便您及时收到消息，并可以分析和处理这些消息。本文将介绍如何使用觅睿 Java 版本 Pulsar SDK 进行消息接收实践。

准备工作

- 从GitHub获取源码工程：[Meari Pulsar SDK](#)
- 准备Java运行环境。
- 准备一个能连接觅睿云的智能设备，并绑定到APP。

第一步：配置开发环境

1. 打开下载好的Meari Pulsar SDK项目。



2. 修改com.meari.open.sdk.example.ConsumerExample文件参数

```
private static String URL = MqConfigs.CN_SERVER_URL;  
private static String ACCESS_ID = "";  
private static String ACCESS_KEY = "";
```



参数说明如下：

参数	说明
URL	接入地址，根据测试的区域进行选择，由觅睿提供。
ACCESS_ID	觅睿提供的partnerKey参数。
ACCESS_KEY	觅睿提供的partnerSecret参数。

第二步：测试接收消息

1. 运行com.meiri.open.sdk.example.ConsumerExample#main方法

```
1 package com.meiri.open.sdk.example;
2
3 import com.alibaba.fastjson.JSON;
4 import com.meiri.open.sdk.mq.AESBase64Utils;
5 import com.meiri.open.sdk.mq.MqConfigs;
6 import com.meiri.open.sdk.mq.MqConsumer;
7 import org.apache.pulsar.client.api.MessageId;
8
9 public class ConsumerExample {
10
11     private static String URL = MqConfigs.CN_SERVER_URL;
12     private static String ACCESS_ID = "";
13     private static String ACCESS_KEY = "";
14
15     public static void main(String[] args) throws Exception {
16         MqConsumer mqConsumer = MqConsumer.build().serviceUrl(URL).accessId(ACCESS_ID).accessKey(ACCESS_KEY)
17             .messageListener(message -> {
18                 MessageId msgId = message.getMessageId();
19                 String tid = message.getProperty("tid");
20                 long publishTime = message.getPublishTime();
21                 String payload = new String(message.getData());
22                 System.out.println("<msgId>" + msgId + "<tid>" + tid + "<publishTime>" + publishTime + "<payload>" + payload);
23                 payloadHandler(payload);
24             });
25         mqConsumer.start();
26     }
27
28     private static void payloadHandler(String payload) {
29         try {
30             MessageVO messageVO = JSON.parseObject(payload, MessageVO.class);
31             //decryption data
32             String dataJsonStr = AESBase64Utils.decrypt(messageVO.getData(), ACCESS_KEY.substring(6, 22));
33             System.out.println("messageVO=" + messageVO.toString() + "\n" + "data after decryption dataJsonStr=" + dataJsonStr);
34         } catch (Exception e) {
35             e.printStackTrace();
36         }
37     }
38 }
```

2. 操作设备，在APP上打开设备报警开关，并触发事件报警。

3. 在控制台上自动接收到消息。

```
Run: ConsumerExample x
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
###PULSAR_MSG => pulsar consumer initial success
###PULSAR_MSG => start receive next message
<msgId>13848:79:-1:0<tid>null<publishTime>1717070017891
<msgId>13848:79:-1:0<tid>null<publishTime>1717070017891<payload>{"msgType":"alarm","t":1717070017871,"version":"1.0"}
MessageVO{data='null', msgType='alarm', version='1.0', t=1717070017871}
messageVO=MessageVO{data='null', msgType='alarm', version='1.0', t=1717070017871}
data after decryption dataJsonStr=null
business processing cost={105
<msgId>13848:79:-1:0<tid>null<publishTime>1717070017891
###PULSAR_MSG => start receive next message
<msgId>13848:80:-1:0<tid>null<publishTime>1717070017942
<msgId>13848:80:-1:0<tid>null<publishTime>1717070017942<payload>{"msgType":"alarm","t":1717070017942,"version":"1.0"}
MessageVO{data='null', msgType='alarm', version='1.0', t=1717070017942}
```

第三步：消息内容分析

1. 接收到的数据格式

```
{
  "data": "",
  "msgType": "alarm",
  "version": "1.0",
  "t": 1717059971006
}
```

参数说明如下：

参数	说明
data	加密的消息内容
msgType	消息类型。'alarm'表示报警消息，后续根据需求会拓展其它类型。
version	版本默认1.0
t	觅睿服务器的时间戳

解密方法请查看

com.meari.open.sdk.example.ConsumerExample#payloadHandler方法