FSTP开发手册（草稿）

# 目录

[目录 1](#_Toc511743179)

[修订（开发）记录 2](#_Toc511743180)

[一．安装 3](#_Toc511743181)

[环境准备： 3](#_Toc511743182)

[二．启动服务步骤： 3](#_Toc511743183)

[三．开发一个FSTP服务 4](#_Toc511743184)

[（一）PubExampleService 4](#_Toc511743185)

[1）创建服务 4](#_Toc511743186)

[2）配置config文件 5](#_Toc511743187)

[3）创建BO 5](#_Toc511743188)

[4）完成publish 7](#_Toc511743189)

[5）运行程序 8](#_Toc511743190)

[6） 结果 8](#_Toc511743191)

[（二）SubExampleService 9](#_Toc511743192)

[1）创建服务并配置配置文件 9](#_Toc511743193)

[2）完成服务 9](#_Toc511743194)

# 修订（开发）记录

| 日期 | 版本 | 修订说明 | 修订者 |
| --- | --- | --- | --- |
| 2018-4-17 | 0.0.1 | 创建文档，加入pub和sub的示例程序。  目前程序可供使用的模块：   1. RPC：pub/sub，QNS，Query/Respond(目前不可用) 2. Cache：重新封装，弃用CacheClient，代替为FCache 3. DB：采用Spring-Data-jpa动态加载 4. BO：采用proto封装，可操作类后缀为\_OTW 5. MonitorService目前功能： 6. 侦测其他服务，以及心跳 7. 管理各个服务 8. 解析QNS请求   下一步计划做的工作：   1. 完成python、C/C++的接口开发工作，即开发python客户端时，任何service继承自PService，完成两者之间的兼容 2. Query/Respond模式的修复（因为BO序列化的原因，该功能需要重新设计） 3. 写python端的数据接口，对接tushare、Vnpy，以及量化平台 | 姜瀚 |

# 一．安装

# 环境准备：

1. JRE:1.8
2. Redis：3.0.5
3. Qpid：7.0.0
4. MySQL:5.7

# 二．启动服务步骤：

1）服务端安装以上环境

Redis配置文件：config/redis.properties

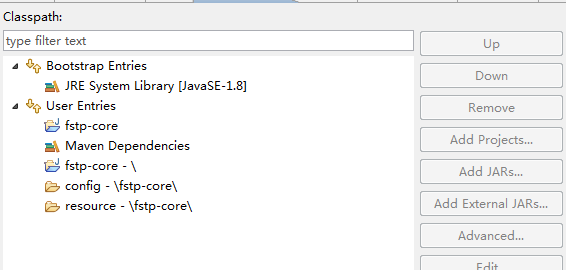
Qpid配置文件： src/main/java/com/purefun/fstp/core/qpid/qpid.properties

MySQL采用Spring-Data-JPA动态加载，标准见config/StockRDS/DEV/1/DBConfig4.xml

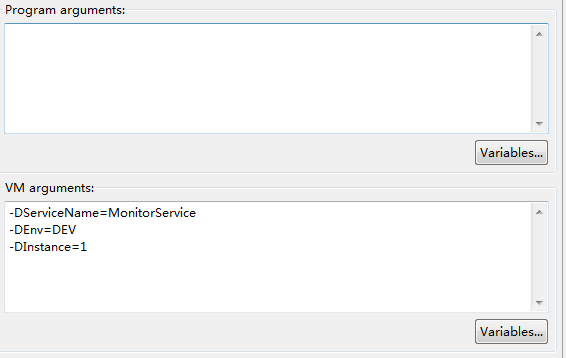
2）配置Qpid

fstp-core\src\main\java\com\purefun\fstp\core\qpid\qpid.properties

3）配置classpath：



1. 配置JVM参数，至少先要启动一个MonitorService



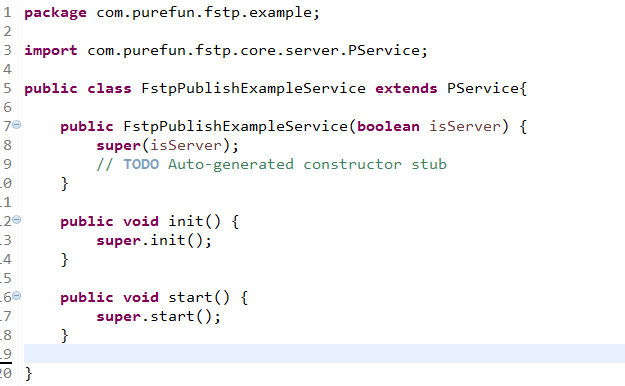
# 三．开发一个FSTP服务

本教程将指导开发一个典型的pub-sub类型的服务

## （一）PubExampleService

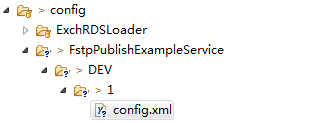
### 1）创建服务

创建一个类FstpPublishExampleService，并继承PService。



### 2）配置config文件

在config中建立一个config文件，按如下目录组织



服务启动时，会根据JVM参数来加载对应的服务，配置文件如下



注意其中的beanid=MainServer处，要改成自己的服务类

### 3）创建BO

在FSTP目前有三类对应的通信方式:

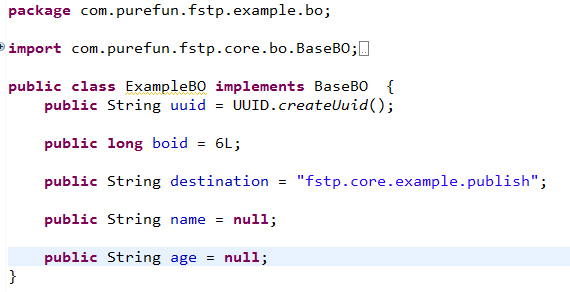
Pub/Sub方式 : 广播、订阅方式

Query/Reply方式 ：请求应答模式

QNS：首先做一次Query，再订阅某个topic

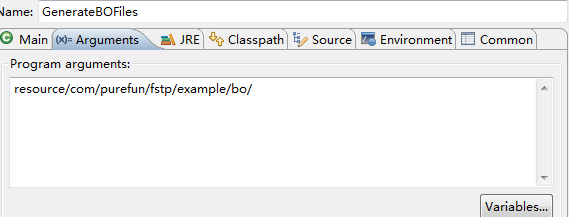
但每种通信模式都使用BO（Bussniss Object）作为传输主体，为了传输，FSTP中目前采用Protobuf进行序列化。过程如下：

首先创建bo类

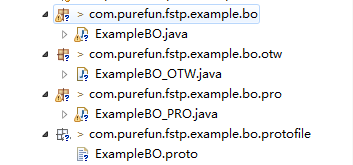


其中uuid，boid，destination是每一个bo都需要定义的。

打开resource下的GenerateBOFile.java，把BO的路径作为参数进行执行

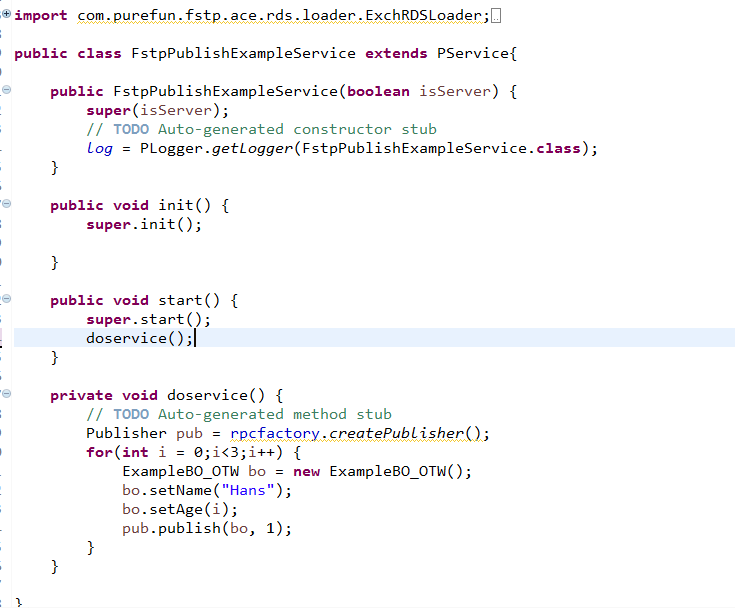


执行结束后生成如下类

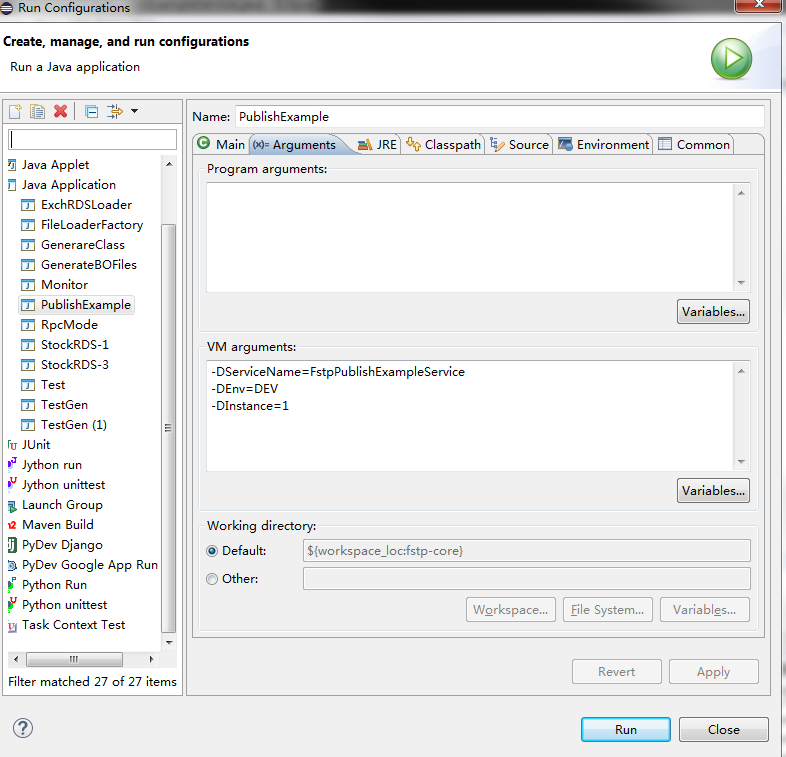


.proto是proto的文件，OTW的是bo操作类，PRO是序列化类

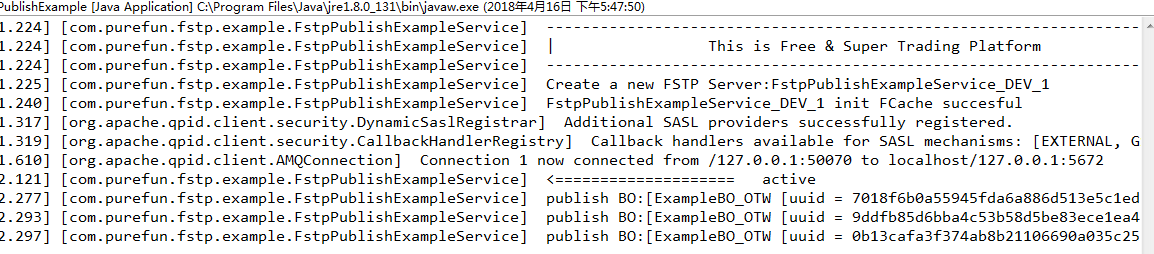
### 4）完成publish



### 5）运行程序



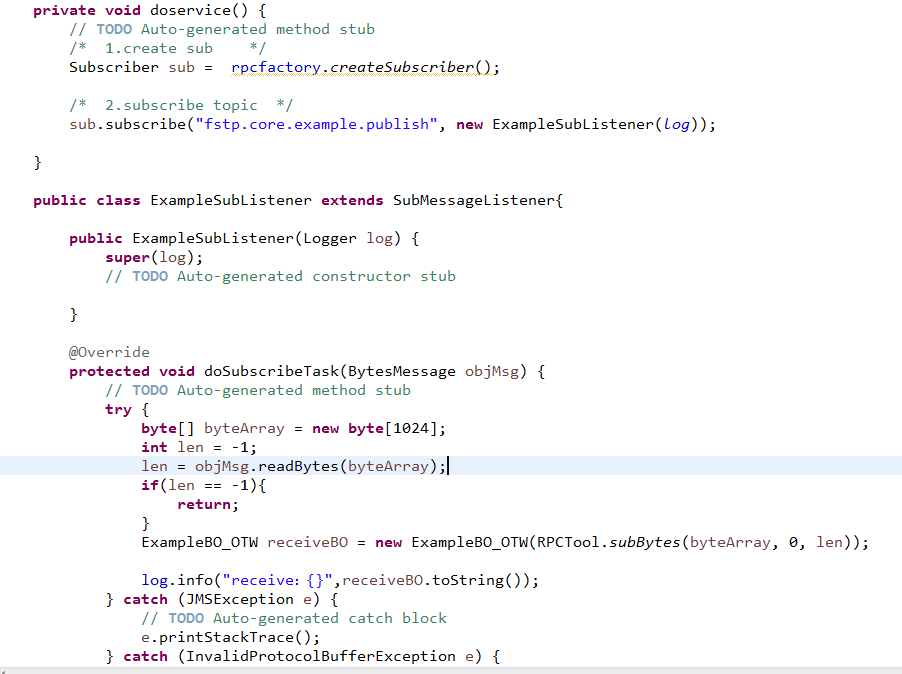
### 结果



## （二）SubExampleService

### 1）创建服务并配置配置文件

### 2）完成服务



3）结果

先启动PublishExample，再启动subscribeExample

