图文 93 如何在Intellij IDEA中启动Broker以及本地调试源码?

详情 评论

如何在Intellij IDEA中启动Broker以及本地调试源码?

继《从零开始带你成为JVM实战高手》后，救火队长携新作再度出山，重磅推荐：

（点击下方蓝字试听）

《从零开始带你成为MySQL实战优化高手》

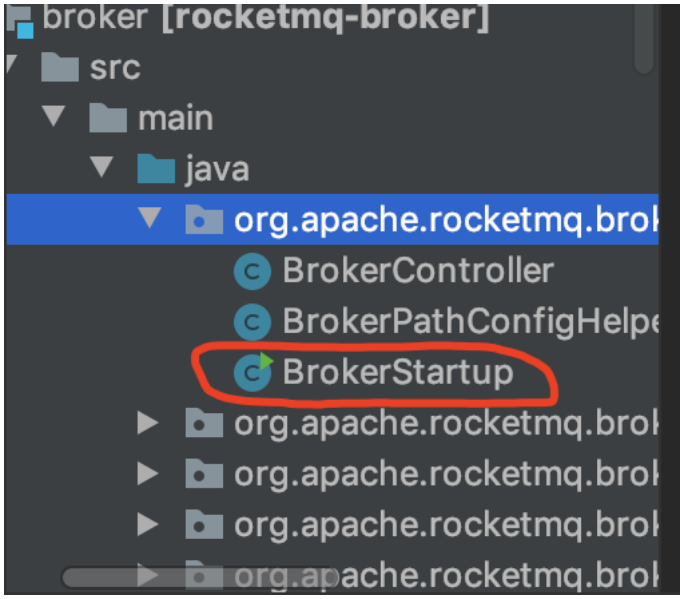
1、对Intellij IDEA中的broker模块进行配置

上一讲我们已经成功的堆Intellij IDEA中的NameServer配置了环境变量，而且设置好了运行目录，配置文件等等，而且已经成功的在Intellij IDEA中启动了NameServer。

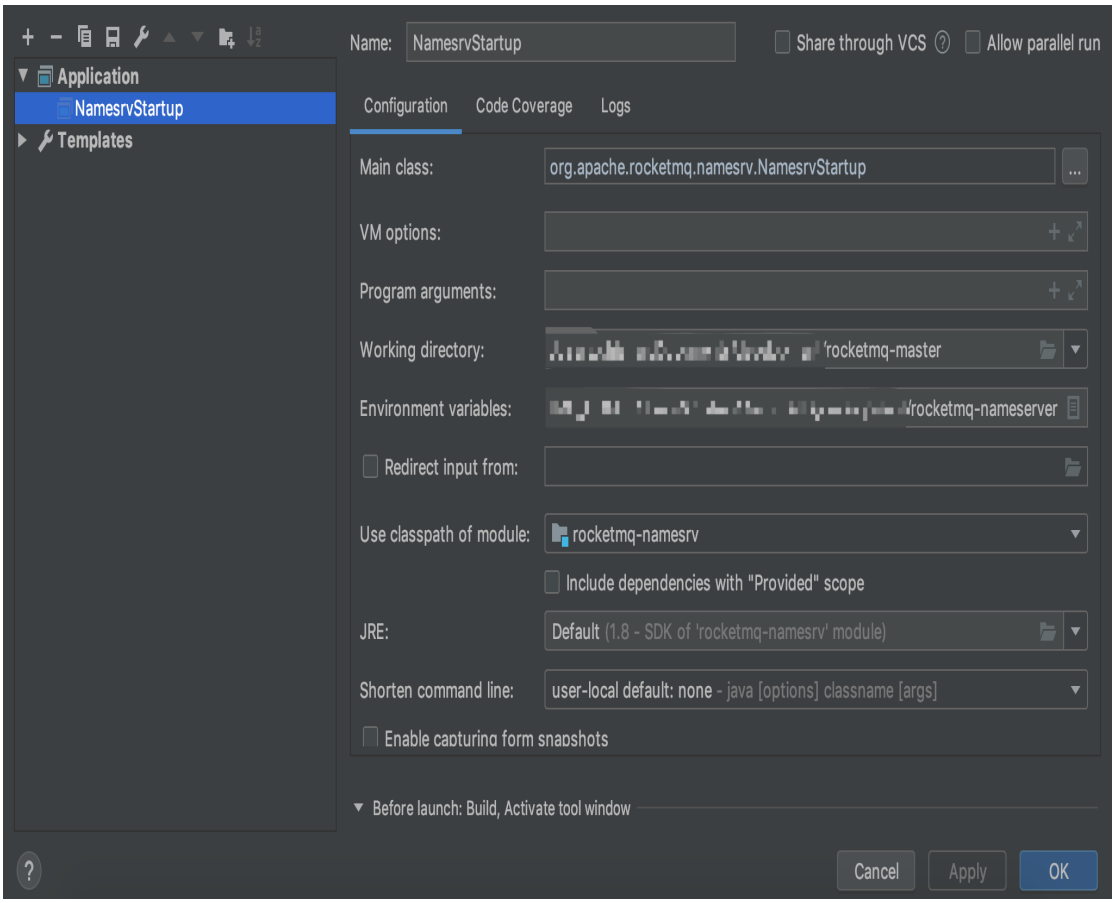
接下来有了NameServer，我们就可以在Intellij IDEA中启动一个Broke了。

首先我们依然是在Intellij IDEA中找到broker模块，然后展开他的目录，就可以找到一个BrokerStartup类，这个类是用来启动Broker进程的

我们看下面的图

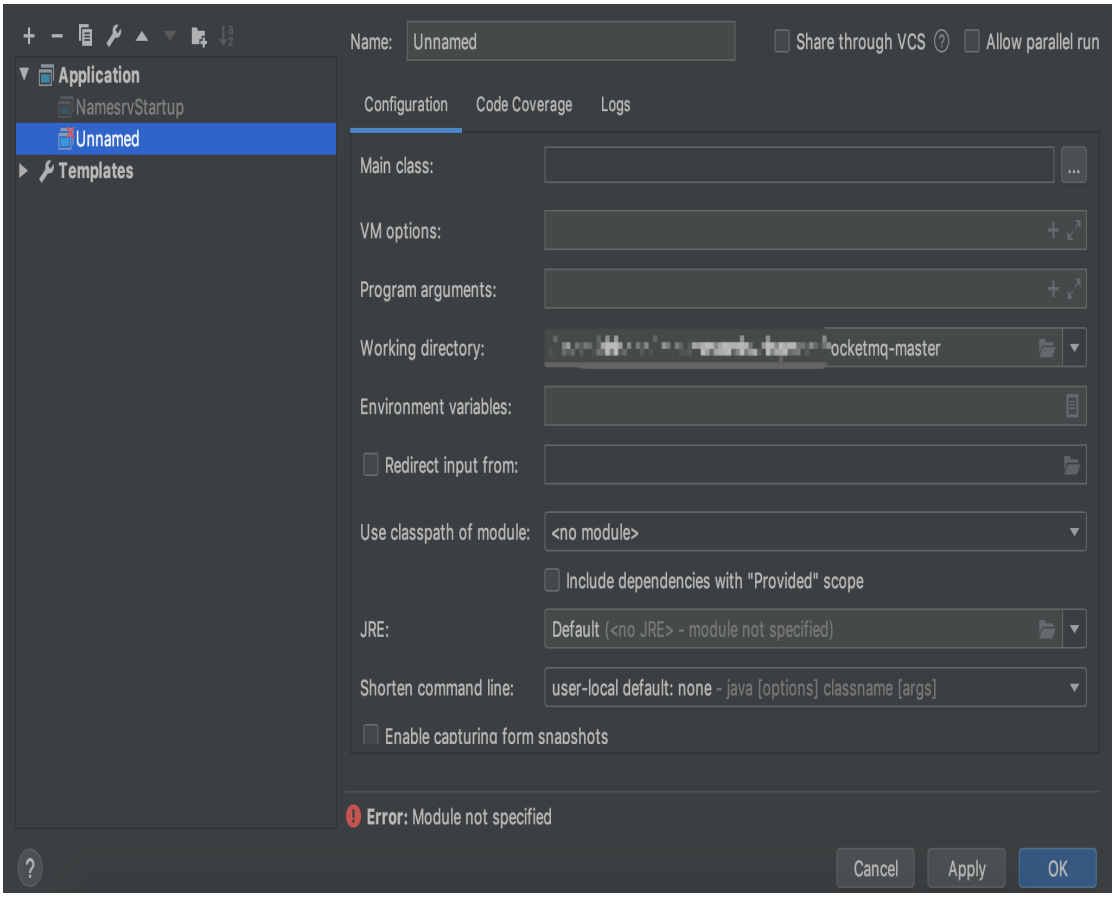


然后我们选中这个BrokerStartup类，接着依然在Intellij IDEA的上面选择Edit Configuration，进入这个启动类的配置编辑界面，但是刚开始他会直接显示出来NamesvrStartup的配置编辑界面，如下图所示。

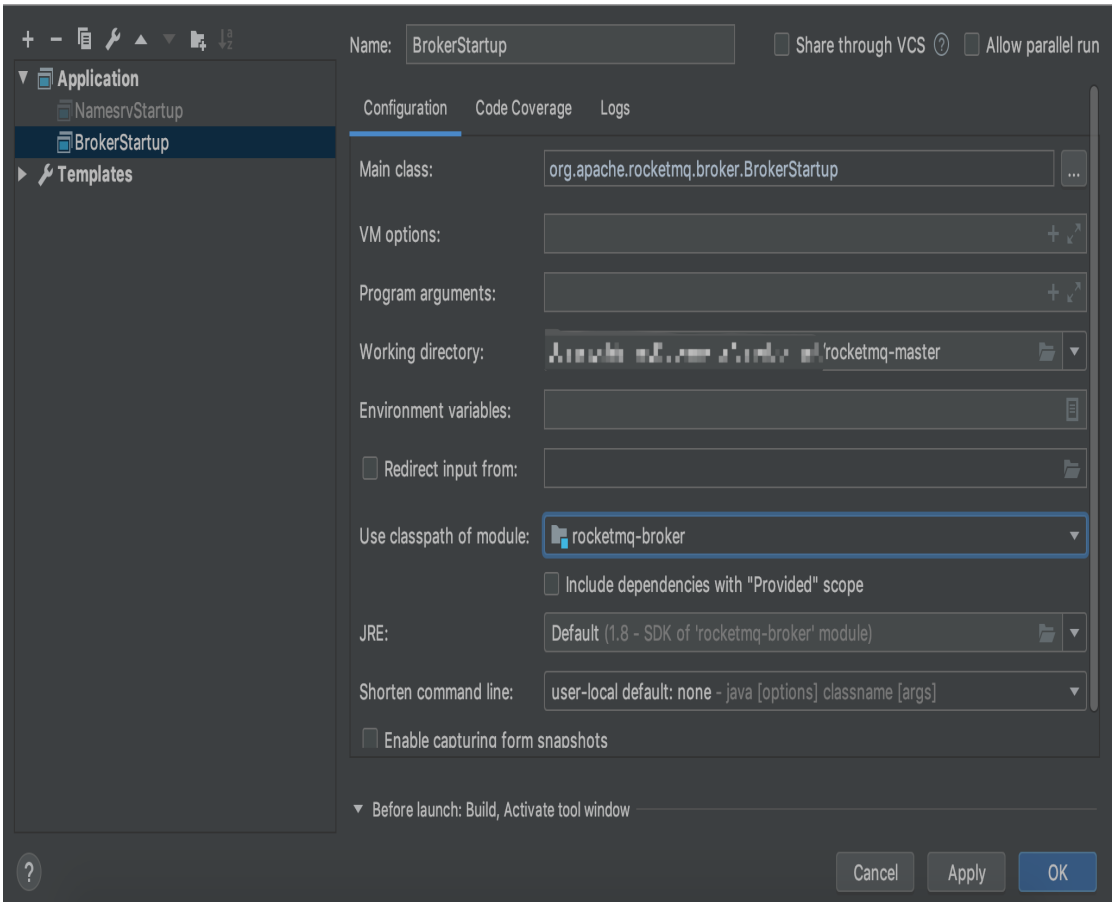


我们发现他的Main class指定的是：org.apache.rocketmq.namesrv.NamesrvStartup，这不是我们要的类，所以这个时候，我们得重新为BrokerStartup类新建一个配置模板

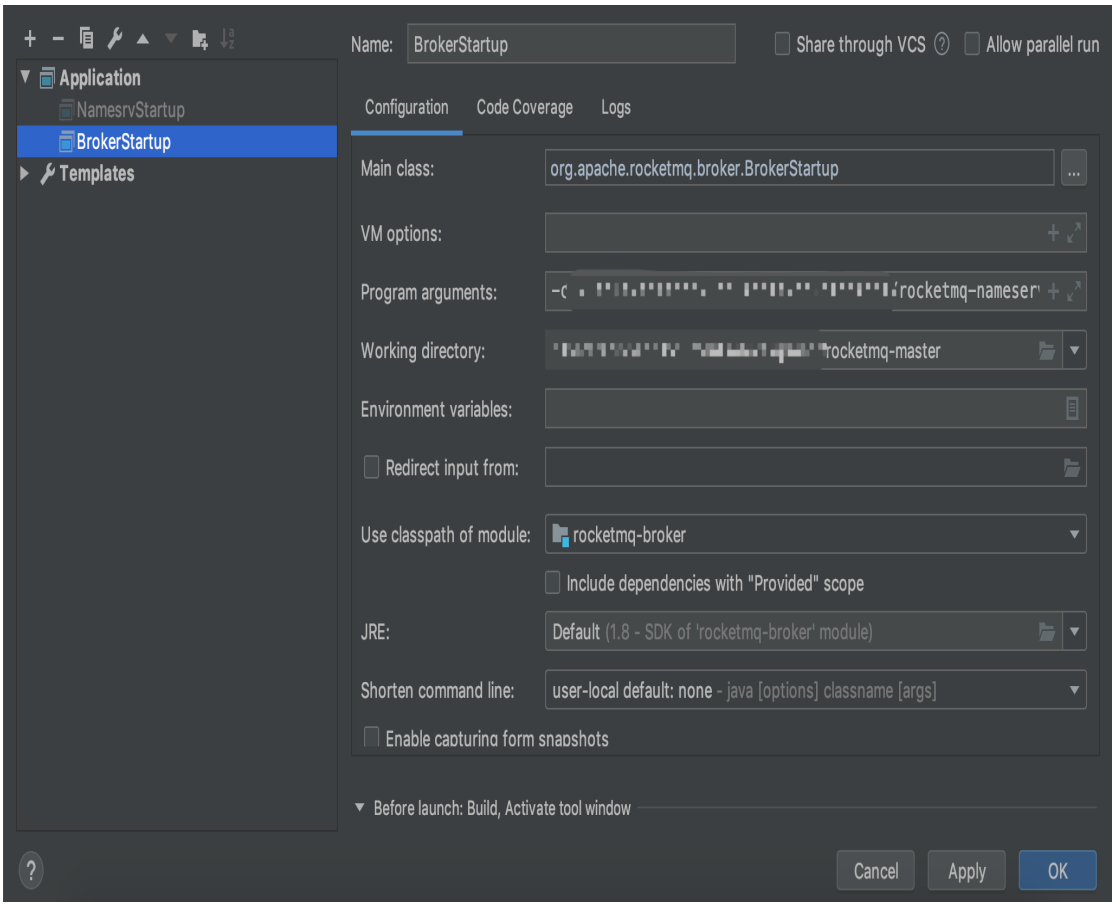
在上面那个界面的左上角有一个+号，我们可以点一下，然后选择Application，此时会出现一个新的配置模板，如下图。



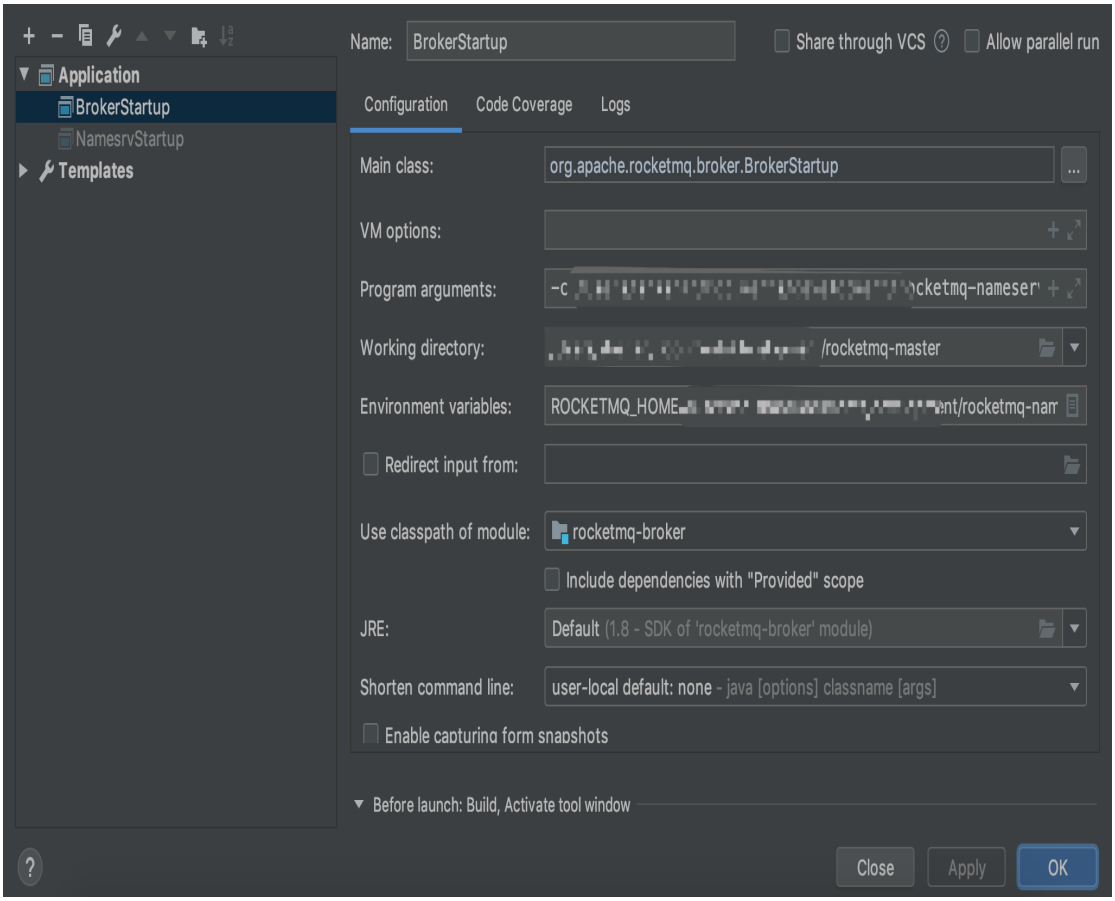
这个配置模板此时是没有名字的，我们在Name中输入BrokerStartup，Main class可以选择broker模块下的BrokerStartup类，Use classpath of module中可以选择broker这个module，然后会看到如下图所示。



接着我们就可以对他进行配置了，首先在Program arguments里，我们需要输入下面的内容，给Broker启动的时候指定一个配置文件存放地址：-c 你的rocketmq运行目录/conf/broker.conf，如下图所示。



接着我们需要配置环境变量，也就是ROCKETMQ_HOME，此时我们可以在Environment Variables里面添加一个ROCKETMQ_HOME环境变量，他的值就是我们的rocketmq运行目录就可以了，就是里面有conf、store、logs几个目录的，如下图所示。



这些都配置好了之后，那么我们的BrokerStartup启动类就配置好了，因为这个时候Broker启动会收到一个-c以及配置文件的参数，而且他知道环境变量ROCKETMQ_HOME，知道运行目录是哪个，接着他就会基于这个配置文件来启动，同时在这个运行目录中存储数据，包括写入日志。

2、再看broker的配置文件的内容

接着我们再来看一次broker配置文件的 内容，就是broker.conf里的内容，是我们上一篇文章编辑的，如下所示。

大家会发现，这里主要是配置了NameServer的地址，然后配置了Broker的数据存储路径，包括commitlog文件、consume queue文件、index文件、checkpoint文件的存储路径，这些文件之前我们都给大家讲解过是干什么用的了，大家应该还记得。

```
brokerClusterName = DefaultCluster
brokerName = broker-a
brokerId = 0
# 这是nameserver的地址
namesrvAddr=127.0.0.1:9876
deleteWhen = 04
fileReservedTime = 48
brokerRole = ASYNC_MASTER
flushDiskType = ASYNC_FLUSH
# 这是存储路径，你设置为你的rocketmq运行目录的store子目录
storePathRootDir=你的rocketmq运行目录/store
# 这是commitLog的存储路径
storePathCommitLog=你的rocketmq运行目录/store/commitlog
# consume queue文件的存储路径
storePathConsumeQueue=你的rocketmq运行目录/store/consumequeue
# 消息索引文件的存储路径
storePathIndex=你的rocketmq运行目录/store/index
# checkpoint文件的存储路径
storeCheckpoint=你的rocketmq运行目录/store/checkpoint
# abort文件的存储路径
abortFile=你的rocketmq运行目录/abort
# 设置topic会自动创建
autoCreateTopicEnable=true
```

所以只要我们基于上述的broker配置文件来启动broker，那么他就会跟指定的nameserver来进行通信，然后在指定的目录里存放各种数据文件，包括在运行目录的logs目录里写入他自己的日志。

同时我们别忘了，在rocketmq-master源码目录下的distribution里，有一个logback-broker.xml，需要把这个拷贝到运行目录的conf目录中去，然后修改里面的地址，把${user.hom}都修改为你的rocketmq运行目录。

3、使用debug模式启动Broker

接着我们就可以使用debug模式启动BrokerStartup类了，右击他点击Debug BrokerStartup.main()，就可以启动他。

接着我们会看到如下的一段提示，就说明broker启动成功了：

```
Connected to the target VM, address: '127.0.0.1:55275', transport: 'socket'
The broker[broker-a, 192.168.3.9:10911] boot success. serializeType=JSON
```

然后我们在rocketmq运行目录下的logs中，会找到一个子目录是rocketmqlogs，里面有一个broker.log，就可以看到Broker的启动日志了，如下所示。

```
2020-02-05 14:19:55 INFO main - Try to start service thread:AllocateMappedFileService started:false lastThread:null
2020-02-05 14:19:56 WARN main - Load default transaction message hook service: TransactionalMessageServiceImpl
2020-02-05 14:19:56 WARN main - Load default discard message hook service: DefaultTransactionalMessageCheckListener
2020-02-05 14:19:56 INFO main - The broker dose not enable acl
2020-02-05 14:19:56 INFO main - Try to start service thread:ReputMessageService started:false lastThread:null
```

```
2020-02-05 14:19:56 INFO main - Try to start service thread:AcceptSocketService started:false lastThread:null
2020-02-05 14:19:56 INFO main - Try to start service thread:GroupTransferService started:false lastThread:null
2020-02-05 14:19:56 INFO main - Try to start service thread:HAClient started:false lastThread:null
2020-02-05 14:19:56 INFO main - Try to start service thread:FlushConsumeQueueService started:false lastThread:null
2020-02-05 14:19:56 INFO main - Try to start service thread:FlushRealTimeService started:false lastThread:null
2020-02-05 14:19:56 INFO main - Try to start service thread:StoreStatsService started:false lastThread:null
2020-02-05 14:19:56 INFO main - Try to start service thread:FileWatchService started:false lastThread:null
2020-02-05 14:19:56 INFO FileWatchService - FileWatchService service started
2020-02-05 14:19:56 INFO main - Try to start service thread:PullRequestHoldService started:false lastThread:null
2020-02-05 14:19:56 INFO PullRequestHoldService - PullRequestHoldService service started
2020-02-05 14:19:56 INFO main - Try to start service thread:TransactionalMessageCheckService started:false lastThread:null
2020-02-05 14:19:56 INFO main - The broker[broker-a, 192.168.3.9:10911] boot success. serializeType=JSON
2020-02-05 14:20:06 INFO BrokerControllerScheduledThread1 - dispatch behind commit log 0 bytes
2020-02-05 14:20:06 INFO BrokerControllerScheduledThread1 - Slave fall behind master: 0 bytes
```

这就说明Broker已经启动成功了，下一篇文章我们就可以测试下本地启动的BrokerMQ可不可以进行消息的发送和消费了。

End

重要说明：

如何提问：每篇文章都有评论区，大家可以尽情留言提问，我会逐一答疑

如何加群：购买狸猫技术窝专栏的小伙伴都可以加入狸猫技术交流群，一个非常纯粹的技术交流的地方

具体加群方式，请参见目录菜单下的文档：《付费用户如何加群》（**购买后可见**）