

图文 92 如何在IntelliJ IDEA中启动NameServer以及本地调试源码?

481 人次阅读 2020-02-06 08:59:05

[返回](#)
[前进](#)
[重新加载](#)
[打印](#)[详情](#) [评论](#)

狸猫技术

[如何在IntelliJ IDEA中启动NameServer以及本地调试源码?](#)[进店逛](#)**相关频道**

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、为什么要在IntelliJ IDEA中启动RocketMQ?**

上一篇文章我们说到已经把RocketMQ的源码从Github上下载下来，并且导入到IntelliJ IDEA中去了，而且还熟悉了一下RocketMQ的源码结构。

接着我们来考虑一个问题了，如果要分析RocketMQ源码的话，我们是直接就没头没脑的去翻看里面的源码吗？

这种分析源码的方式肯定是错误的，正确的做法，应该是尝试在IntelliJ IDEA中去启动RocketMQ，然后你就可以在源码中打一些断点，去观察RocketMQ源码的运行过程，而且在这个过程中，还需要从RocketMQ实际运行和使用的角度，去观察他的源码运行的流

程。

什么意思呢？比如RocketMQ的使用的时候，刚开始肯定是先启动NameServer，那么我们是不是可以在NameServer的源码中打入断点，然后在IntelliJ IDEA中启动NameServer，接着观察他启动时候的源码运行流程？

接着下一步肯定是启动Broker，那么是不是可以在Broker源码中打入断点，然后在IntelliJ IDEA中启动Broker，去观察他启动时候的源码运行流程？

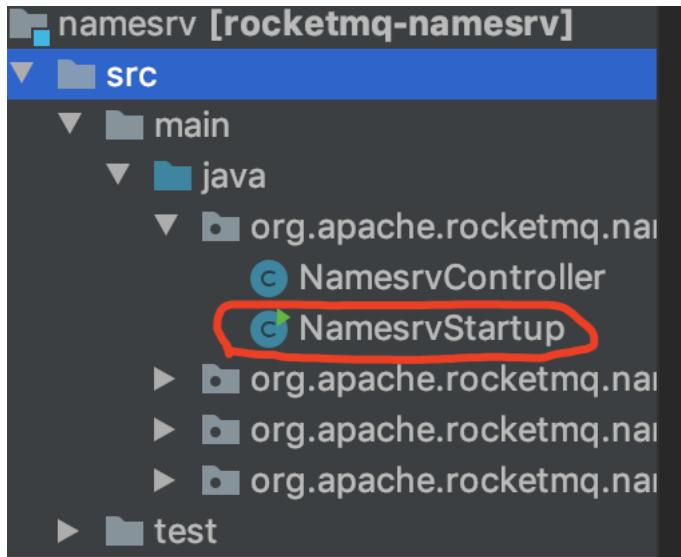
包括我们的客户端发送消息到Broker，Broker的主从同步，实际上我们都可以用这种方式在源码里打断点，然后在IntelliJ IDEA中启动和运行RocketMQ，来观察各种场景下的源码运行流程。

所以我们首先肯定要先能在IntelliJ IDEA中启动和调试RocketMQ的源码，接着才能进一步继续去分析他的源码运行流程。

2、在IntelliJ IDEA中对NameServer启动类进行配置

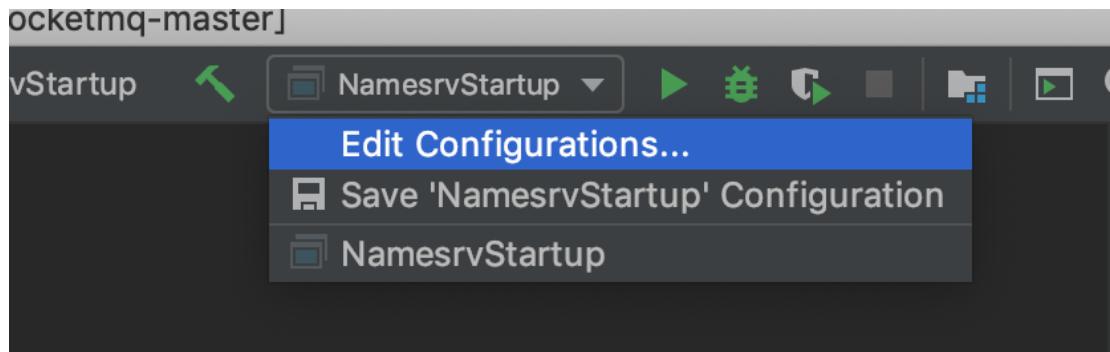
返回
前进
重新加载
打印

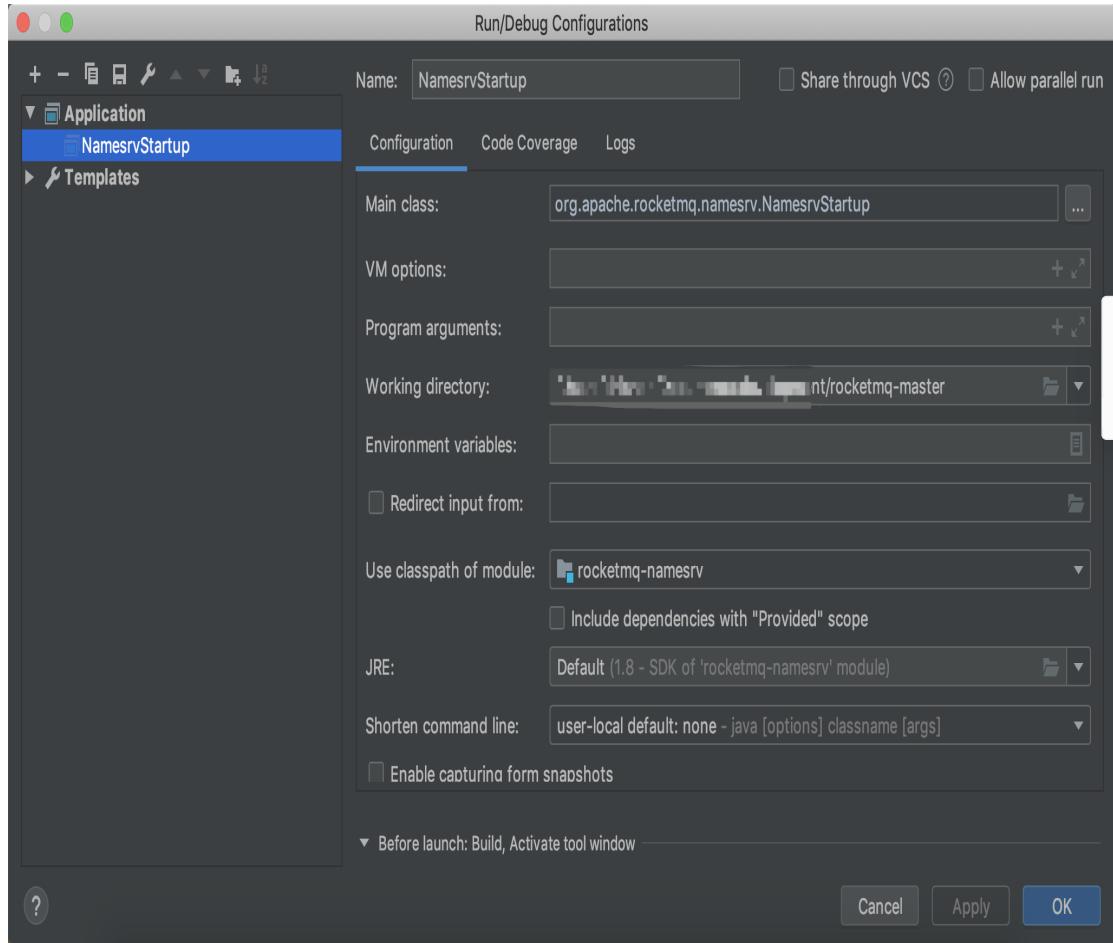
首先，我们需要在IntelliJ IDEA中启动NameServer，所以我们先在RocketMQ源码目录中找到namesrv这个工程，然后展开他的目录，找到NamesrvStartup.java这个类，我们看下面的图示。



然后我们要选中NamesrvStartup这个类，接着在IntelliJ IDEA中的上面可以看到一个Edit Configuration的按钮，我们点击这个按钮

大家看下面的图示，接着会进入一个编辑NamesrvStartup这个启动类的界面。



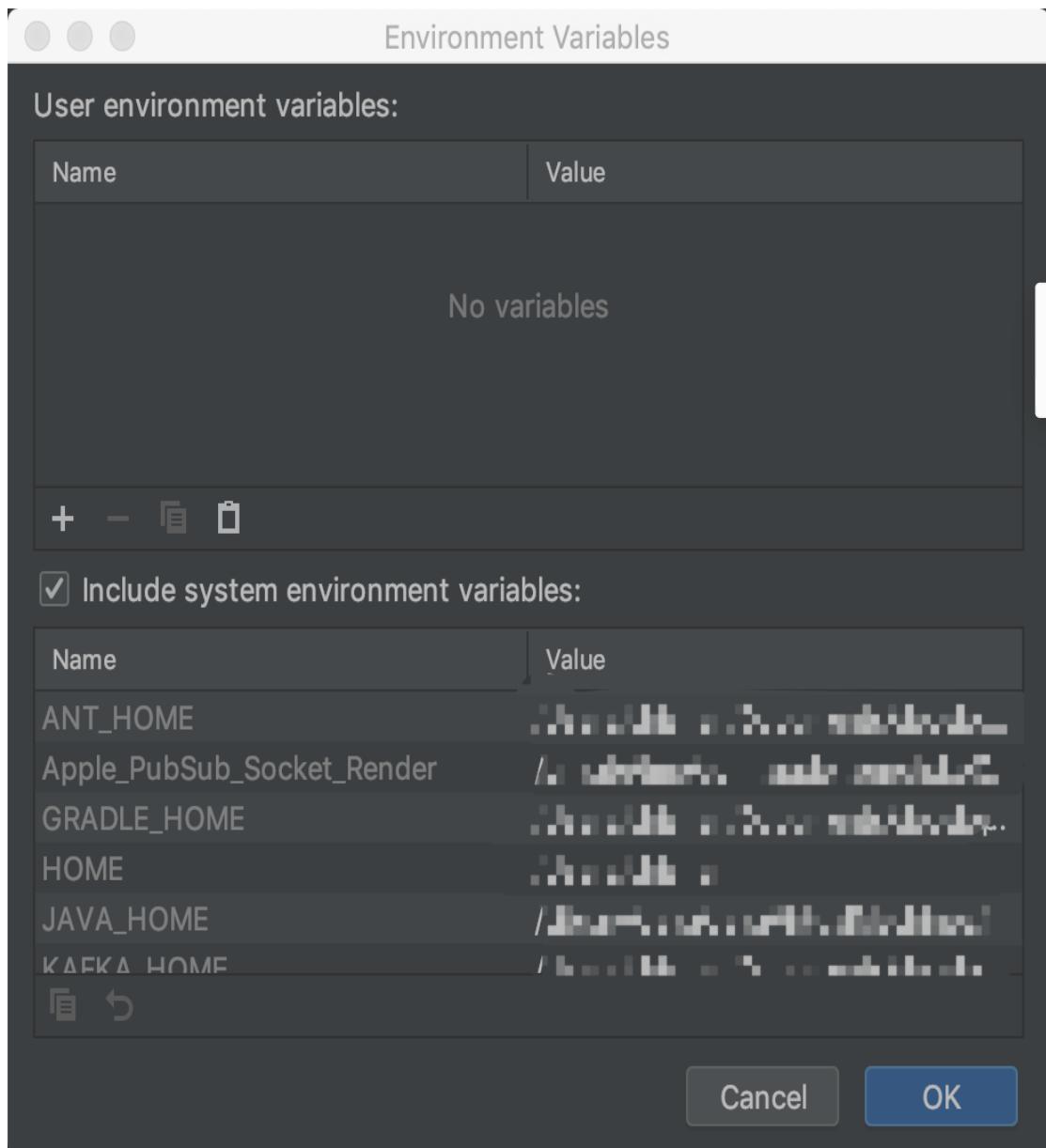


接着我们需要配置一个环境变量，就是ROCKETMQ_HOME，因为NameServer启动的时候他就是要求要有这个环境变量的，所以说我们需要在NamesrvStartup的配置编辑界面里给他加入一个环境变量的配置。

大家其实可以看到，上面的界面中，我们可以给一个类配置很多东西，包括他启动时候的JVM虚拟机的参数（VM options），包括我们要传递给他的main()方法的参数（Program options），这都是很实用的。

而我们要配置的是Environment Variables，就是环境变量，我们在上图中找到这个东西，他右边有一个按钮，大家可以点击一下，就会进入到一个添加环境变量的界面中去，大家看下面的图示。





在上面那个图里有一个+号，你点击就可以添加环境变量，我们添加一个ROCKETMQ_HOME的环境变量，他的值你就输入一个新建的目录就好了

比如：xxx/rocketmq-nameserver，你填入你自己本地的目录，这个不要跟rocketmq-master源码目录一样，因为他是运行目录。

然后回到Edit Configuration的目录，就是点击下面的Apply按钮和OK按钮，就好了，我们就给NamesvrStartup这个启动类设置好了ROCKETMQ_HOME的环境变量了，他启动的时候是可以找到这个环境变量的。

3、在rocketmq运行目录中创建需要的目录结构以及拷贝配置文件

接着我们就需要在rocketmq-nameserver运行目录中创建我们需要的目录结构，此时我们需要创建conf、logs、store三个文件夹，因为后续NameServer运行是需要使用一些目录的。

然后我们把RocketMQ源码目录中的distrbution目录下的broker.conf、logback_namesvr.xml两个配置文件拷贝到刚才新建的conf目录中去，接着就需要修改这两个配置文件。

首先修改logback_namesvr.xml这个文件，修改里面的日志的目录，修改为你的rocketmq运行目录中的logs目录。里面有很多的\${user.home}，你直接把这些\${user.home}全部替换为你的rocketmq运行目录就可以了。

接着就是修改broker.conf文件，改成如下所示：

```
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运行目录的store子目录
# 这是commitLog的存储路径
storePathCommitLog=你的rocketmq运行目录的store子目录/commitlog
# consume queue文件的存储路径
storePathConsumeQueue=你的rocketmq运行目录的store子目录/consumequeue
# 消息索引文件的存储路径
storePathIndex=你的rocketmq运行目录的store子目录/index
# checkpoint文件的存储路径
storeCheckpoint=你的rocketmq运行目录的store子目录/checkpoint
# abort文件的存储路径
abortFile=你的rocketmq运行目录/abort
```

返回
前进
重新加载
打印

4. 启动NameServer

上面的东西都搞定之后, 接着就可以右击NamesrvStartup类, 选择Debug NamesrvStartup.main()了, 就可以用debug模式去启动NameServer了, 他会自动找到ROCKETMQ_HOME环境变量, 这个目录就是你的运行目录, 里面有conf、logs、store几个目录。

他会读取conf里的配置文件, 所有的日志都会打印在logs目录里, 然后数据都会写在store目录里, 启动成功之后, 在IntelliJ IDEA的命令行里就会看到下面的提示。

```
Connected to the target VM, address: '127.0.0.1:54473', transport: 'socket'
The Name Server boot success. serializeType=JSON
```

End

专栏版权归公众号**狸猫技术窝**所有

未经许可不得传播, 如有侵权将追究法律责任

狸猫技术窝精品专栏及课程推荐:

[《从零开始带你成为JVM实战高手》](#)
[《21天互联网Java进阶面试训练营》\(分布式篇\)](#)
[《互联网Java工程师面试突击》\(第1季\)](#)
[《互联网Java工程师面试突击》\(第3季\)](#)

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)

返回
前进
重新加载
打印

