



链滴

SpringCloud Alibaba 微服务实战八 - Seat a 整合 Nacos

作者: [jianzh5](#)

原文链接: <https://ld246.com/article/1577264200396>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



导读：本篇作为SpringCloud Alibaba微服务实战系列的第八篇，主要内容是让Seata使用Nacos作注册配置中心。系列文章，欢迎持续关注。

上篇文章中我们使用的是Seata的file配置实现分布式事务，既然本系列文章用的是Springcloud alibaba全家桶而Seata又恰好支持Nacos，那么我们肯定也将选择Nacos作为Seata的注册配置中心。本篇章我们将一步一步把Seata整合上Nacos，大家伙跟我一起动手试试吧~

注本篇文章依赖于上篇文章[SpringCloud Alibaba微服务实战七 - 分布式事务](#)

配置调整

服务端配置修改

服务端主要是修改`\seata\conf\registry.conf`文件，让Seata使用Nacos作为注册配置中心，修改后文件如下：

```
registry {
  # file、nacos、eureka、redis、zk、consul、etcd3、sofa
  type = "nacos"

  nacos {
    serverAddr = "192.168.136.130:8848"
    namespace = ""
    cluster = "default"
  }
}

config {
  # file、nacos、apollo、zk、consul、etcd3
```

```
type = "nacos"

nacos {
  serverAddr = "192.168.136.130:8848"
  namespace = ""
}
}
```

将Seata配置推到Nacos

此处主要修改`\seata\conf\nacos-config.txt`文件，作如下两处修改：

- 修改tx-service-group

```
transport.thread-factory.share-boss-worker=false
transport.thread-factory.client-selector-thread-prefix=NettyClientSelector
transport.thread-factory.client-selector-thread-size=1
transport.thread-factory.client-worker-thread-prefix=NettyClientWorkerThread
transport.thread-factory.boss-thread-size=1
transport.thread-factory.worker-thread-size=8
transport.shutdown.wait=3
service.vgroup_mapping.my_test_tx_group=default
service.enableDegrade=false
service.disable=false
service.max.commit.retry.timeout=-1
```

如上图所示，修改的格式为`service.vgroup_mapping.${spring.alibaba.seata.tx-service-group}=default`，结合我们的系统服务修改后的结果如下：

```
service.vgroup_mapping.account-service-seata=default
service.vgroup_mapping.product-service-seata=default
service.vgroup_mapping.order-service-seata=default
```

- 开启数据库自动代理

```
support.spring.datasource.autoproxy=true
```

- 将配置推送到Nacos

这里我们在`\seata\conf`下打开git终端，执行`sh nacos-config.sh 192.168.136.130`

名称	修改日期	类型	大小
META-INF	2019/12/18 18:37	文件夹	
db_store.sql	2019/10/16 15:38	SQL 文件	2 KB
db_undo_log.sql	2019/10/16 15:38	SQL 文件	1 KB
file.conf	2019/12/18 17:08	CONF 文件	4 KB
logback.xml	2019/10/16 15:38	XML 文档	3 KB
nacos-config.py	2019/10/16 15:38	PY 文件	1 KB
nacos-config.sh	2019/10/16 15:38	Shell Script	1 KB
nacos-config.txt	2019/12/24 18:54	文本文档	3 KB
registry.conf	2019/12/24 18:44	CONF 文件	1 KB

```

MINGW64/d/project_jianzh5/cloud-aliaba/seata/conf
./nacos-config.sh nacosIp
jianshang11@jianshang11 MINGW64 /d/project_jianzh5/cloud-aliaba/seata/conf
$ ls
db_store.sql  file.conf  META-INF/  nacos-config.sh*  registry.conf
db_undo_log.sql  logback.xml  nacos-config.py*  nacos-config.txt

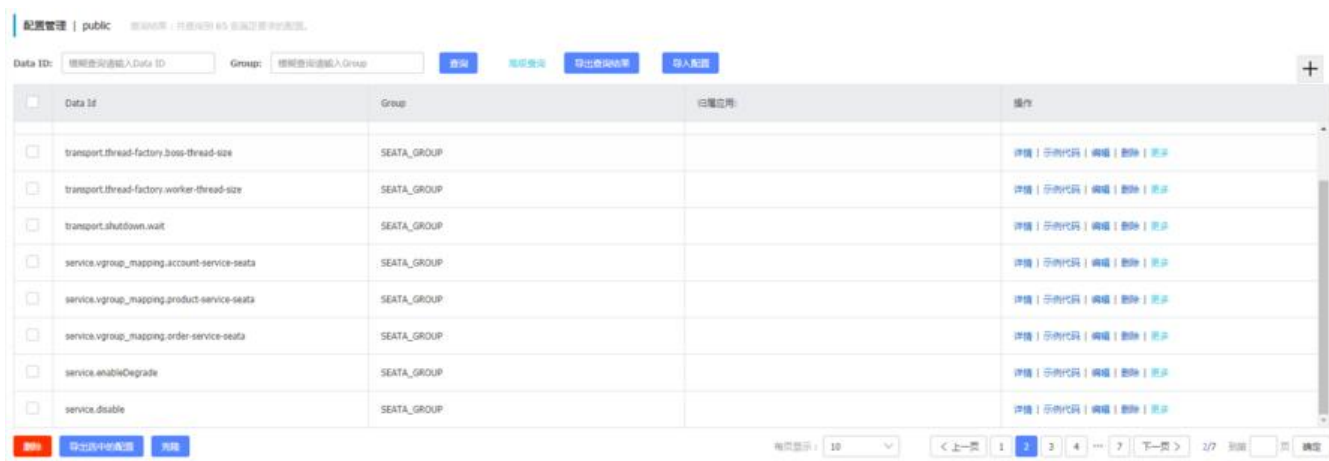
jianshang11@jianshang11 MINGW64 /d/project_jianzh5/cloud-aliaba/seata/conf
$
jianshang11@jianshang11 MINGW64 /d/project_jianzh5/cloud-aliaba/seata/conf
$ sh nacos-config.sh
set nacosIp=10.0.10.48
\r\n set transport.type = TCP
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 4 0 4 0 0 36 0 --:--:-- --:--:-- --:--:-- 36
\033[42;37m true \033[0m
\r\n set transport.server = NIO
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 4 0 4 0 0 64 0 --:--:-- --:--:-- --:--:-- 64
\033[42;37m true \033[0m
\r\n set transport.heartbeat = true

```

执行完成后在git终端看到如下日志说明推送成功

init nacos config finished, please start seata-server.

● 登录nacos控制台验证配置



如上可以看到seata相关的63条配置。

吐槽一下：这配置看起来好乱，期待后续这一块可以优化！

Seata Server端存储模式 (store.mode) 现有file、db两种 (本系列用的是file模式，使用db模式可参照官网配置<http://seata.io/zh-cn/docs/ops/deploy-guide-beginner.html>)，两种模式的区别下：

- file模式为单机模式，全局事务会话信息内存中读写并持久化本地文件root.data，性能较高；

- db模式为高可用模式，全局事务会话信息通过db共享，相应性能差些。

客户端修改

上篇文章中我们在服务项目的resource文件夹中放入了2个文件file.conf， registry.conf，这次我们需要删掉file.conf，然后再将服务端修改后的registry.conf重新拷贝到resource文件夹。

```
registry.conf x
1 registry {
2     # file , nacos , eureka , redis , zk , consul , etcd3 , sofa
3     type = "nacos"
4     nacos {
5         serverAddr = "192.168.136.130:8848"
6         namespace = ""
7         cluster = "default"
8     }
9 }
10 config {
11     # file , nacos , apollo , zk , consul , etcd3
12     type = "nacos"
13     nacos {
14         serverAddr = "192.168.136.130:8848"
15         namespace = ""
16     }
17 }
```

测试

按照如下过程进行测试：

- 启动Seata-Server



登录nacos控制台可以发现Seata已经注册进nacos

- 启动微服务

```

2019-12-25 09:34:59.386 INFO [ServerHandlerThread_1_500]io.seata.core.rpc.DefaultServerMessageListenerImpl.onRegRmMessage:116 -rm register success,message:RegisterRMRequest [resourceIds='jdbc:mysql://10.0.10.48:3306/cloud_alibaba', applicationId='account-service', transactionServiceGroup='account-service-seata'],channel:[id: 0x7b6d2c8a, L:/10.1.61.10:8091 - R:/10.1.61.10:62048]
2019-12-25 09:35:01.500 INFO [NettyServerNIOWorker_2_8]io.seata.core.rpc.DefaultServerMessageListenerImpl.onRegTmMessage:133 -checkAuth for client:10.1.61.10:62065 vgroup:account-service-seata ok
2019-12-25 09:35:31.673 INFO [NettyServerNIOWorker_4_8]io.seata.core.rpc.DefaultServerMessageListenerImpl.onRegTmMessage:133 -checkAuth for client:10.1.61.10:62168 vgroup:product-service-seata ok
2019-12-25 09:35:31.698 INFO [ServerHandlerThread_2_500]io.seata.core.rpc.DefaultServerMessageListenerImpl.onRegRmMessage:116 -rm register success,message:RegisterRMRequest [resourceIds='jdbc:mysql://10.0.10.48:3306/cloud_alibaba', applicationId='product-service', transactionServiceGroup='product-service-seata'],channel:[id: 0x6ef472a0, L:/10.1.61.10:8091 - R:/10.1.61.10:62169]
2019-12-25 09:35:38.822 INFO [ServerHandlerThread_3_500]io.seata.core.rpc.DefaultServerMessageListenerImpl.onRegRmMessage:116 -rm register success,message:RegisterRMRequest [resourceIds='jdbc:mysql://10.0.10.48:3306/cloud_alibaba', applicationId='order-service', transactionServiceGroup='order-service-seata'],channel:[id: 0xe42fc1b6, L:/10.1.61.10:8091 - R:/10.1.61.10:62231]
2019-12-25 09:35:40.900 INFO [NettyServerNIOWorker_6_8]io.seata.core.rpc.DefaultServerMessageListenerImpl.onRegTmMessage:133 -checkAuth for client:10.1.61.10:62250 vgroup:order-service-seata ok

```

在Seata控制台可以看到服务注册

- 业务测试

正常情况会看到如下日志

```

2019-12-25 09:46:23.046 INFO OrderController:27 - create order:OrderDTO(id=null, orderNo=null, accountCode=jianzh5, productCode=P003, count=5, amount=null, price=5)
2019-12-25 09:46:23.062 INFO DefaultGlobalTransaction:106 - Begin new global transaction [10.1.61.10:8091:2030965698]
Time: 51 ms - ID: com.javadaily.order.mapper.OrderMapper.insert
Execute SQL: io.seata.rm.datasource.PreparedStatementProxy@8100f4a

2019-12-25 09:46:23.115 INFO OrderServiceImpl:43 - ORDER XID is: 10.1.61.10:8091:2030965698
2019-12-25 09:46:23.283 INFO DefaultGlobalTransaction:145 - [10.1.61.10:8091:2030965698] commit status:Committed
2019-12-25 09:46:23.619 INFO RmMessageListener:66 - onMessage:xid=10.1.61.10:8091:2030965698,branchId=2030965699,branchType=AT, resourceId=jdbc:mysql://10.0.10.48:3306/cloud_alibaba,applicationData=null
2019-12-25 09:46:23.619 INFO AbstractRMHandler:95 - Branch committing: 10.1.61.10:8091:2030965698 2030965699 jdbc:mysql://10.0.10.48:3306/cloud_alibaba null
2019-12-25 09:46:23.619 INFO AbstractRMHandler:103 - Branch commit result: PhaseTwo_Committed

```

如果出现异常需要回滚会看到这样的日志

```

2019-12-25 09:45:23.126 INFO OrderServiceImpl:43 - ORDER XID is: 10.1.61.10:8091:2030965695
2019-12-25 09:45:23.383 INFO RmMessageListener:66 - onMessage:xid=10.1.61.10:8091:2030965695,branchId=2030965696,branchType=AT, resourceId=jdbc:mysql://10.0.10.48:3306/cloud_alibaba,applicationData=null
2019-12-25 09:45:23.383 INFO AbstractRMHandler:122 - Branch Rollbacking: 10.1.61.10:8091:2030965695 2030965696 jdbc:mysql://10.0.10.48:3306/cloud_alibaba
2019-12-25 09:45:23.397 INFO AbstractUndoLogManager:332 - xid 10.1.61.10:8091:2030965695 branch 2030965696, undo_log deleted with GlobalFinished
2019-12-25 09:45:23.398 INFO AbstractRMHandler:130 - Branch Rollbacked result: PhaseTwo_Rollbacked
2019-12-25 09:45:23.402 INFO DefaultGlobalTransaction:185 - [10.1.61.10:8091:2030965695] rollback status:Rollbacked
2019-12-25 09:45:23.404 ERROR [dispatcherServlet]:175 - Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed; nested exception is feign.FeignException$InternalServerError: status 500 reading ProductFeign#deduct (String,Integer)] with root cause

```

好了，各位朋友们，本期的“SpringCloud Alibaba微服务实战八 - Seata 整合Nacos”到此就全部束啦，咱们下篇见！

one more thing! 如果你觉得这篇文章有点用的话，求“转发”、求“在看”。名人不说暗话，我喜欢这种赶脚！

one more thing again! 再见之前让我再求一波关注，请扫描下方二维码与我联系~



系列文章

- [SpringCloud Alibaba微服务实战七 - 分布式事务](#)
- [SpringCloud Alibaba微服务实战六 - 配置隔离](#)
- [SpringCloud Alibaba微服务实战五 - 限流熔断](#)
- [SpringCloud Alibaba微服务实战四 - 版本管理](#)
- [SpringCloud Alibaba微服务实战三 - 服务调用](#)
- [SpringCloud Alibaba微服务实战二 - 服务注册](#)
- [SpringCloud Alibaba微服务实战一 - 基础环境准备](#)