



链滴

Seata 实战安装部署与 Spring Cloud 应用

作者: [hnuscwy](#)

原文链接: <https://ld246.com/article/1675756057727>

来源网站: 链滴

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

@[TOC](Seata实战安装部署与Spring Cloud应用)

前言：由于项目上使用微服架构，各服务都有独立的数据库，那么如何保证事务的一致性？而Seata一款开源的分布式事务解决方案，致力于提供高性能和简单易用的分布式事务服务。Seata 将为用户供了 AT、TCC、SAGA 和 XA 事务模式，为用户打造一站式的分布式解决方案。

一、Seata版本选择

下载中心

Seata

GitHub: <https://github.com/seata/seata> 发布说明: <https://github.com/seata/seata/releases>

1.6.1 (2022-12-21)

[source](#) | [binary](#)

▶ [Release notes](#)

1.6.0 (2022-12-17)

[source](#) | [binary](#)

▶ [Release notes](#)

1.5.2 (2022-07-12)

[source](#) | [binary](#)

▶ [Release notes](#)

1.5.1 (2022-05-17)

[source](#) | [binary](#)

二、windows下载安装步骤-安装包方式

1、下载安装包1.6.1

2、修改seata/seata/conf/application.yml内容，由于使用nacos作为注册中心主要seata的config、egistry、store

修改前：

```
seata:
  config:
    # support: nacos, consul, apollo, zk, etcd3
    type: file
  registry:
    # support: nacos, eureka, redis, zk, consul, etcd3, sofa
    type: file
  store:
    # support: file 、 db 、 redis
    mode: file
# server:
# service-port: 8091 #If not configured, the default is '${server.port} + 1000'
security:
  secretKey: SeataSecretKey0c382ef121d778043159209298fd40bf3850a017
  tokenValidityInMilliseconds: 1800000
  ignore:
    urls: /,/**/*.css,/**/*.js,/**/*.html,/**/*.map,/**/*.svg,/**/*.png,/**/*.ico,/console-fe/public/**,/api/v1/auth/login
```

修改后

```
seata:
  config:
    # support: nacos, consul, apollo, zk, etcd3
    type: nacos
  nacos:
    server-addr: 127.0.0.1:8848
    namespace: 967df0d8-156b-441a-a5ab-615b4a865168
    group: MY_GROUP
    ##if use MSE Nacos with auth, mutex with username/password attribute
    #access-key: ""
    #secret-key: ""
    data-id: seataServer.properties

  registry:
    # support: nacos, eureka, redis, zk, consul, etcd3, sofa
    type: nacos
    # preferred-networks: 30.240.*
  nacos:
    application: seata-tc-server
    server-addr: 127.0.0.1:8848
    group: MY_GROUP
    namespace: 967df0d8-156b-441a-a5ab-615b4a865168
    #cluster: SH
    ##if use MSE Nacos with auth, mutex with username/password attribute
    #access-key: ""
    #secret-key: ""

  store:
    # support: file 、 db 、 redis
    mode: db
  db:
    datasource: druid
    db-type: mysql
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://127.0.0.1:3306/seata?useUnicode=true&rewriteBatchedStatements=true
    serverTimezone=GMT
```

```
user: root
password: 123456
min-conn: 5
max-conn: 100
global-table: global_table
branch-table: branch_table
lock-table: lock_table
distributed-lock-table: distributed_lock
query-limit: 100
max-wait: 5000
```

3、根据seata/seata/script/server/db/mysql.sql建数据库表seata

```
----- The script used when storeMode is 'db' -----
-----
-- the table to store GlobalSession data
CREATE TABLE IF NOT EXISTS `global_table`
(
  `xid`          VARCHAR(128) NOT NULL,
  `transaction_id` BIGINT,
  `status`       TINYINT  NOT NULL,
  `application_id` VARCHAR(32),
  `transaction_service_group` VARCHAR(32),
  `transaction_name` VARCHAR(128),
  `timeout`      INT,
  `begin_time`   BIGINT,
  `application_data` VARCHAR(2000),
  `gmt_create`   DATETIME,
  `gmt_modified` DATETIME,
  PRIMARY KEY (`xid`),
  KEY `idx_status_gmt_modified` (`status`, `gmt_modified`),
  KEY `idx_transaction_id` (`transaction_id`)
) ENGINE = InnoDB
  DEFAULT CHARSET = utf8mb4;

-- the table to store BranchSession data
CREATE TABLE IF NOT EXISTS `branch_table`
(
  `branch_id`    BIGINT  NOT NULL,
  `xid`          VARCHAR(128) NOT NULL,
  `transaction_id` BIGINT,
  `resource_group_id` VARCHAR(32),
  `resource_id`   VARCHAR(256),
  `branch_type`   VARCHAR(8),
  `status`       TINYINT,
  `client_id`    VARCHAR(64),
  `application_data` VARCHAR(2000),
  `gmt_create`   DATETIME(6),
  `gmt_modified` DATETIME(6),
  PRIMARY KEY (`branch_id`),
  KEY `idx_xid` (`xid`)
) ENGINE = InnoDB
  DEFAULT CHARSET = utf8mb4;
```

```
-- the table to store lock data
CREATE TABLE IF NOT EXISTS `lock_table`
(
  `row_key`    VARCHAR(128) NOT NULL,
  `xid`       VARCHAR(128),
  `transaction_id` BIGINT,
  `branch_id`  BIGINT    NOT NULL,
  `resource_id` VARCHAR(256),
  `table_name` VARCHAR(32),
  `pk`        VARCHAR(36),
  `status`    TINYINT    NOT NULL DEFAULT '0' COMMENT '0:locked ,1:rollbacking',
  `gmt_create` DATETIME,
  `gmt_modified` DATETIME,
  PRIMARY KEY (`row_key`),
  KEY `idx_status` (`status`),
  KEY `idx_branch_id` (`branch_id`),
  KEY `idx_xid` (`xid`)
) ENGINE = InnoDB
DEFAULT CHARSET = utf8mb4;
```

```
CREATE TABLE IF NOT EXISTS `distributed_lock`
(
  `lock_key`   CHAR(20) NOT NULL,
  `lock_value` VARCHAR(20) NOT NULL,
  `expire`     BIGINT,
  primary key (`lock_key`)
) ENGINE = InnoDB
DEFAULT CHARSET = utf8mb4;
```

```
INSERT INTO `distributed_lock` (lock_key, lock_value, expire) VALUES ('AsyncCommitting', ' ', 0);
```

```
INSERT INTO `distributed_lock` (lock_key, lock_value, expire) VALUES ('RetryCommitting', ' ', 0);
```

```
INSERT INTO `distributed_lock` (lock_key, lock_value, expire) VALUES ('RetryRollbacking', ' ', 0);
```

```
INSERT INTO `distributed_lock` (lock_key, lock_value, expire) VALUES ('TxTimeoutCheck', ' ', 0);
```

4、在nacos中新增配置seataServer.properties

```
# 数据存储方式, db代表数据库
store.mode=db
store.db.datasource=druid
store.db.dbType=mysql
store.db.driverClassName=com.mysql.cj.jdbc.Driver
store.db.url=jdbc:mysql://127.0.0.1:3306/seata?useUnicode=true&rewriteBatchedStatements
true&serverTimezone=GMT
```

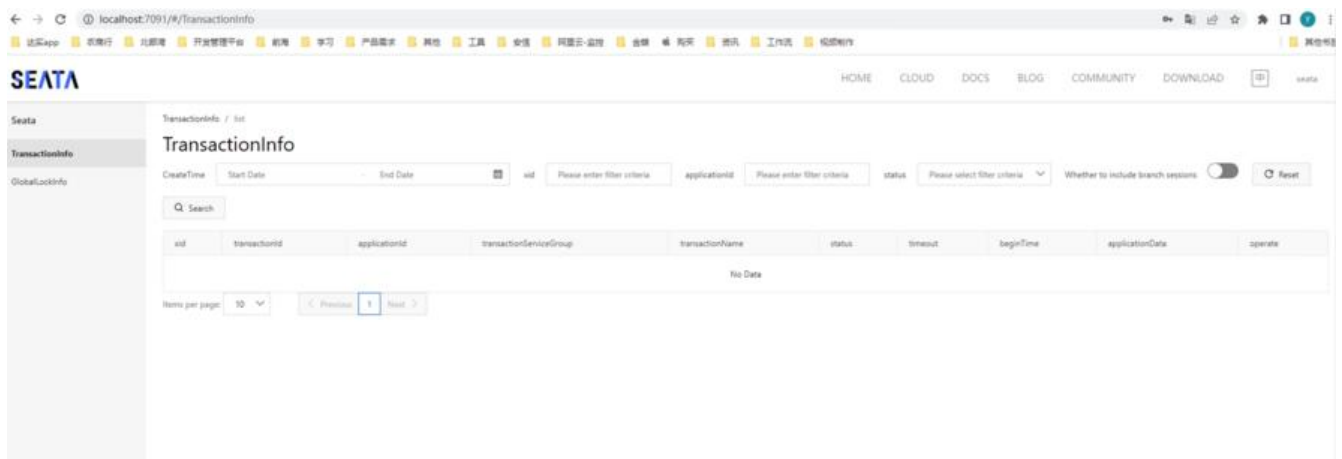
```
store.db.user=root
store.db.password=123456
store.db.minConn=5
store.db.maxConn=30
store.db.globalTable=global_table
store.db.branchTable=branch_table
store.db.queryLimit=100
store.db.lockTable=lock_table
```

```
store.db.maxWait=5000
# 事务、日志等配置
server.recovery.committingRetryPeriod=3000
server.recovery.asynCommittingRetryPeriod=3000
server.recovery.rollbackingRetryPeriod=3000
server.recovery.timeoutRetryPeriod=3000
server.maxCommitRetryTimeout=-1
server.maxRollbackRetryTimeout=-1
server.rollbackRetryTimeoutUnlockEnable=false
server.undo.logSaveDays=7
server.undo.logDeletePeriod=86400000
```

```
# 客户端与服务端传输方式
transport.serialization=seata
transport.compressor=none
# 关闭metrics功能, 提高性能
metrics.enabled=false
metrics.registryType=compact
metrics.exporterList=prometheus
metrics.exporterPrometheusPort=9898
```

5、访问后台<http://127.0.0.1/#/login>

账号: seata 密码: seata



三、linux下载安装步骤-docker方式

参考: 链接: [docker部署seata](#)

```
docker run -d --restart always --name seata-server -p 8091:8091 -p 7091:7091 -v /usr/local/ata/seata-server:/seata-server -e SEATA_PORT=8091 -e SEATA_IP=[自己的ip地址] seataio/seata-server:1.6.1
```

```
# Copyright 1999-2019 Seata.io Group.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
```

```
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

server:
  port: 7091

spring:
  application:
    name: seata-tc-server

logging:
  config: classpath:logback-spring.xml
  file:
    path: ${user.home}/logs/seata
  extend:
    logstash-appender:
      destination: 127.0.0.1:4560
    kafka-appender:
      bootstrap-servers: 127.0.0.1:9092
      topic: logback_to_logstash

console:
  user:
    username: seata
    password: seata

seata:
  config:
    # support: nacos, consul, apollo, zk, etcd3
    type: nacos
    nacos:
      server-addr: 127.0.0.1:8848
      namespace: 967df0d8-156b-441a-a5ab-615b4a865168
      group: MY_GROUP
      ##if use MSE Nacos with auth, mutex with username/password attribute
      #access-key: ""
      #secret-key: ""
      data-id: seataServer.properties
  registry:
    # support: nacos, eureka, redis, zk, consul, etcd3, sofa
    type: nacos
    # preferred-networks: 30.240.*
    nacos:
      application: seata-tc-server
      server-addr: 127.0.0.1:8848
      group: MY_GROUP
      namespace: 967df0d8-156b-441a-a5ab-615b4a865168
      #cluster: SH
      ##if use MSE Nacos with auth, mutex with username/password attribute
      #access-key: ""
```

```

    #secret-key: ""
store:
  # support: file 、 db 、 redis
  mode: db
  db:
    datasource: druid
    db-type: mysql
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://127.0.0.1:3306/seata?useUnicode=true&rewriteBatchedStatements=true
serverTimezone=GMT
    user: root
    password:123456
    min-conn: 5
    max-conn: 100
    global-table: global_table
    branch-table: branch_table
    lock-table: lock_table
    distributed-lock-table: distributed_lock
    query-limit: 100
    max-wait: 5000

server:
  #service-port: 8091 #If not configured, the default is '${server.port} + 1000'
  vgroup-mapping: {<!-- -->"default_tx_group" : "default"}
security:
  secretKey: SeataSecretKey0c382ef121d778043159209298fd40bf3850a017
  tokenValidityInMilliseconds: 1800000
  ignore:
    urls: /,/**/*.css,/**/*.js,/**/*.html,/**/*.map,/**/*.svg,/**/*.png,/**/*.ico,/console-fe/public/*
,./api/v1/auth/login

```

四、Spring Cloud 应用相关配置

seata版本可以根据自己项目版本进行选择，请查看 [版本说明](#)

1、父工程pom文件引入

```

<properties>
  <seata.version>1.6.1</seata.version>
</properties>

</dependencies>
<dependency>
  <groupId>io.seata</groupId>
  <artifactId>seata-all</artifactId>
  <version>${seata.version}</version>
</dependency>
</dependencies>

```

2、子工程pom文件引入

```

<!--分布式事务seata-->
<dependency>
  <groupId>io.seata</groupId>

```



```

    <artifactId>seata-spring-boot-starter</artifactId>
    <version>${seata.version}</version>
</dependency>
<dependency>
  <groupId>com.alibaba.cloud</groupId>
  <artifactId>spring-cloud-starter-alibaba-seata</artifactId>
  <version>${spring.cloud.alibaba.version}</version>
  <exclusions>
    <exclusion>
      <groupId>io.seata</groupId>
      <artifactId>seata-all</artifactId>
    </exclusion>
    <exclusion>
      <artifactId>seata-spring-boot-starter</artifactId>
      <groupId>io.seata</groupId>
    </exclusion>
  </exclusions>
</dependency>

```

3、yml配置

```

# seata 配置， 代替file.conf和registry.conf配置
# seata 配置， 代替file.conf和registry.conf配置
sfs:
  nacos:
    server-addr: 127.0.0.1:8848
    namespace: 967df0d8-156b-441a-a5ab-615b4a865168
    group: MY_GROUP
seata:
  enabled: true
  application-id : ${spring.application.name}
  tx-service-group: default_tx_group
  use-jdk-proxy: true
  enable-auto-data-source-proxy: true
  registry:
    type: nacos
    nacos:
      application: seata-tc-server
      server-addr: ${sfs.nacos.server-addr}
      namespace: ${sfs.nacos.namespace}
      group: ${sfs.nacos.group}
  config:
    type: nacos
    nacos:
      server-addr: ${sfs.nacos.server-addr}
      namespace: ${sfs.nacos.namespace}
      group: ${sfs.nacos.group}
  service:
    vgroupMapping:
      default_tx_group: default

#禁用hystrix导致启动失败
feign:
  hystrix:

```

enabled: false

4、实现类上加上

```
@GlobalTransactional(rollbackFor = Exception.class)
```

5、每个数据库上加上undo_log

```
SET NAMES utf8mb4;
```

```
SET FOREIGN_KEY_CHECKS = 0;
```

```
-----  
-- Table structure for undo_log  
-----
```

```
DROP TABLE IF EXISTS `undo_log`;
```

```
CREATE TABLE `undo_log` (
```

```
  `branch_id` bigint(20) NOT NULL COMMENT '分支事务ID',
```

```
  `xid` varchar(100) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL COMMENT '全事务ID',
```

```
  `context` varchar(128) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL COMMENT '上下文',
```

```
  `rollback_info` longblob NOT NULL COMMENT '回滚信息',
```

```
  `log_status` int(11) NOT NULL COMMENT '状态, 0正常, 1全局已完成',
```

```
  `log_created` datetime(6) NOT NULL COMMENT '创建时间',
```

```
  `log_modified` datetime(6) NOT NULL COMMENT '修改时间',
```

```
  UNIQUE INDEX `ux_undo_log`(`xid`, `branch_id`) USING BTREE
```

```
) ENGINE = InnoDB CHARACTER SET = utf8 COLLATE = utf8_general_ci COMMENT = 'AT transaction mode undo table' ROW_FORMAT = Compact;
```

```
SET FOREIGN_KEY_CHECKS = 1;
```

五、遇到问题

1、No available service

```
2023-02-06 09:51:29.885 [TID: N/A] [http-nio-8671-exec-1] INFO io.seata.tm.TransactionManagerHolder - TransactionManager Singleton io.seata.tm.DefaultTransactionManager@254a608
```

```
2023-02-06 09:51:29.900 [TID: N/A] [http-nio-8671-exec-1] WARN c.d.pi.exception.MyExceptionHandler - No available service
```

```
io.seata.common.exception.FrameworkException: No available service
```

```
    at io.seata.core.rpc.netty.AbstractNettyRemotingClient.loadBalance(AbstractNettyRemotingClient.java:266)
```

解决：逐步排除配置信息，请参考链接：[seata异常信息排除](#)

2、Seata报错 can not get cluster name in registry config 'service.vgroupMapping.default

解决：请参考链接：[seata相关问题解决](#)

3、get table meta of the table XXX error: Failed to fetch schema of XXX

解决：数据表中没有主键，请参考链接：[表设置主键](#)

4、链路中出现某服务报错，各服务不进行回滚

解决：先查看各服务的XID是否一致。

使用spring-cloud-starter-alibaba-seata，已经实现XID传递，注意配置中禁用feign的hystrix（也是不能使用hystrix，参考链接：[hystrix线程切换导致threadLocal丢失问题](#)）

```
<!--分布式事务seata-->
<dependency>
  <groupId>io.seata</groupId>
  <artifactId>seata-spring-boot-starter</artifactId>
  <version>${seata.version}</version>
</dependency>
<dependency>
  <groupId>com.alibaba.cloud</groupId>
  <artifactId>spring-cloud-starter-alibaba-seata</artifactId>
  <version>${spring.cloud.alibaba.version}</version>
  <exclusions>
    <exclusion>
      <groupId>io.seata</groupId>
      <artifactId>seata-all</artifactId>
    </exclusion>
    <exclusion>
      <artifactId>seata-spring-boot-starter</artifactId>
      <groupId>io.seata</groupId>
    </exclusion>
  </exclusions>
</dependency>
```

```
feign:
  hystrix:
    enabled: false
```

参考链接: [2022年SpringCloud Alibaba Seata1.42整合springboot处理分布式事务排坑版教程](#)

参考链接: [Seata分布式事务失效，不生效（事务不回滚）的常见场景](#)

参考链接: [feign调用服务，被调用服务seata事务不开启或者xid为空](#)