Quick Start Guid

English

构建与启动

构建jar包

- build jar file

```
mvn clean package -Ddockerfile.skip
```

- start server

```
java -jar syncer-webapp/syncer-webapp.jar
```

- start in background

```
nohup java -jar syncer-webapp/syncer-webapp.jar &
```

docker构建

- 构建docker镜像

```
mvn clean package
```

- 启动容器

```
docker run -idt -m 2G -p 8000:80 --name redissyncer redissyncer:v2.0.8
```

- docker-compose.yml

```
# deploy local redissyncer service
version: "2.4"

services:
  redissyncer-server:
      image: jiashiwen/redissyncer:v2.0.8
      ports:
```

```
      - "8000:80"
      mem_limit: 2g
      environment:
        SPRING_ENV: "--server.port=80"
      volumes:
        - /etc/localtime:/etc/localtime:ro
        - ./log:/opt/redissyncer/log
      container_name: redissyncer-server
```

相关启动参数配置请参阅配置详情

如何使用

redissyncer为多任务服务端程序,用户通过curl请求方式实现下列操作步骤

使用基本步骤

- 创建同步任务
- 启动同步任务
- 查看同步任务状态
- 停止任务
- 删除任务

redis单实例同步实例

假设redissyncer所在服务器ip地址为10.0.0.100:8080;源redis地址192.168.0.10;目标redis地址192.168.0.20

- 创建同步任务

```
curl -X POST \
  http://10.0.0.100:8080/api/v1/createtask \
  -H 'Content-Type: application/json' \
  -d '{
    "sourcePassword": "xxxxxx",
    "sourceRedisAddress": "192.168.0.10:6379;",
    "targetRedisAddress": "192.168.0.20:6379",
    "targetPassword": "xxxxxx",
    "targetRedisVersion": 4,
    "taskName": "firsttest"
}'
```

返回值

```
{
    "msg": "Task created successfully",
    "code": "2000",
    "data": {
        "taskids": [
```

```
            "10F7B3A0E5344598BAA9F847ADBFF9D6"
        ]
    }
}
```

返回值显示创建任务的"taskids",该id为任务唯一id,便于查询和删除任务时定位;当源库为集群时,任务会依据集群节点数被拆成若干任务,每个任务一个id

- 查看任务状态

```
curl -X POST \
  http://10.0.0.100:8080/api/v1/listtasks \
  -H 'Content-Type: application/json' \
  -d '{
    "regulation": "bynames",
    "tasknames": [
        "firsttest"
    ]
}'
```

- 启动任务

```
curl -X POST \
  http://10.0.0.100:8080/api/v1/starttask \
  -H 'Content-Type: application/json' \
  -d '{
    "taskid": "10F7B3A0E5344598BAA9F847ADBFF9D6"
}'
```

任务启动时"taskid"为必填参数每次仅支持单任务启动

- 停止任务

```
curl -X POST \
  http://10.0.0.100:8080/api/v1/stoptask \
  -H 'Content-Type: application/json' \
  -d '{
    "taskids": [
        "10F7B3A0E5344598BAA9F847ADBFF9D6"
    ]
}'
```

```
"taskids": ["id1","id2" ..."idn"]
```

停止任务支持多任务停止，

- 删除任务

```
curl -X POST \
  http://10.0.0.100:8080/api/v1/removetask \
  -H 'Content-Type: application/json' \
  -d '{
    "taskids": [
        "10F7B3A0E5344598BAA9F847ADBFF9D6"
    ]
}'
```

删除任务状态能为"RUN"

- API及详细参数请参阅api文档
- 更多场景案例请参阅scenecases

使用客户端

客户端使用手册

注意事项

- 迁移中和业务切换之前，请反复观察日志信息，确认是否有异常，日志位置~/log。
- 业务切换之前，请充分检查，特别是数据的一致性。
- redissyncer建议部署在单独空闲机器上,防止资源不足，比如内存，带宽，IOPS。
- 注意RDB传输是否超时；
- redis client buf中的slave项，设置足够大的buffer size和超时时间。
- 动同步任务前，请确认redis源机器是否有足够的内存允许至少生成一个RDB文件。
- 双向同步不保证非幂等命令（INCR、HINCRBY）的数据一致性，除非业务保证不同时对同一个key做非幂等操作