**# flood-distributed-lock**

**## 简介**

基于redis的分布式锁。

<br>

**## 导入**

**\*\*SpringBoot项目导入：\*\***

```xml

<dependency>

<groupId>cn.flood</groupId>

<artifactId>flood-distributed-lock</artifactId>

<version>2.0.0</version>

</dependency>

```

**\*\*jar包方式导入：\*\***

<br>

**## 用法**

**### 在SpringBoot中使用**

在`application.properties`有如下配置可选：

**########################################################**

**### lock setting.**

**########################################################**

flood.distributed.lock.type: "redis"

flood.distributed.lock.redis.host: 10.10.10.51

flood.distributed.lock.redis.port: 6379

flood.distributed.lock.redis.db: 0

flood.distributed.lock.redis.password: winsunion123

flood.distributed.lock.redis.poolConfig.maxTotal: 10

flood.distributed.lock.redis.poolConfig.maxIdle: 10

flood.distributed.lock.redis.poolConfig.minIdle: 2

flood.distributed.lock.redis.poolConfig.maxWaitMillis: 30000

flood.distributed.lock.redis.poolConfig.testOnBorrow: true

flood.distributed.lock.redis.poolConfig.testOnReturn: true

flood.distributed.lock.redis.poolConfig.timeBetweenEvictionRunsMillis: 10000

flood.distributed.lock.redis.poolConfig.minEvictableIdleTimeMillis: 30000

flood.distributed.lock.redis.poolConfig.numTestsPerEvictionRun: -1

**### redis 锁**

```java

//获取分布式锁

// DistributedLock distributedLock = DistributedLocks.getLock("access-service","myLock", 30L, 0);

// if (distributedLock == null) {

// LOGGER.error("Failed to get lock of name : {}", lockName);

// return "Failed to get lock " + lockName;

// }

//

// if (!distributedLock.tryLock()) {

// LOGGER.error("Failed to try lock");

// return "Failed to try lock " + lockName;

// }

//

// 注意，必须用try/finally 或者 try/catch/finally来释放锁

// try {

// 干活

// doSomeThingWithinLock();

// }

// finally {

// distributedLock.unlock();

// }

```