Atitit 数据源关闭问题

目录

[3：根本原因 1](#_Toc3070)

[4：解决方案 1](#_Toc14199)

[使用spring自己对线程池ThreadPoolTaskExecutor 2](#_Toc31806)

# **3：根本原因**

1：SpringBoot Test 主线程退出，导致Spring 容器关闭。  
2：Spring容器关闭，导致DruidDataSource 关闭  
3：此时用户线程去访问已关闭的数据源，导致报错。

这样不行啊，那我测试用例怎么跑完呢？  
可能这个时候有人会想到在测试用例最后面加以下代码：

**TimeUnit.SECONDS.sleep(30);**

可是本人觉得不够优雅，也不灵活。

# **4：解决方案**

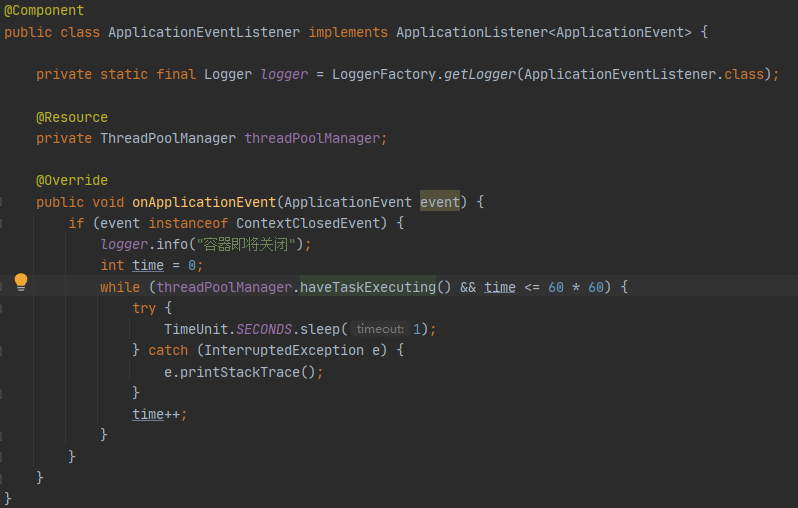
监听容器关闭，关闭前先判断线程池中是否存在未执行玩任务的线程。

使用spring对task任务池.

在总main线程sleep等待线程池启动。。。

总main sleep 等待线程关闭

因此我们是可以知道是否存在未执行完任务的线程的。

所以我在项目中写了以下代码：  
[](https://img2020.cnblogs.com/blog/1537140/202011/1537140-20201117161751115-1371611703.png)

**当线程池中存在未执行完任务的线程的时候，就先等待，直到所有任务完成或超过等待时间。**

可把自己牛逼坏了（叉会腰）！

## 使用spring自己对线程池ThreadPoolTaskExecutor

**private** **static** TaskExecutor iniTaskPool() {

ThreadPoolTaskExecutor taskExecutor = **new** ThreadPoolTaskExecutor();

taskExecutor.setCorePoolSize(Runtime.*getRuntime*().availableProcessors()-1);

taskExecutor.setMaxPoolSize(Runtime.*getRuntime*().availableProcessors());

**return** taskExecutor;

}