



链滴

Spring 定时器在 Tomcat 上执行两次的问题

作者: [96XL](#)

原文链接: <https://ld246.com/article/1566874521913>

来源网站: [链滴](#)

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

前言

  springboot项目中用到了定时器，用的是spring的@Scheduled注解，每天早上九点执行，发送公众号消息提醒。今天接收到了提醒，发现提醒发送了两次，然后查看日志文件现定时任务到点执行了两次。百度了一下发现有几种说法，总结一下。

正文

1. 第一种说法，因为没有移除springboot项目的内置tomcat，所以在使用外部tomcat运行时springboot项目启动了两次，移除内置的tomcat即可。这种方法对我没有用，因为我本来就已经移除了内置tomcat。

```
<!--移除内置tomcat -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
  <exclusions>
    <exclusion>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-tomcat</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

2. 第二种说法，tomcat的server.xml配置错误，项目启动时会去找appBase目录下的项目加载一次，后再去找docBase目录下的项目再加载一次。所以同一个项目被加载了两次，定时器也被加载了两次。将文件中的<Host>修改为如下配置，appBase置空，docBase修改为项目的绝对路径即可。这种方法对我来说也没有用，因为我服务器上的tomcat本来就是这么配置的。

```
<Host name="你的域名" appBase="" unpackWARs="true" autoDeploy="true">
  <Context path="" docBase="项目的绝对路径，如/opt/tomcat/webapps/abc" reloadable="true" privileged="true" debug="0"/>
  <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" prefix="localhost_access_log." suffix=".txt" pattern="%h %l %u %t &quot;%r&quot; %s %b" />
</Host>
```

  还有其他修改server.xml的方法，具体请参考 <https://www.nblogs.com/Syney/p/7678714.html>。这篇博客里的其他方法我没有尝试，因为我在一个tomcat部署了多个项目，其他方法并不适用，所以我也没有尝试。关于appBase和docBase的区别可以参考<https://blog.csdn.net/sqjucheng/article/details/8510058>。

3. 第三种说法，在java代码中手动阻止项目第一次启动加载。

  项目被部署在服务器tomcat的webapps目录里，导致项目被tomcat初始化2次，部署成功了2次，一次访问路径是项目名，一次访问路径是/。实际中那个访问路径带项目名的不需要的，所以直接阻止它启动，这样项目就只成功启动了一次，问题就可以解决。

  阻止具体方法，创建一个bean，实现ServletContextAware接口，重写setServletContext()方法。当这个bean创建时，会自动调用setServletContext()，在方法里我们判断下当前项目访问路径是否为空或者是否为"/"，如果是，正常通过。如果不是，说明当前tomcat正在初始访问路径为项目名的项目，所以我们要阻止它，这时候抛出个运行异常，当前bean就会创建失败，时候这个项目就会启动失败了。

这个接口在哪儿实现都可以，我直接在定时器的配置类中实现了这个接口。

```
public class AsyncConfig implements ServletContextAware

@Override
public void setServletContext(ServletContext servletContext) {
    String contextPath = servletContext.getContextPath();
    if ("".equals(contextPath) || "/".equals(contextPath)) {
        // 这句日志打印可以不要
        logger.info("启动成功，本次启动映射路径为：" + contextPath);
    } else {
        throw new RuntimeException("为阻止tomcat初始化2次,导致spring定时器启动2次,阻止本次启动,本次启动映射路径为：" + contextPath);
    }
}
}
```

我用的是第三种方法，这种方法在项目第一次加载时会抛出大量重复的运行时异常，不止一次。第二次加载恢复正常，启动成功，然后测试一下，定时任务只会执行一次。