<http://www.cnblogs.com/hustzzl/p/7056487.html>

在开发中，有时候我们会想在项目启动时就执行某些操作，如将某些存在数据库里的数据刷到内存里以便在项目里快速使用这些数据、跑一些批处理。

　　此处介绍两类方法：

第一类：

　　项目启动时，利用spring容器初始化bean来实现。

　　共3种方法： （1）通过@PostConstruct方法实现初始化bean进行操作

　　　　　　　　（2）通过在bean相关的xml配置文件中配置init-method方法

　　　　　　　　（3）通过bean实现InitializingBean接口

第二类：

　　项目启动后，通过quartz，立即执行该操作。

　　（4）通过org.springframework.scheduling.quartz.SimpleTriggerBean方式，可以配置间隔多长时间执行一次任务，如<property name="repeatInterval" value="3000" />就是指定3秒执行一次任务。

　　下面逐一介绍这4种方法：

方法一：@PostConstruct方法

　　在实现类和方法上加注解，类上加bean注解，方法上加@PostConstruct注解。

[复制代码](javascript:void(0);)

1 //本人此类是在将黑名单从数据库中查询出来，并放到内存

2 @Service("phoneBlacklistCache")

3 public class PhoneBlacklistCache {

4 　　public List<String> phoneBlacklist = new ArrayList<String>();

6 　　//次注解是操作的关键

9 　　@PostConstruct

10 　　public void init(){

11 　　　　//想进行的操作

12 　　　　//比如：查询数据库，写到内存

13 }

14 }

[复制代码](javascript:void(0);)

方法二：init-method方法

　　在xml文件里配置bean，bean的配置中配置init-method方法（该方法配置成你想要执行的操作方法）。

<bean id="PoneBlacklistCache" class="com.jd.qrx.trade.cache.impl.PhoneBlacklistCache" scope="singleton" init-method="init"></bean>

　　定义类文件。

[复制代码](javascript:void(0);)

1 public class PhoneBlacklistCache {

2 　　public List<String> phoneBlacklist = new ArrayList<String>();

3 　　public void init(){

4 　　　　//想进行的操作

5 　　　　//比如：查询数据库，写到内存

6 }

7 }

[复制代码](javascript:void(0);)

方法三：InitializingBean方法

　　定义相应类实现InitializingBean接口。

[复制代码](javascript:void(0);)

1 public class PhoneBlacklistCache implements InitializingBean{

2 public List<String> phoneBlacklist = new ArrayList<String>();

3 @Override

4 public void afterPropertiesSet() throws Exception {

5 　　　　//想进行的操作

6 　　　　//比如：查询数据库，写到内存

7 }

8 }

[复制代码](javascript:void(0);)

　　其中，afterPropertiesSet()方法里写想要操作的代码。

方法四：quartz方法

　　通过将SimpleTriggerBean配置成项目启动后立即执行，且重复执行次数配置成0，不重复执行。即可达到项目启动立即执行一此操作的目的。

具体配置如下：

　　1、我的项目是springMVC框架，在web.xml中作如下配置：

[复制代码](javascript:void(0);)

1 <context-param>

2 <param-name>contextConfigLocation</param-name>

3 <param-value>

4 classpath:conf/spring-config.xml,

5 classpath:conf/quartz-config.xml

6 </param-value>

7 </context-param

[复制代码](javascript:void(0);)

　　2、然后在quartz-config.xml中做如下配置

[复制代码](javascript:void(0);)

1 <?xml version="1.0" encoding="UTF-8"?>

2 <beans xmlns="http://www.springframework.org/schema/beans"

3 xmlns:context="http://www.springframework.org/schema/context" xmlns:tx="http://www.springframework.org/schema/tx"

4 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

5 xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd

6 http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd

7 http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.0.xsd"

8 default-lazy-init="false">

9 <bean id="quartzJob" class="xx.xx.xx.QuartzJob"/>

10 <bean id="jobDetail" class="org.springframework.scheduling.quartz.MethodInvokingJobDetailFactoryBean">

11 <property name="targetObject">

12 <ref bean="quartzJob"/>

13 </property>

14 <property name="targetMethod">

15 <value>xxxxxx(你在QuartzJob中的方法)</value>

16 </property>

17 </bean>

18

19 <!-- 项目启动后任务就执行一次 -->

20 <bean id="simpleTrigger" class="org.springframework.scheduling.quartz.SimpleTriggerBean">

21 <property name="jobDetail" ref="jobDetail"/>

22 <property name="startDelay" value="500"/>

23 <property name="repeatInterval" value="0"/>

24 <property name="repeatCount" value="0"/>

25 </bean>

26

27 <span style="white-space:pre"></span>

28 <bean id="startQuertz" class="org.springframework.scheduling.quartz.SchedulerFactoryBean">

29 <property name="triggers">

30 <list>

31 <ref bean="simpleTrigger"/>

32 </list>

33 </property>

34 </bean>

35 </beans>

[复制代码](javascript:void(0);)