



Spring在Web容器启动时执行初始化方法

转载 2018年02月24日 10:26:00 标签：spring / web / 启动执行 / 初始化

6 编辑 删除

需求：在tomcat启动时开启一个定时任务。

想法：容器启动时执行方法，最容易想到的就是servlet中可以配置load-on-startup，设置一个正整数也就可以随容器一起启动。

问题：上面的方法很好，但是由于定时任务需要去操作数据库，而项目采用了spring的依赖注入来管理对象，而servlet并不受Spring的管理。若此时在servlet中注入Spring管理的对象，则会报错：javax.naming.NameNotFoundException: Name com.test.InitServlet is not bound in this Context。

所以想要再servlet中操作数据库，只能手动去创建一个service，这显然违背了我们使用Spring的初衷，让项目看起来不伦不类的。那么如何才能在启动WEB容器的时候执行一段初始化代码，并且可以让其被Spring管理呢？

解决方案：Spring提供了当一个Bean初始化后执行方法的扩展点：InitializingBean。这里的初始化指的就是当该Bean的属性被注入完成后（注意：这里并不是所有属性都需要被设置），所以InitializingBean接口提供的方法名称也很形象：afterPropertiesSet()。

使用的时，将该Bean注入到Spring容器，之后我们便可以获得Spring容器中的对象了，也就是说，可以得到service方法来执行我们的定时任务了。

具体代码如下：

```
[java]
1.  @Component
2.  public class InitServlet implements InitializingBean {
3.
4.      /**
5.       *
6.       */
7.      private static final long serialVersionUID = 1L;
8.
9.      @Resource
10.     private DispatchesService dispatchesService;
11.
12.     @Override
13.     public void afterPropertiesSet() throws Exception {
14.         dispatchesService.spyDDetails();
15.     }
16.
17. }
```

另外还有两种方法也可以实现如上的功能。

- 1、若采用XML来配置Bean的话，可以指定属性init-method
- 2、通过注解@PostConstruct来修改初始化方法

值得注意的是，三者可以同时存在，触发的顺序是先触发@PostConstruct修饰的方法，再触发afterPropertiesSet()，最后触发init-method

其中@PostConstruct是通过注册一个BeanPostProcessor，在Bean的初始化方法之前调用，而afterPropertiesSet()和init-method都在初始化方法中调用

关于@PostConstruct详细的介绍可以看这里：<http://blog.csdn.net/yaerfeng/article/details/8447530>

下面是Spring中调用Bean的初始化代码的源代码：

```
[java]
1.  protected void invokeInitMethods(String beanName, final Object bean, RootBeanDefinition mbd)
2.      throws Throwable {
3.
4.      boolean isInitializingBean = (bean instanceof InitializingBean);
5.      if (isInitializingBean && (mbd == null || !mbd.isExternallyManagedInitMethod("afterPropertiesSet"))) {
6.          if (logger.isDebugEnabled()) {
7.              logger.debug("Invoking afterPropertiesSet() on bean with name '" + beanName + "'");
8.          }
9.          if (System.getSecurityManager() != null) {
10.             try {
11.                 AccessController.doPrivileged(new PrivilegedExceptionAction<Object>() {
12.                     public Object run() throws Exception {
13.                         ((InitializingBean) bean).afterPropertiesSet(); // 这里触发afterPropertiesSet
14.                     }
15.                 }, getAccessControlContext());
16.             }
17.             catch (PrivilegedActionException pae) {
18.                 throw pae.getException();
19.             }
20.             else {
21.                 ((InitializingBean) bean).afterPropertiesSet(); // 这里触发afterPropertiesSet
22.             }
23.         }
24.     }
25.
26.     if (mbd != null) {
27.         String initMethodName = mbd.getInitMethodName(); // 这里是触发init-method
28.         if (initMethodName != null && !(isInitializingBean && "afterPropertiesSet".equals(initMethodName)) &&
29.             !mbd.isExternallyManagedInitMethod(initMethodName)) {
30.             invokeCustomInitMethod(beanName, bean, mbd);
31.         }
32.     }
33. }
34. }
```

补充：

还有一种方法，是当Spring将所有的Bean都初始化完成后，会留给我们一个入口，我们可以实现如下接口

```
[java]
1.  @Component
2.  public class InstantiationTracingBeanPostProcessor implements
3.      ApplicationListener<ContextRefreshedEvent> {
4.
5.      @Override
6.      public void onApplicationEvent(ContextRefreshedEvent arg0) {
7.          System.out.println("-----所有Bean载入完成---");
8.      }
9.  }
```



发表你的评论

spring在web容器启动时执行初始化方法



u012345283 2014年09月25日 23:20 15192

spring在web容器启动时执行初始化方法

spring 在web容器启动时执行初始化方法



hbiao68 2016年09月07日 17:26 40

需求说明：最近设计了一个“权限管理系统”，在用户登录的时候就需要知道用户对资源所拥有的权限，如果每个用户在登录的时候去查询对应的表（8张），因此对数据库和系统都是极大的负担，因此想缓存用户权限相关表的...