ringbean定义的依赖性检查 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/289/2021_2022_ringbean_E5_AE_c104_289948.htm 在spring的工程里，bean的定义是不可避免的。虽然有各种可以减少很多bean的配置，特别是action众多的时候，我们通常可以通过引入autowire拦截器来避免action类bean的定义。但还会有众多的bean需要定义，他们之间的依赖关系通常也是复杂、多变的。在这过程中我们通常会发生一些弱智的错误，而这些错误也通常需要等到spring容器启动后，甚至在页面调到相应功能时才能被发现（其实你如果有写单元测试的习惯是不会发生这种情况的）。下面是我的一个简单的junit单元测试，我用来对bean的依赖检查，在每加入新bean或是改动到bean代码后 跑一下，希望对初学者有所帮助。 ublic class SpringFactoryTest extends AbstractDependencyInjectionSpringContextTests ...{ @Override protected String[] getConfigLocations() ...{ return new String[] ...{ "classpath:applicationContext-*.xml" }. } public void testAllBeanDefined() throws Exception ...{ System.out.println("There are " applicationContext.getBeanDefinitionCount() " beans was defined."). String[] beanNames = applicationContext.getBeanDefinitionNames(). Integer errorCount = 0. for (String beanName : beanNames) ...{ try ...{ Object bean = applicationContext.getBean(beanName). assertNotNull(bean). } catch (BeanIsAbstractException e) ...{ } catch (Exception e) ...{ errorCount . System.out.println(e.getMessage()). } } if (errorCount

> 0)...{ System.out.println("There are " errorCount " beans error"). fail(). } } } 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com