如何正确控制springboot中bean的加载顺序总结

铂赛东
1.2k • 3
关注作者

如何正确控制springboot中bean的加载顺序总结

铂赛东
1.2k • 3

发布于
2020-07-13 上海

1.为什么需要控制加载顺序

`springboot`遵从约定大于配置的原则，极大程度的解决了配置繁琐的问题。在此基础上，又提供了spi机制，用`spring.factories`可以完成一个小组件的自动装配功能。

在一般业务场景，可能你不大关心一个bean是如何被注册进spring容器的。只需要把需要注册进容器的bean声明为`@Component`即可，spring会自动扫描到这个Bean完成初始化并加载到spring上下文容器。

而当你在项目启动时需要提前做一个业务的初始化工作时，或者你正在开发某个中间件需要完成自动装配时。你会声明自己的Configuration类，但是可能你面对的是好几个有互相依赖的Bean。如果不加以控制，这时候可能会报找不到依赖的错误。

但是你明明已经把相关的Bean都注册进spring上下文了呀。这时候你需要通过一些手段来控制springboot中的bean加载顺序。

2.几个误区

在正式说如何控制加载顺序之前，先说2个误区。

如何正确控制springboot中bean的加载顺序总结

铂赛东
1.2k • 3
关注作者

被parse成spring内部对象的这一过程，但是加载方式并没有大的改变。

利用@Order这个标注能进行加载顺序的控制

严格的说，不是所有的Bean都可以通过@Order这个标注进行顺序的控制。你把@Order这个标注加在普通的方法上或者类上一点鸟用都没有。

那@Order能控制哪些bean的加载顺序呢，我们先看看官方的解释：

```
{@code @Order} defines the sort order for an annotated component. Since Spring 4.0, ann
```

最开始@Order注解用于切面的优先级指定；在 4.0 之后对它的功能进行了增强，支持集合的注入时，指定集合中 bean 的顺序，并且特别指出了，它对于但实例的 bean 之间的顺序，没有任何影响。

目前用的比较多的有以下3点：

- 控制AOP的类的加载顺序，也就是被@Aspect标注的类
- 控制ApplicationListener实现类的加载顺序
- 控制CommandLineRunner实现类的加载顺序

3.如何控制

3.1@DependsOn

@DependsOn注解可以用来控制bean的创建顺序，该注解用于声明当前bean依赖于另外一个bean。所依赖的bean会被容器确保在当前bean实例化之前被实例化。

示例：

```
@Configuration
public class BeanOrderConfiguration {

    @Bean
    @DependsOn("beanB")
    public BeanA beanA(){
```

如何正确控制springboot中bean的加载顺序总结

铂赛东
1.2k • 3
关注作者

```
    public BeanB beanB(){
        System.out.println("bean B init");
        return new BeanB();
    }

    @Bean
    @DependsOn({"beanD","beanE"})
    public BeanC beanC(){
        System.out.println("bean C init");
        return new BeanC();
    }

    @Bean
    @DependsOn("beanE")
```

以上代码bean的加载顺序为：

```
bean B init
bean A init
bean E init
bean D init
bean C init
```

@DependsOn的使用：

- 直接或者间接标注在带有@Component注解的类上面;
- 直接或者间接标注在带有@Bean注解的方法上面;
- 使用@DependsOn注解到类层面仅仅在使用 component-scanning 方式时才有效，如果带有@DependsOn注解的类通过XML方式使用，该注解会被忽略，<bean depends-on="..."/>这种方式会生效。

3.2 参数注入

在@Bean标注的方法上，如果你传入了参数，springboot会自动会为这个参数在spring上下文里寻找这个类型的引用。并先初始化这个类的实例。

利用此特性，我们也可以控制bean的加载顺序。

示例：

如何正确控制springboot中bean的加载顺序总结

铂赛东
1.2k • 3
关注作者

```
    System.out.println("bean A init");
    return new BeanA();
  }

  @Bean
  public BeanB beanB(){
    System.out.println("bean B init");
    return new BeanB();
  }
```

以上结果，beanB先于beanA被初始化加载。

需要注意的是，springboot会按类型去寻找。如果这个类型有多个实例被注册到spring上下文，那你就需要加上`@Qualifier("Bean的名称")`来指定

3.3 利用bean的生命周期中的扩展点

在spring体系中，从容器到Bean实例化&初始化都是有生命周期的，并且提供了很多的扩展点，允许你在这些步骤时进行逻辑的扩展。

这些可扩展点的加载顺序由spring自己控制，大多数是无法进行干预的。我们可以利用这一点，扩展spring的扩展点。在相应的扩展点加入自己的业务初始化代码。从来达到顺序的控制。

具体关于spring容器中大部分的可扩展点的分析，之前已经写了一篇文章详细介绍了：《Springboot启动扩展点超详细总结，再也不怕面试官问了》。

3.4 @AutoConfigureOrder

这个注解用来指定配置文件的加载顺序。但是在实际测试中发现，以下这样使用是不生效的：

```
@Configuration
@AutoConfigureOrder(2)
public class BeanOrderConfiguration1 {
    @Bean
    public BeanA beanA(){
        System.out.println("bean A init");
        return new BeanA();
    }
}
```

如何正确控制springboot中bean的加载顺序总结

铂赛东
1.2k • 3
关注作者

```
    public BeanB beanB(){
        System.out.println("bean B init");
        return new BeanB();
    }
}
```

无论你2个数字填多少，都不会改变其加载顺序结果。

那这个`@AutoConfigureOrder`到底是如何使用的呢。

经过测试发现，`@AutoConfigureOrder`只能改变外部依赖的`@Configuration`的顺序。如何理解是外部依赖呢。

能被你工程内部scan到的包，都是内部的Configuration，而spring引入外部的Configuration，都是通过spring特有的spi文件：`spring.factories`

换句话说，`@AutoConfigureOrder`能改变`spring.factories`中的`@Configuration`的顺序。

具体使用方式：

```
@Configuration
@AutoConfigureOrder(10)
public class BeanOrderConfiguration1 {
    @Bean
    public BeanA beanA(){
        System.out.println("bean A init");
        return new BeanA();
    }
}

@Configuration
@AutoConfigureOrder(1)
public class BeanOrderConfiguration2 {
    @Bean
    public BeanB beanB(){
        System.out.println("bean B init");
        return new BeanB();
    }
}
```

如何正确控制springboot中bean的加载顺序总结

铂赛东
1.2k • 3
关注作者

```
com.example.demo.BeanOrderConfiguration2
```

4.总结

其实在工作中，我相信很多人碰到过复杂的依赖关系的bean加载，把这种不确定性交给spring去做，还不如我们自己去控制，这样在阅读代码的时候 ， 也能轻易看出bean之间的依赖先后顺序。

5.联系作者

微信关注 ***jishuyuanren*** 获取更多技术干货

springboot spring java

铂赛东

开源作者&内容创作者，专注于架构，开源，微服务，分布式等领域的技术研究和原创分享

1.2k 声望 **10.3k** 粉丝

关注作者

5 条评论

得票 最新

22 18 5

如何正确控制springboot中bean的加载顺序总结

铂赛东
1.2k • 3
关注作者

评论支持部分 Markdown 语法：**粗体** _斜体_ [链接](http://example.com) `代码` - 列表 > 引用。你还可以使用 @ 来通知其他用户。

andy：指出文章中的错误：

在标注了@Configuration的类中，写在前面的@Bean一定会被先注册，这句话是对的。

希望作者写文章的时候严谨一点吧。

• 回复 • 2022-08-27 • 来自北京

> **铂赛东**（作者）：@andy 有了解过生命周期这个概念吗，这句话我说的是对的。
>
> talk is cheap, show me the code
>
> 随便做了一个demo，出来结果并不是ABC，而是BEA
>
> demo位置：https://github.com/bryan31/sp...
>
> 欢迎来辩，希望大家都严谨
>
> • 回复 • 2022-08-30
>
> **andy**：@铂赛东 看在你贴了代码的份上，我就花点时间指出你错在哪里吧：
> 在你的demo中：
> A implements FactoryBean
> B implements BeanFactoryPostProcessor
> E implements InstantiationAwareBeanPostProcessor
> 你还敢说是随便写了一个demo（这个我是真的服）。
> 你了解过FactoryBean、BeanFactoryPostProcessor、InstantiationAwareBeanPostProcessor的生命周期吗？实现了这三个接口的bean已经不是普通bean了，它们的执行顺序
> BeanFactoryPostProcessor>InstantiationAwareBeanPostProcessor>FactoryBean
>
> Although you showed me the code，But your level is just too low ，so you think you code is correct。用中文翻译就是夜郎自大，越是知识局限的人越是固执的认为自己是对的。
> 我也贴上普通bean demo的例子吧：
> https://github.com/mkl3436780...
>
> • 回复 • 2022-09-02
>
> **andy**：@andy 欢迎来辩，希望大家都严谨
>
> • 回复 • 2022-09-02

如何正确控制springboot中bean的加载顺序总结

铂赛东
1.2k • 3
关注作者

推荐阅读

Jvm上如何运行其他语言？JSR223规范最详细讲解

有的同学可能会说，在java项目里执行其他语言，这不吃饱了撑着么，java体系那么庞大，各种工具一应俱全...

铂赛东　阅读 217

刨根问底 Redis， 面试过程真好使

充满寒气的互联网如何在面试中脱颖而出，平时积累很重要，八股文更不能少！下面带来的这篇 Redis 问答...

菜农曰　赞 13　阅读 665

Java 编译器 javac 及 Lombok 实现原理解析

javac 是 Java 代码的编译器12，初学 Java 的时候就应该接触过。本文整理一些 javac 相关的高级用法。...

nullwy　赞 9　阅读 5.4k

我终于会写 Java 的定时任务了！

前言学过定时任务，但是我忘了，忘得一干二净，害怕，一直听别人说：你写一个定时任务就好了。写个定...

god23bin　赞 10　阅读 1.8k

spring boot 锁

由于当前的项目中由于多线程操作同一个实体，会出现数据覆盖的问题，后保存的实体把先保存的实体的数...

weiewiyi　赞 3　阅读 8.9k

记录一个关于 GBK 编码的问题

UTF-8 是一种国际化的编码方式，包含了世界上大部分的语种文字（简体中文字、繁体中文字、英文、日文...

编程码农　赞 5　阅读 1.4k

NB的Github项目，看到最后一个我惊呆了！

最近看到不少好玩的、实用的 Github 项目，就来给大家推荐一把。中国制霸生成器最近在朋友圈非常火的一...

艾小仙　赞 5　阅读 1.4k　评论 1