

编程辅导班 C语言 / C++ / 数据结构
一对一辅导 + 一对一答疑

猛击报名

Spring Boot导入Spring配置

[< 上一节](#)[下一节 >](#)

默认情况下，Spring Boot 中是不包含任何的 Spring 配置文件的，即使我们手动添加 Spring 配置文件到项目中，也不会被识别。那么 Spring Boot 项目中真的就无法导入 Spring 配置吗？答案是否定的。

Spring Boot 为了我们提供了以下 2 种方式来导入 Spring 配置：

- 使用 `@ImportResource` 注解加载 Spring 配置文件
- 使用全注解方式加载 Spring 配置

@ImportResource 导入 Spring 配置文件

在主启动类上使用 `@ImportResource` 注解可以导入一个或多个 Spring 配置文件，并使其中的内容生效。

1. 以 helloworld 为例，在 `net.biancheng.www.service` 包下创建一个名为 `PersonService` 的接口，代码如下。

```
01. package net.biancheng.www.service;
02.
03. import net.biancheng.www.bean.Person;
04.
05. public interface PersonService {
06.     public Person getPersonInfo();
07. }
```

2. 在 `net.biancheng.www.service.impl` 包下创建一个名为 `PersonServiceImpl` 的类，并实现 `PersonService` 接口，代码如下。

```
01. package net.biancheng.www.service.impl;
```



```

02.
03. import net.biancheng.www.bean.Person;
04. import net.biancheng.www.service.PersonService;
05. import org.springframework.beans.factory.annotation.Autowired;
06.
07. public class PersonServiceImpl implements PersonService {
08.     @Autowired
09.     private Person person;
10.
11.     @Override
12.     public Person getPersonInfo() {
13.         return person;
14.     }
15. }

```

3. 在该项目的 resources 下添加一个名为 beans.xml 的 Spring 配置文件，配置代码如下。

```

01. <?xml version="1.0" encoding="UTF-8"?>
02. <beans xmlns="http://www.springframework.org/schema/beans"
03.        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
04.        xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
05.
06.     <bean id="personService" class="net.biancheng.www.service.impl.PersonServiceImpl">
07.     </bean>
08. </beans>

```

4. 修改该项目的单元测试类 HelloWorldApplicationTests，校验 IOC 容器是否已经 personService，代码如下。

```

01. package net.biancheng.www;
02.
03. import net.biancheng.www.bean.Person;
04. import org.junit.jupiter.api.Test;
05. import org.springframework.beans.factory.annotation.Autowired;
06. import org.springframework.boot.test.context.SpringBootTest;
07. import org.springframework.context.ApplicationContext;
08.
09. @SpringBootTest
10. class HelloWorldApplicationTests {
11.
12.     @Autowired
13.     Person person;

```

```

14. //IOC 容器
15. @Autowired
16. ApplicationContext ioc;
17.
18. @Test
19. public void testHelloService() {
20.     //校验 IOC 容器中是否包含组件 personService
21.     boolean b = ioc.containsBean("personService");
22.     if (b) {
23.         System.out.println("personService 已经添加到 IOC 容器中");
24.     } else {
25.         System.out.println("personService 没添加到 IOC 容器中");
26.     }
27. }
28.
29. @Test
30. void contextLoads() {
31.     System.out.println(person);
32. }
33.
34. }

```

6. 运行单元测试代码，结果如下图。

[illegible]

图1：直接添加 Spring 配置到 Spring Boot 项目中

7. 在主启动程序类上使用 `@ImportResource` 注解，将 Spring 配置文件 `beans.xml` 加载到项目中，代码如下。

```
01. package net.biancheng.www;
02.
03. import org.springframework.boot.SpringApplication;
04. import org.springframework.boot.autoconfigure.SpringBootApplication;
05. import org.springframework.context.annotation.ImportResource;
06.
07. //将 beans.xml 加载到项目中
```

```
08. @ImportResource(locations = {"classpath:/beans.xml"})
09. @SpringBootApplication
10. public class HelloworldApplication {
11.
12.     public static void main(String[] args) {
13.         SpringApplication.run(HelloworldApplication.class, args);
14.     }
15.
16. }
```

8. 再次执行测试代码，结果如下图。

```

      ____          _            __ _ _
     /  _ \        | |          |_||_|
    / ___ \\       | |___   _____|_||_|
   /  ___ \|      | |_____|_____||_||_|
  /_/   \_\       |_|_____|_____|_||_|

:: Spring Boot ::                (v2.4.5)


2021-05-08 17:52:42.992 INFO 18828 --- [main] n.b.www.HelloworldApplicationTests : Starting HelloworldApplicationTests using Java 1.8.0_131 on LAPTOP-C67MRMAG with PID
18828 (started by 79350 in D:\eclipse workspace4\helloworld)
2021-05-08 17:52:42.996 INFO 18828 --- [main] n.b.www.HelloworldApplicationTests : No active profile set, falling back to default profiles: default
2021-05-08 17:52:44.277 INFO 18828 --- [main] o.s.o.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2021-05-08 17:52:44.685 INFO 18828 --- [main] n.b.www.HelloworldApplicationTests : Started HelloworldApplicationTests in 2.044 seconds (JVM running for 2.936)
personService 已经添加到 IOC 容器中
2021-05-08 17:52:45.023 INFO 18828 --- [extShutdownHook] o.s.o.concurrent.ThreadPoolTaskExecutor : Shutting down ExecutorService 'applicationTaskExecutor'

Process finished with exit code 0
```

图2: 使用 @ImportResource 注解加载 Spring 配置文件结果

全注解方式加载 Spring 配置

Spring Boot 推荐我们使用全注解的方式加载 Spring 配置，其实现方式如下：

1. 使用 `@Configuration` 注解定义配置类，替换 Spring 的配置文件；
2. 配置类内部可以包含有一个或多个被 `@Bean` 注解的方法，这些方法会被 `AnnotationConfigApplicationContext` 或 `AnnotationConfigWebApplicationContext` 类扫描，构建 bean 定义（相当于 Spring 配置文件中的 `<bean>` `</bean>` 标签），方法的返回值会以组件的形式添加到容器中，组件的 id 就是方法名。

以 helloworld 为例，删除主启动类的 @ImportResource 注解，在 net.biancheng.www.config 包下添加一个名为 MyAppConfig 的配置类，代码如下。

```
01. package net.biancheng.www.config;
02.
03. import net.biancheng.www.service.PersonService;
04. import net.biancheng.www.service.impl.PersonServiceImpl;
05. import org.springframework.context.annotation.Bean;
06. import org.springframework.context.annotation.Configuration;
07.
08. /**
09.  * @Configuration 注解用于定义一个配置类，相当于 Spring 的配置文件
```

```

10. * 配置类中包含一个或多个被 @Bean 注解的方法，该方法相当于 Spring 配置文件中的 <bean>
    标签定义的组件。
11. */
12. @Configuration
13. public class MyAppConfig {
14.
15.     /**
16.      * 与 <bean id="personService" class="PersonServiceImpl"></bean> 等价
17.      * 该方法返回值以组件的形式添加到容器中
18.      * 方法名是组件 id（相当于 <bean> 标签的属性 id）
19.      */
20.     @Bean
21.     public PersonService personService() {
22.         System.out.println("在容器中添加了一个组件:peronService");
23.         return new PersonServiceImpl();
24.     }
25. }

```

执行测试代码，执行结果如下图。

```

      ____ _
     / ___ \_/_ _/\__ \| | | |\__ \| | | |
    /___\_\_\/_/___/|_| |_| |___/|_| |_|
( ()\_____|'_'_|_|_|'_\'_'\_____\\
 \\___)l_l_|_|_|_|_|_|_|_) ) ) )
   '|_____|_|_|_|_|_|_|_|_|_|_|_|_|_|
=====|_|=====|_|_|_|_|_|_|_|_|_|
:: Spring Boot ::                (v2.4.5)


2021-05-10 11:02:33.030 INFO 9980 --- [          main] n.b.www.HelloworldApplicationTests : Starting HelloworldApplicationTests using Java 1.8.0_131 on LAPTOP-C67MRNAG with PID 9980
         (started by 79330 in D:\eclipse workspace4\helloworld)
2021-05-10 11:02:33.033 INFO 9980 --- [          main] n.b.www.HelloworldApplicationTests : No active profile set, falling back to default profiles: default
在容器中添加了一个组件:personService
2021-05-10 11:02:34.483 INFO 9980 --- [          main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2021-05-10 11:02:34.759 INFO 9980 --- [          main] n.b.www.HelloworldApplicationTests : Started HelloworldApplicationTests in 2.09 seconds (JVM running for 3.039)
personService 已经添加到 IOC 容器中
2021-05-10 11:02:34.931 INFO 9980 --- [extShutdownHook] o.s.s.concurrent.ThreadPoolTaskExecutor : Shutting down ExecutorService 'applicationTaskExecutor'

Process finished with exit code 0
```

图3：全注解方式添加 Spring 配置

关注公众号「站长严长生」，在手机上阅读所有教程，随时随地都能学习。本公众号由站长亲自运营，长期更新，坚持原创，专注于分享创业故事+学习历程+工作记录+生活日常+编程资料。



微信扫码关注公众号

精美而实用的网站，分享优质编程教程，帮助有志青年。千锤百炼，只为大作；精益求精，处处斟酌；这种教程，看一眼就倾心。

[关于网站](#) | [关于站长](#) | [如何完成一部教程](#) | [公众号](#) | [联系我们](#) | [网站地图](#)

Copyright ©2012-2022 biancheng.net, 冀ICP备2022013920号, 冀公网安备13110202001352号

biancheng.net