



黑马程序员™
www.itheima.com

传智播客旗下
高端IT教育品牌



Spring Boot原理分析

目录 Contents

- ◆ SpringBoot 自动配置
- ◆ SpringBoot 监听机制
- ◆ SpringBoot 启动流程分析

Condition

Condition 是在Spring 4.0 增加的条件判断功能，通过这个可以功能可以实现选择性的创建 Bean 操作。



思考

SpringBoot是如何知道要创建哪个Bean的？比如SpringBoot是如何知道要创建RedisTemplate的？

Condition



案例：需求

在 Spring 的 IOC 容器中有一个 User 的 Bean，现要求：

1. 导入Jedis坐标后，加载该Bean，没导入，则不加载。
2. 将类的判断定义为动态的。判断哪个字节码文件存在可以动态指定。

Condition – 小结

- 自定义条件:

- ① 定义条件类: 自定义类实现Condition接口, 重写 matches 方法, 在 matches 方法中进行逻辑判断, 返回 boolean值。 matches 方法两个参数:
 - context: 上下文对象, 可以获取属性值, 获取类加载器, 获取BeanFactory等。
 - metadata: 元数据对象, 用于获取注解属性。
- ② 判断条件: 在初始化Bean时, 使用 `@Conditional` (条件类.class) 注解

- SpringBoot 提供的常用条件注解:

- `ConditionalOnProperty`: 判断配置文件中是否有对应属性和值才初始化Bean
- `ConditionalOnClass`: 判断环境中是否有对应字节码文件才初始化Bean
- `ConditionalOnMissingBean`: 判断环境中没有对应Bean才初始化Bean

切换内置web服务器

SpringBoot的web环境中默认使用tomcat作为内置服务器，其实SpringBoot提供了4中内置服务器供我们选择，我们可以很方便的进行切换。

@Enable*注解

SpringBoot中提供了很多Enable开头的注解，这些注解都是用于动态启用某些功能的。而其底层原理是使用@Import注解导入一些配置类，实现Bean的动态加载。



思考

SpringBoot 工程是否可以直接获取jar包中定义的Bean?

@Import注解

@Enable*底层依赖于@Import注解导入一些类，使用@Import导入的类会被Spring加载到IOC容器中。而@Import提供4中用法：

- ① 导入Bean
- ② 导入配置类
- ③ 导入 ImportSelector 实现类。一般用于加载配置文件中的类
- ④ 导入 ImportBeanDefinitionRegistrar 实现类。

@EnableAutoConfiguration 注解

- @EnableAutoConfiguration 注解内部使用 `@Import (AutoConfigurationImportSelector.class)` 来加载配置类。
- 配置文件位置：META-INF/spring.factories，该配置文件中定义了大量的配置类，当 SpringBoot 应用启动时，会自动加载这些配置类，初始化Bean
- 并不是所有的Bean都会被初始化，在配置类中使用Condition来加载满足条件的Bean



案例：需求

自定义redis-starter。要求当导入redis坐标时，SpringBoot自动创建Jedis的Bean。



案例：实现步骤

- ① 创建 redis-spring-boot-autoconfigure 模块
- ② 创建 redis-spring-boot-starter 模块,依赖 redis-spring-boot-autoconfigure的模块
- ③ 在 redis-spring-boot-autoconfigure 模块中初始化 Jedis 的 Bean。并定义META-INF/spring.factories 文件
- ④ 在测试模块中引入自定义的 redis-starter 依赖，测试获取 Jedis 的Bean，操作 redis。



黑马程序员
www.itheima.com

传智播客旗下高端IT教育品牌

目录 Contents

- ◆ SpringBoot 自动配置
- ◆ SpringBoot 监听机制
- ◆ SpringBoot 启动流程分析

Java 监听机制

SpringBoot 的监听机制，其实是对Java提供的事件监听机制的封装。

Java中的事件监听机制定义了以下几个角色：

- ① 事件：Event，继承 java.util.EventObject 类的对象
- ② 事件源：Source，任意对象Object
- ③ 监听器：Listener，实现 java.util.EventListener 接口 的对象

SpringBoot 监听机制

SpringBoot 在项目启动时，会对几个监听器进行回调，我们可以实现这些监听器接口，在项目启动时完成一些操作。

ApplicationContextInitializer、SpringApplicationRunListener、CommandLineRunner、ApplicationRunner



目录 Contents

- ◆ SpringBoot 自动配置
- ◆ SpringBoot 监听机制
- ◆ SpringBoot 启动流程分析

SpringBoot 启动流程分析

启动流程

[SpringBoot启动流程.png](#)



传智播客旗下高端IT教育品牌