

## 1. springboot的自动配置原理

condition接口增加条件判断功能，用于选择性的创建Bean对象到spring容器中。

他有一个matches方法. 返回值是布尔类型, 如果容器中存在该类的全限定类名, 则返回true, 没有返回false, 如果加入起步依赖就会存在该类

类会在配置类中进行注册, 注册时会在方法上添加conditionoon相关注解, 判断是否存在该类, 存在就注入, 不存在不注入

ConditionalOnBean 当spring容器中有某一个bean时使用

ConditionalOnClass 当判断当前类路径下有某一个类时使用

ConditionalOnMissingBean 当spring容器中没有某一个bean时才使用

ConditionalOnMissingClass 当当前类路径下没有某一个类的时候才使用

ConditionalOnProperty 当配置文件中有某一个key value的时候才使用

## 2. springboot不只嵌tomcat

web容器有4种类型:

+ tomcat容器

+ jetty

+ netty

+ undertow

可以排除tomcat, 启用其他服务器

## 3. @Enable\*类型的注解说明

@SpringBootApplication注解里面有@EnableAutoConfiguration, 那么这种

@Enable\*开头就是springboot中定义的一些动态启用某些功能的注解, 他的底层实现原理实际上用的就是@import注解导入一些配置, 自动进行配置, 加载Bean.

## 4. @ComponentScan注解

在启动类的注解@SpringBootApplication注解里面又修饰了@ComponentScan注解, 该注解的作用用于组件扫描包类似于xml中的context-componet-scan, 如果不指定扫描路径, 那么就扫描该注解修饰的启动类所在的包以及子包。这就是为什么我们在第一天的时候写了controller 并没有扫描也能使用的原因。

## 5. @import注解

import注解用于导入其他的配置, 让spring容器进行加载和初始化。import的注解有以下几种方式使用:

- 直接导入Bean
- 导入配置类
- 导入ImportSelector的实现类, 通常用于加载配置文件中的Bean
- 导入ImportBeanDefinitionRegistrar实现类

## 6. @EnableAutoConfiguration

1. @import注解 导入配置
2. selectImports导入类中的方法中加载配置返回Bean定义的字符数组
3. 加载META-INF/spring.factories 中获取Bean定义的全路径名返回
4. 最终返回回去即可

## 7. 在springboot项目中，我们部署项目有两种方式：

- jar包直接通过java命令运行执行
- war包存储在tomcat等servlet容器中执行