Spring Boot的前世今生

Spring Framework AOP、IOC/DI

Spring 万能胶

- 如何对配置进行轻量化

思考： 用springmvc去构建一个web项目发布一个helloword的http端口

- 创建一个项目结构（maven/gradle）
- spring的依赖，spring mvc 、servlet api的依赖
- web.xml， DispatcherServlet
- 启动一个Spring mVC的配置，Dispatcher-servlet.xml
- **创建一个Controller 发布一个http请求**
- 发布到jsp/servlet容器

Spring boot的产生

- 2012年10月份，一个叫Mike Youngstrom(扬斯特罗姆)在Spring Jira中创建了一个功能请求，要求在Spring Framework中支持无容器Web应用程序体系结构，他谈到了在主容器引导 spring 容器内配置 Web 容器服务。
- Spring Boot刚出生的时候，引起了很多开源社区的关注，并且也有个人和企业开始尝试使用Spring Boot。 其实直到2016年，Spring Boot才真正在国内被使用起来。我之前在挖财的时候，2015年公司就开始采用Spring Boot来构建基于Dubbo的微服务架构。到现在，Spring Boot几乎是所有公司的第一选择。

到底什么是Spring Boot

> 约定优于配置理念下的一个产物

- 只要依赖的spring-boot-starter-web的jar，就会自动内置一个tomcat容器(替换)
- 项目结构
- 默认提供了配置文件application.properties
- starter启动依赖 - 如果是一个webstarter ，默认认为你是去构建一个spring mvc的应用.

如何Spring MVC 的web项目

约定优于配置

Spring Boot 如何应用，集成Mybatis

//TODO 详见代码

Spring Boot 和微服务

那为什么Spring Cloud会采用Spring Boot来作为基础框架呢？原因很简单

1. Spring Cloud它是关注服务治理领域的解决方案，而服务治理是依托于服务架构之上，所以它仍然需要一个承载框架
2. Spring Boot 可以简单认为它是一套快速配置Spring应用的脚手架，它可以快速开发单个微服务

所以spring cloud的版本和spring boot版本的兼容性有很大关联

Spring Boot的特性

- EnableAutoConfiguration 自动装配？
- Starter 启动依赖 依赖于自动装配的技术
- Actuator 监控，提供了一些endpoint， http、jmx形式去进行访问， health信息。 metrics 信息、。。。
- Spring Boot CLI（命令行操作的功能， groovy脚本） 客户端, groovy

Spring 注解驱动的发展过程

Spring Framework 的注解驱动的发展历史

spring 1.x

在SpringFramework1.x时代，其中在1.2.0是这个时代的分水岭，当时Java5刚刚发布，业界正兴起了使用Annotation的技术风，Spring Framework自然也提供了支持，比如当时已经支持了@Transactional等注解，但是这个时候，XML配置方式还是唯一选择。

```
<bean name="" class=""/>
```

Spring 2.x阶段

Spring Framework2.x时代，2.0版本在Annotation中添加了@Required、@Repository以及AOP相关的@Aspect等注解，同时也提升了XML配置能力，也就是可扩展的XML，比如Dubbo这样的开源框架就是基于Spring XML的扩展来完美的集成Spring，从而降低了Dubbo使用的门槛。

在2.x时代，2.5版本也是这个时代的分水岭， 它引入了一些很核心的Annotation

- Autowired 依赖注入
- @Qualifier 依赖查找
- @Component、@Service 组件声明
- @Controller、@RequestMappring等spring mvc的注解

尽管Spring 2.x时代提供了不少的注解，但是仍然没有脱离XML配置驱动，比如context:annotation-config context:componet-scan，前者的职责是注册Annotation处理器，后者是负责扫描classpath下指定包路径下被Spring模式注解标注的类，将他们注册成为Spring Bean

@Required/ @Repository（Dao）/@Aspect

spring 2.5

@Component (组件)

@Service service

@Controller (controller)

@RequetsMapping

spring 3.x版本

Spring Framework3.0是一个里程碑式的时代，他的功能特性开始出现了非常大的扩展，比如全面拥抱Java5、以及Spring Annotation。更重要的是，它提供了配置类注解@Configuration， 他出现的首要任务就是取代XML配置方式

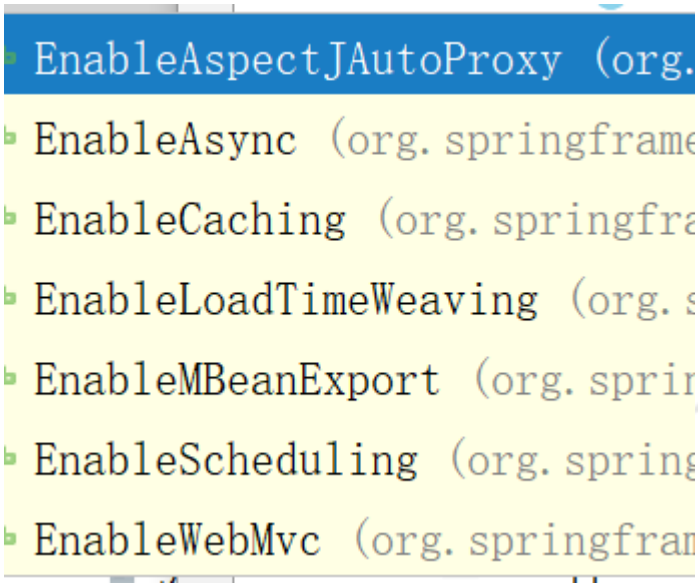
- @Configuration 去xml化

> 核心目的是：把bean对象如何更加便捷的方式去加载到Spring IOC容器中

- Component-Scan - @Service @Repository @Controller
- Import
-

Enable模块驱动

自动完成相关相关组件的bean的装配



Spring3.x版本中，集成Redis或者mybatis

- 创建一个配置类
- @bean注解来声明一个bean

```
@Bean
DefaultKaptcha defaultKaptcha(){

}
```

- @Enable启动一个模块，把相关组件的bean自动装配到IOC容器中

```
ScheduledAnnotationBeanPostProcessor
```

```java
@Role(BeanDefinition.ROLE_INFRASTRUCTURE)
public class SchedulingConfiguration {

    @Bean(name = TaskManagementConfigUtils.SCHEDULED_ANNOTATION_PROCESSOR_BEAN_NAME)
    @Role(BeanDefinition.ROLE_INFRASTRUCTURE)
    public ScheduledAnnotationBeanPostProcessor scheduledAnnotationProcessor() {
        return new ScheduledAnnotationBeanPostProcessor();
    }

}
```