**春天的杰作Spring源码分析**

主讲：Cat 老师

北京动力节点教育科技有限公司  
2020 • 北京

动力节点•版权所有•禁止传播

## 基于注解的Spring IOC容器宏观总结

1、创建AnnotationConfigApplicationContext;

2、创建AnnotationConfigApplicationContext的时候会先创建DefaultListableBeanFactory;

3、创建好DefaultListableBeanFactory之后，解析和扫描类上的注解并将所有Bean封装成BeanDefinition对象，把BeanDefinition放入到一个Map，该BeanDefinition对象封装了Spring所有bean的相关定义描述信息；

4、对Spring配置的所有bean做实例化，根据BeanDefinition对象的信息，底层采用反射创建具体的bean对象；

5、让给创建的bean进行属性赋值和依赖注入，可能还有循环依赖等情况；

6、将实例化的bean对象放入到一个Map中，后续应用程序就能从该Map中获取到bean对象，该Map也就是我们的IOC容器；

## 手写Spring IOC

当我们对spring ioc的核心流程了解后，参考它自己手写实现ioc，则就不是很难，在课程中我们手写了一份基于注解版本的spring ioc，大家可以参考，包括第一期也手写了一份基于xml版本的spring ioc，也可以参考；

手写spring ioc这个对实际工作没有很大作用，是一种技术修行，在潜移默化中提升编码能力，思维逻辑能力，手写框架的能力，主要是习得这种能力，并不需要一定写一遍；

比如：封装一个高效、轻量级、灵活好用的DAO框架，封装一套自己的权限管理框架，或者封装一个日常中Excel、PDF、Word的操作组件都可以作为一种技术修行；

## Spring MVC 源码分析

Tomcat是运行Java EE规范下的web程序的服务器，它实现了Java EE 规范，Java EE本身是提出了一系列的技术标准和规范（接口），但是没有具体是实现，具体的实现都是各个软件厂商实现的，在Java生态下，要实现web项目开发，切入点是JavaEE规范下的Servlet，Servlet是规范和接口，Servlet可以在Tomcat服务器上运行，像我们此次需要分析SpringMVC的源码，SpringMVC就是把JavaEE规范下的Servlet整合进来，这样就可以实现使用SpringMVC开发Java web项目，整个逻辑可以如下进行表述：

Oracle(原来sun公司) --> Web开发的Servlet规范 -->被SpringMVC整合 --> 部署到Tomcat中运行；

## 注解版SpringMVC如何启动

没有web.xml配置，SpringMVC如何通过继承于

AbstractAnnotationConfigDispatcherServletInitializer

的类来启动SpringMVC？

1. Tomcat在启动的过程中会查找所有实现

javax.servlet.ServletContainerInitializer接口的实现类；

2、javax.servlet.ServletContainerInitializer的实现类通过@HandlesTypes注解进行扩展；

3、Tomcat启动时会通过加载被@HandlesTypes注解的类，并读取@HandlesTypes注解的value值，将读取出来的值作为参数传给

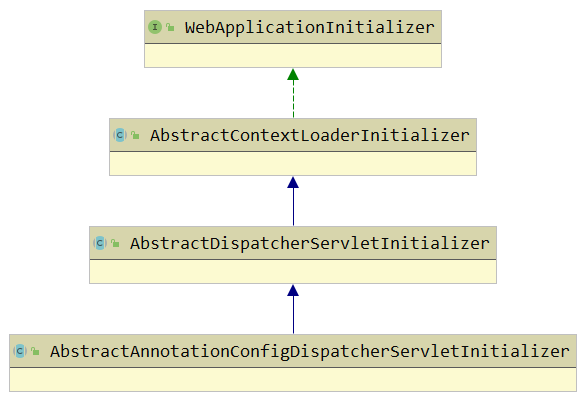
javax.servlet.ServletContainerInitializer实现类的onStartup ()方法；

4、Tomcat内部启动时调用 javax.servlet.ServletContainerInitializer的onStartup()方法，这个onStartup()方法的第一个参数就是@HandlesTypes注解的value 值指定的类的集合；

5、而在Spring MVC中，对javax.servlet.ServletContainerInitializer的实现就是org.springframework.web.SpringServletContainerInitializer, 而

org.springframework.web.SpringServletContainerInitializer类上面标注了@HandlesTypes注解，该注解指定的类就是org.springframework.web.WebApplicationInitializer接口，这样就将javax.servlet.ServletContainerInitializer和javax.servlet.WebApplicationInitializer关联起来了；

6、然后我们只需要在项目中写一个类实现org.springframework.web.WebApplicationInitializer接口，然后通过覆盖该接口里面的方法将我们的配置类告诉SpringMVC，即可完成对SpringMVC在Tomcat启动初始化过程中进行扩展，后续SpringMVC启动时就知道我们配置了什么信息，这就相当于把原来在web.xml里面做的工作移到了我们的配置类里面来完成，所以就可以废除web.xml文件；



org.springframework.web.WebApplicationInitializer接口