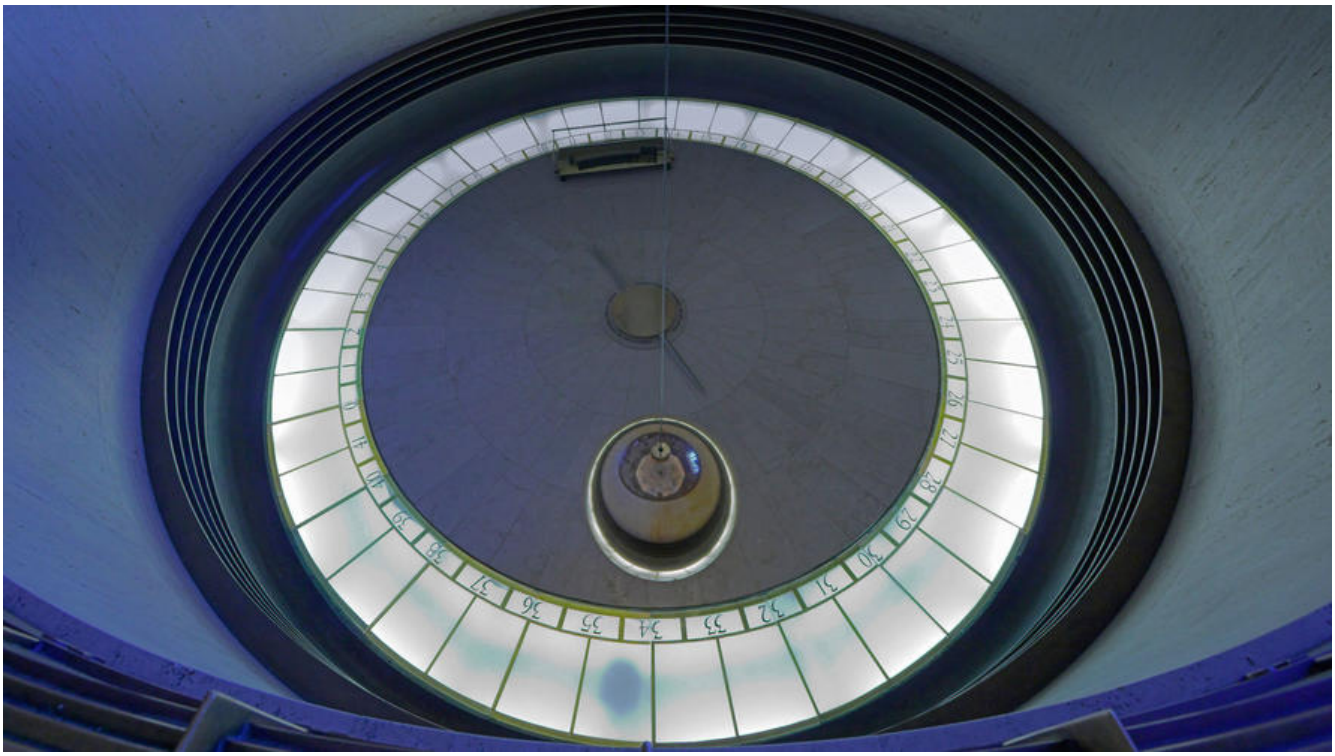
链滴

Spring IOC

作者：someone33881
原文链接：https://ld246.com/article/1542542768145
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)



    本文主要是记录在学习
pring IOC容器过程中的一些知识点总结！

关键词：**工厂方法模式、反射机制**

一、Spring

Spring是一个开源的、轻量级的J2EE开发框架，核心思想是IOC实现松耦合，利用AOP将应用的业逻辑与系统服务分离。

二、Spring IOC

实现原理：工厂模式 + 反射机制

** **把IOC容器的工作模式看做是工厂模式的升华，可以把IOC容器看作是一个工厂，这个工厂里要产的对象都在配置文件中给出定义，然后利用编程语言提供的反射机制，根据配置文件中给出的类名成相应的对象。**从实现来看，IOC是把以前在工厂方法里写死的对象生成代码，改变为由配置文件来义，也就是把工厂和对象生成这两者独立分隔开来，目的就是提高灵活性和可维护性。**

IOC控制反转：就是应用本身不负责所依赖对象的创建和维护，而是由外部容器对依赖对象进行创和维护，这种对依赖对象的控制权转移到了外部容器，称之为控制反转；**（->也就是说一个对象依赖其他对象会通过被动的方式传递进来，而不是这个对象自己创建或查找依赖对象；；可以理解IOC与JII相反，不是对象从容器中查找依赖，而是容器在对象初始化时不等到对象请求时就主动将依赖传递它）**

** DI依赖注入：**与IOC相应地会出现一个依赖注入的概念，指在应用程序运行期间，由外部容器动地将依赖对象注入到组件中，一般通过**构造函数或setter传递或接口传递**

Spring IOC容器原理：就是spring在启动时，会读取应用程序所提供的Bean配置信息，并在Sping容器中对应生成一份Bean配置注册表，然后根据该注册表实例化Bean，和装配好Bean之间的依关系；然后将Bean实例放到Spring容器的Bean缓存池（HashMap实现）中，最终支撑上层应用的行！

原文链接：Spring IOC

简言之，就是Spring在启动的时候，通过一个配置文件描述Bean和Bean之间的依赖关系，并生成相的配置注册表，然后利用java语言的反射机制实例化Bean及建立Bean之间的依赖关系

反射是Java语言的一个特性，它允许程序**在运行时（注意不是编译的时候）来进行自我检查并且对部的成员进行操作**

IOC容器初始化过程

** 定位Resource资源**：由ResourceLoader通过统一的Resource接口完成对Resource外部资源定位，也即BeanDefinition的资源定位

载入BeanDefinition：由BeanDefinitionReader读取、解析定位的Resource资源，也即将用户义好的Bean表示成IOC容器的内部数据结构BeanDefinition

注册：通过BeanDefinitionRegistry接口，向IOC容器注册上一步中载入的BeanDefinition（其实一个HashMap进行管理的），IOC容器通过HashMap数据结构对BeanDefinition进行维护管理

依赖注入：是发生在在应用第一次调用getBean（**BeanFactory接口的方法**）时向容器获取Bea时进行依赖注入的，若某个Bean设置lazyinit属性，则该Bean在容器初始化时就会被依赖注入

Spring容器 - Spring容器分为两类

BeanFactory：**最常用的BeanFactory实现就是XmlBeanFactory类，它根据XML文件中的定义载beans，该容器从XML文件读取配置元数据并用它去创建一个完全配置的系统或应用，**BeanFacto y是Spring框架的基础设施面向Spring本身；

ApplicationContext应用上下文：基于BeanFactory之上构建，并提供面向应用（Spring框架的开者）的服务

关系：ApplicationContext由BeanFactory派生，通过该类很多功能可以直接通过配置方式实现（在BeanFactory中则需要以编程方式实现）

SpringIOC容器中重要的实现类

BeanDefinition：Spring配置文件中每一个节点元素
对应容器中国的一个BeanDefinition对象，描述Bean的配置信息；

BeanDefinitionRegistry：该接口提供向容器手动注册BeanDefinit on对象的方法；

ListableBeanFactory：该接口定义了访问容器中Bean基本信息
若干方法，如Bean个数、某一类型Bean的配置名、是否包含某一Bean等；

HierarchicalBeanFactory：该接口提供子容器可访问父容器中Bean
功能，也即通过该接口SpringIOC容器可以建立父子级联的容器体系（如SpringMVC中表现层Bean于一个子容器中，业务层和持久层位于父容器中，因此表现层Bean可以引用业务层和持久层的Bean而业务层和持久层不能使用表现层Bean）；

ConfigurableBeanFactory：该接口用于增强Ioc容器的可定制性
提供设置类装载器、属性编辑器、容器初始化后置处理器等方法；

AutowireCapableBeanFactory：该接口定义了可将容器中Bean按照某种规则（如名字匹配、型匹配）进行自动装配 的方法；

SingletonBeanRegistry：该接口提供运行期间向容器注册单例实例B an的方法；

三、SpringMVC - WebApplicationContext

** WebApplicationContext是专门为Web应用准备的（必须在拥有web容器的前提下才能完成启工作），允许从相对于Web根目录的路径中装载配置文件完成初始化工作。**

启动过程：

web容器如tomcat -> web.xml（配置自启动的Servlet或定义Web容器监听器） -> Servlet如ContextLoaderServlet，也即在tomcat中创建一个ServletContext实例（用于启动WebApplicationContext） -> WebApplicationContext中引用ServletContext并将整个应用的Spring上下对象WebApplicationContext作为属性（以ROOT_WEB_
PPLICATION_CONTEXT_ATTRIBUTE为键）放
ServletContext中 -> 因而web容器可以通过ServletContext实例获取到Spring上下文对象

Spring中分别提供了用于启动WebApplicationContext的Servlet或web容器监听器：

org.springframework.web.context.ContextLoaderServlet

org.springframework.web.context.ContextLoaderListener

参考：

1、SpringIOC原理总结