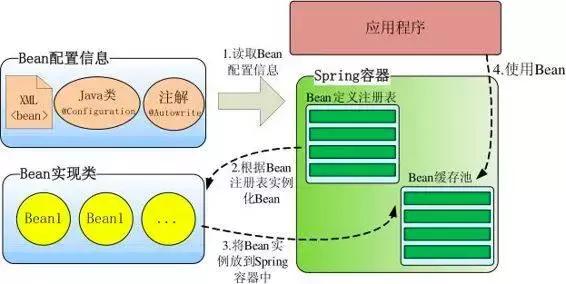
Spring IoC 自定义Bean

Spring IoC是轻量级IoC容器，创建容器核心是通过让容器感知创建对象与对象(Bean)的关系，其中Bean是Spring框架的核心概念，Spring容器可以理解为工厂，而Bean是该工厂的产品。Spring容器根据各种形式的Bean配置信息在容器内部建立Bean定义注册表，然后根据注册表加载、实例化Bean并建立Bean之间的依赖关系，最后将其放入Bean缓存池中，以供外层应用程序的调用，如下图所示：



Spring提供了三种Bean定义方法：

* 基于XML的bean定义，需要提供Setter方法
* 基于注解的Bean定义，不需要提供Setter方法，与XML定义一样
* 基于Java类的Bean定义，需要提供Setter方法

这几种方式实现的效果是相同的，使用XML或者基于Java类对大量Bean进行配置比较复杂，目前常用基于注解的Bean定义方法，可以使程序变得简洁。

# 基于XML的bean定义

* 定义Bean POJO类

|  |  |
| --- | --- |
| *@Setter*  *@Getter*  *public class Teacher {*  *private String name;*  *}* | *@Setter*  *@Getter*  *public class Student {*  *private String name;*  *private Teacher teacher;*  *}* |

* 在XML配置application-context.xml中定义类的属性值

*<?xml version="1.0" encoding="UTF-8"?>*

*<beans xmlns="http://www.springframework.org/schema/beans"*

*xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"*

*xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">*

*<bean id="student" class="com.fys.spring.basic.Student">*

*<property name="name" value="tom"/>*

*<property name="teacher" ref="teacher"/>*

*</bean>*

*<bean id="teacher" class="com.fys.spring.basic.Teacher">*

*<property name="name" value="Jerry"/>*

*</bean>*

*</beans>*

* 定义测试类

*import org.springframework.context.support.FileSystemXmlApplicationContext;*

*public class TestXMLDI {*

*public static void main(String[] args) {*

*FileSystemXmlApplicationContext context =*

*new FileSystemXmlApplicationContext(application-context.xml");*

*Student student = (Student) context.getBean("student");*

*Teacher teacher = (Teacher) context.getBean("teacher");*

*System.out.println("Stuent Name: " + student.getName() +*

*" , teacher is "+ student.getTeacher().getName());*

*System.out.println("Teacher Name:" + teacher.getName());*

*}*

*}*

执行结果如下：

*Stuent Name:tom , teacher is Jerry*

*Teacher Name:Jerry*

# 基于注解的Bean定义

Spring提供4个注解，其作用与XML中bean定义效果一致，在于将组件交给Spring容器管理，组件的名称默认为类名，也可以自己修改：

* @Component，当对组件的层次难以定位的时候使用这个注解
* @Controller，表示控制层的组件
* @Service，表示业务逻辑层的组件
* @Repository，表示数据访问层的组件

1. 定义Bean类

|  |  |
| --- | --- |
| *@Getter*  *@Controller("teacher")*  *public class Teacher {*  *@Value("Jerry")*  *private String name;*  *}* | *@Getter*  *@Component("student")*  *public class Student {*  *@Value("Tom")*  *private String name;*  *@Resource*  *private Teacher teacher;*  *}* |

1. 定义application配置文件

*<?xml version="1.0" encoding="UTF-8"?>*

*<beans xmlns="http://www.springframework.org/schema/beans"*

*xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"*

*xmlns:context="http://www.springframework.org/schema/context"*

*xsi:schemaLocation="http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context.xsd">*

*//定义component Scan的包*

*<context:component-scan base-package="com.fys.spring.annotation"/>*

*</beans>*

测试类与上节测试相同

# 基于Java类的bean定义

定义Bean Configuration

*@Configuration*

*public class BeanConfiguration {*

*@Bean*

*public Student student() {*

*Student student = new Student();*

*student.setName("Tom");*

*student.setTeacher(teacher());*

*return student;*

*}*

*@Bean*

*public Teacher teacher() {*

*Teacher teacher = new Teacher();*

*teacher.setName("Jerry");*

*return teacher;*

*}}*

测试类定义如下：

*public static void main(String[] args) {*

*AnnotationConfigApplicationContext context =*

*new AnnotationConfigApplicationContext(BeanConfiguration.class);*

*Student student = (Student) context.getBean("student");*

*Teacher teacher = (Teacher) context.getBean("teacher");*

*System.out.println("Stuent Name: " + student.getName() +*

*" : teacher is "+ student.getTeacher().getName());*

*System.out.println("Teacher Name:" + teacher.getName());*

*System.out.println("Student#Teacher HashCode:" + student.getTeacher().hashCode());*

*System.out.println("Teacher Hashcode:"+teacher.hashCode());*

*}*

执行结果如下：

*Stuent Name: Tom : teacher is Jerry*

*Teacher Name:Jerry*

*Student#Teacher HashCode: 943081537*

*Teacher Hashcode: 943081537*

**参考链接：**

http://www.cnblogs.com/flowwind/p/4772375.html

https://blog.csdn.net/sinat\_34596644/article/details/53080026

https://www.cnblogs.com/wuchanming/p/5426746.html