Atitit 提升速度 手动注入bean spring

# **Spring 手动注册bean**



[灵颖桥人](https://blog.csdn.net/qq_22076345" \t "https://blog.csdn.net/qq_22076345/article/details/_blank) 2020-05-02 22:31:14 IMG_257 2045 IMG_258 收藏 2 IMG_259 原力计划

分类专栏： [Spring Framework](https://blog.csdn.net/qq_22076345/category_7775777.html" \t "https://blog.csdn.net/qq_22076345/article/details/_blank) 文章标签： [bean](https://www.csdn.net/tags/MtTaEg0sNDExOTktYmxvZwO0O0OO0O0O.html" \t "https://blog.csdn.net/qq_22076345/article/details/_blank) [spring](https://www.csdn.net/tags/MtTaEg0sMDg2NTAtYmxvZwO0O0OO0O0O.html" \t "https://blog.csdn.net/qq_22076345/article/details/_blank)

版权

       一般情况下，我们Spring应用中的bean都是通过注解或者xml注入到容器中的，有些情况下我们可能想手动往容器中注入bean，即编程方式注入bean。

      本文所使用源码包版本：spring-beans-5.0.5.RELEASE.

## **如何注册？**

      Spring 中用BeanDefinition接口描述一个bean，Spring容器中用Map<String, BeanDefinition> beanDefinitionMap

存储beanName和BeanDefinition对象的映射关系【beanDefinitionMap 可参考DefaultListableBeanFactory】。Spring在实例化一个bean，都是先从 beanDefinitionMap 中获取beanDefinition对象，进而构造出对应的bean。因此，我们手动注册bean的问题，就演化为如何往这个 beanDefinitionMap 放入我们要注册bean对应的 BeanDefinition 对象。

      Spring 提供了 BeanDefinitionRegistry 接口来操作底层beanFactory实现的beanDefinitionMap。

****【2020-08-17】备注：推荐通过实现 BeanDefinitionRegistryPostProcessor 接口操作 BeanDefinitionRegistry：****

package com.wz.util;

import org.springframework.beans.factory.support.DefaultListableBeanFactory;

import org.springframework.beans.factory.support.GenericBeanDefinition;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import com.wz.Bean1;

public class SprUti {

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

AnnotationConfigApplicationContext applicationContext = new AnnotationConfigApplicationContext();

DefaultListableBeanFactory beanFactory = (DefaultListableBeanFactory) applicationContext.getBeanFactory();

GenericBeanDefinition beanDefinition = new GenericBeanDefinition();

beanDefinition.setBeanClass(Bean1.class);

beanFactory.registerBeanDefinition("testxKaijyoTime开奖时间2idid", beanDefinition);

applicationContext.refresh();

Bean1 h = (Bean1) applicationContext.getBean(Bean1.class);

h.testBean();

}

}