Bean在Spring和SpringMVC中无所不在，将这个概念内化很重要，下面分享一下我的想法：

一、Bean是啥

1、Java面向对象，对象有方法和属性，那么就需要**对象实例来调用方法和属性（即实例化）；**

2、凡是有方法或属性的类都需要实例化，这样才能具象化去使用这些方法和属性；

3、规律：**凡是子类及带有方法或属性的类都要加上注册Bean到Spring IoC的注解；（**@Component , @Repository , @ Controller , @Service , @Configration**）**

4、把Bean理解为类的代理或代言人（实际上确实是通过反射、代理来实现的），这样它就能代表类拥有该拥有的东西了

5、我们都在微博上@过某某，对方会优先看到这条信息，并给你反馈，那么在Spring中，**你标识一个@符号，那么Spring就会来看看，并且从这里拿到一个Bean（注册）或者给出一个Bean（使用）**

二、注解分为两类：

1、一类是使用Bean，即是把已经在xml文件中配置好的Bean拿来用，完成属性、方法的组装；比如@Autowired , @Resource，可以通过byTYPE（@Autowired）、byNAME（@Resource）的方式获取Bean；

2、一类是注册Bean,@Component , @Repository , @ Controller , @Service , @Configration这些注解都是把你要实例化的对象转化成一个Bean，放在IoC容器中，等你要用的时候，它会和上面的@Autowired , @Resource配合到一起，把对象、属性、方法完美组装。

三、@Bean是啥？

1、原理是什么？先看下源码中的部分内容：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | Indicates that a method produces a bean to be managed by the Spring container.     <h3>Overview</h3>     <p>The names and semantics of the attributes to this annotation are intentionally   similar to those of the {@code <bean/>} element in the Spring XML schema. For   example:     <pre class="code">       @Bean       public MyBean myBean() {           // instantiate and configure MyBean obj           return obj;      }</pre> |

　　意思是@Bean明确地指示了一种方法，什么方法呢——**产生一个bean的方法，并且交给Spring容器管理；**从这我们就明白了为啥@Bean是放在方法的注释上了，因为它很明确地告诉被注释的方法，你给我产生一个Bean，然后交给Spring容器，剩下的你就别管了。

 2、记住，@Bean就放在方法上，就是产生一个Bean，那你是不是又糊涂了，因为已经在你定义的类上加了@Configration等注册Bean的注解了，为啥还要用@Bean呢？这个我也不知道，下面我给个例子，一起探讨一下吧：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  3 | package com.edu.fruit;    //定义一个接口      public interface Fruit<T>{          //没有方法  }    /\*  \*定义两个子类  \*/  package com.edu.fruit;       @Configuration       public class Apple implements Fruit<Integer>{//将Apple类约束为Integer类型    }    package com.edu.fruit;       @Configuration       public class GinSeng implements Fruit<String>{//将GinSeng 类约束为String类型    }  /\*  \*业务逻辑类  \*/  package com.edu.service;         @Configuration         public class FruitService {            @Autowired            private Apple apple;            @Autowired            private GinSeng ginseng;      //定义一个产生Bean的方法         @Bean(name="getApple")         public Fruit<?> getApple(){         System.out.println(apple.getClass().getName().hashCode);           System.out.println(ginseng.getClass().getName().hashCode);         return new Apple();  }  }  /\*  \*测试类  \*/  @RunWith(BlockJUnit4ClassRunner.class)  public class Config {      public Config(){          super("classpath:spring-fruit.xml");      }      @Test      public void test(){          super.getBean("getApple");//这个Bean从哪来，从上面的@Bean下面的方法中来，返回                                                            的是一个Apple类实例对象        }  } |

从上面的例子也印证了我上面的总结的内容：

**1、凡是子类及带属性、方法的类都注册Bean到Spring中，交给它管理；**

**2、@Bean 用在方法上，告诉Spring容器，你可以从下面这个方法中拿到一个Bean**