# 依赖注入方式

## 情况一

问题：尽量使用构造器/setter注入@RequiredArgsConstructor而不推荐使用字段注入@Resoucre/@Autowired的原因。



### 解决方案：

spring推荐的写法：

1. 构造器注入

@Slf4j

@RestController

@RequestMapping("point")

public class PointController {

private final PointBiz1 pointBiz1;

private final PointBiz2 pointBiz1;

@Autowired

public PointController(PointBiz1 pointBiz1,PointBiz2 pointBiz2) {

this.pointBiz1 = pointBiz1;

this.pointBiz2 = pointBiz2;

}

}

这里所有字段必须都是final,若不使用final,用@NotNull注解也是可以的，不过该方法有几率导致Spring循环引用问题

2）setter注入

@Slf4j

@RestController

@RequestMapping("point")

public class PointController {

private final PointBiz1 pointBiz1;

private final PointBiz2 pointBiz1;

@Autowired

public void setPointBiz1(PointBiz1 pointBiz1) {

this.pointBiz1 = pointBiz1;

}

@Autowired

public void setPointBiz2(PointBiz2 pointBiz2) {

this.pointBiz2 = pointBiz2;

}

}

### 构造器注入和Setter注入的区别：

构造器对注入强制性的的依赖是好的。对象需要这些依赖才能正常运转，通过构造器提供这些依赖就能保证对象初始化后就能被使用。使用构造器注入的一个可能的影响就是循环依赖。

Setter应该被用来注入可变的依赖。当没有提供依赖时，这个类也应该能够运转。当实例化对象后，这些依赖也能随时改变。其实也视情况而变,有时，一个不变的对象是理想状态。有时，最好是能在运行期间改变对象的属性。

### 结论：

Field注入应该尽可能地去避免使用。作为替代，你应该使用构构造器注入或Setter注入。他们都有利有弊，需要视情况而定。当然你可以在同一个类中使用这两种方法。构造器注入更适合强制性的注入旨在不变性，Setter注入更适合可变性的注入。

****Autowired的多种方式：****1.构造器注解(constructor)，2.属性setter注解，3.field注解。

不管使用上面3中的哪个方法，Spring都会满足声明的依赖。假如有且只有一个bean匹配依赖的话，那么这个bean将会被装配进来。如果使用2，3方式注解，有多个bean的话，则用Qualifier指定。

public class UserService {

@Autowired

@Qualifier(name="userDao1")

private UserDao userDao;

}

@Autowired(required = false)

private IMediaPlayer CDPlayer;

required=false表示如果没有匹配的话，Spring会让bean处于未装配的样子。使用未装配的属性，会出现NullPointerException

补充下@Resource注解，如果同时指定了name和type，则从Spring上下文中找到唯一匹配的bean进行装配，找不到则抛出