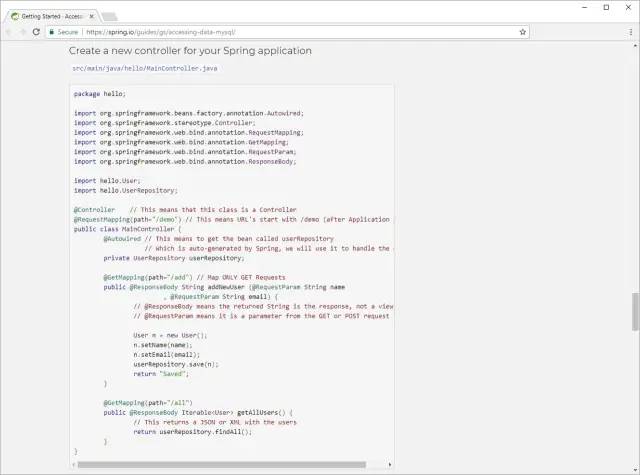
https://mp.weixin.qq.com/s/BOGK5J2Vl3w-3FDec9zHsA

**@Autowired 和 @Resource的区别**





****

### 基于字段的依赖注入缺点

**对于有final修饰的变量不好使**

Spring的IOC对待属性的注入使用的是set形式，但是final类型的变量在调用class的构造函数的这个过程当中就得初始化完成，这个是**基于字段的依赖注入**做不到的地方．只能使用**基于构造函数的依赖注入**的方式

**掩盖单一职责的设计思想**

我们都知道在OOP的设计当中有一个**单一职责思想**，如果你采用的是**基于构造函数的依赖注入**的方式来使用Spring的IOC的时候，当你注入的太多的时候，这个构造方法的参数就会很庞大，类似于下面.

当你看到这个类的构造方法那么多参数的时候，你自然而然的会想一下:**这个类是不是违反了单一职责思想?**.但是使用**基于字段的依赖注入**不会让你察觉，你会很沉浸在@Autowire当中

**与Spring的IOC机制紧密耦合**

当你使用**基于字段的依赖注入**方式的时候，确实可以省略构造方法和setter这些个模板类型的方法，但是，你把控制权全给Spring的IOC了，别的类想重新设置下你的某个注入属性，没法处理(当然反射可以做到).

本身Spring的目的就是解藕和依赖反转，结果通过再次与类注入器（在本例中为Spring）耦合，失去了通过自动装配类字段而实现的对类的解耦，从而使类在Spring容器之外无效.

**隐藏依赖性**

当你使用Spring的IOC的时候，被注入的类应当使用一些public类型(构造方法，和setter类型方法)的方法来向外界表达:我需要什么依赖.但是**基于字段的依赖注入**的方式，基本都是private形式的，private把属性都给封印到class当中了.

**无法对注入的属性进行安检**

**基于字段的依赖注入**方式，你在程序启动的时候无法拿到这个类，只有在真正的业务使用的时候才会拿到，一般情况下，这个注入的都是非null的，万一要是null怎么办，在业务处理的时候错误才爆出来，时间有点晚了，如果在启动的时候就暴露出来，那么bug就可以很快得到修复(当然你可以加注解校验).

如果你想在属性注入的时候，想根据这个注入的对象操作点东西，你无法办到．我碰到过的例子：一些配置信息啊，有些人总是会配错误，等到了自己测试业务阶段才知道配错了，例如线程初始个数不小心配置成了3000，机器真的是狂叫啊!这个时候就需要再某些Value注入的时候做一个检测机制.

### 结论

通过上面，我们可以看到，**基于字段的依赖注入**方式有很多缺点,我们应当**避免使用基于字段的依赖注入**.推荐的方法是使用**基于构造函数**和**基于setter的依赖注入**.对于必需的依赖项，建议使用**基于构造函数**的注入，以使它们成为不可变的，并防止它们为null。对于可选的依赖项，建议使用**基于Setter的注入。**

　　@Resource的作用相当于@Autowired，只不过**@Autowired按byType自动注入，而@Resource默认按 byName自动注入**罢了。@Resource有两个属性是比较重要的，分是name和type，Spring将@Resource注解的name属性解析为bean的名字，而type属性则解析为bean的类型。所以如果使用name属性，则使用byName的自动注入策略，而使用type属性时则使用byType自动注入策略。如果既不指定name也不指定type属性，这时将通过反射机制使用byName自动注入策略。  
　　@Resource装配顺序  
　　1. 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，找不到则抛出异常  
　　2. 如果指定了name，则从上下文中查找名称（id）匹配的bean进行装配，找不到则抛出异常  
　　3. 如果指定了type，则从上下文中找到类型匹配的唯一bean进行装配，找不到或者找到多个，都会抛出异常  
　　4. 如果既没有指定name，又没有指定type，则自动按照byName方式进行装配；如果没有匹配，则回退为一个原始类型进行匹配，如果匹配则自动装配；

## @Autowired 与@Resource的区别：

1、 @Autowired与@Resource都可以用来装配bean. 都可以写在字段上,或写在setter方法上。

2、 @Autowired默认按类型装配（这个注解是属业spring的），默认情况下必须要求依赖对象必须存在，如果要允许null值，可以设置它的required属性为false，如：@Autowired(required=false) ，如果我们想使用名称装配可以结合@Qualifier注解进行使用，如下：

@Autowired () @Qualifier ( "baseDao" )

private BaseDao baseDao;

3、@Resource（这个注解属于J2EE的），默认按照名称进行装配，名称可以通过name属性进行指定，如果没有指定name属性，当注解写在字段上时，默认取字段名进行安装名称查找，如果注解写在setter方法上默认取属性名进行装配。当找不到与名称匹配的bean时才按照类型进行装配。但是需要注意的是，如果name属性一旦指定，就只会按照名称进行装配。

@Resource (name= "baseDao" )

private BaseDao baseDao;