

# 何时不在 Spring / Spring Boot 中使用 @Autowire

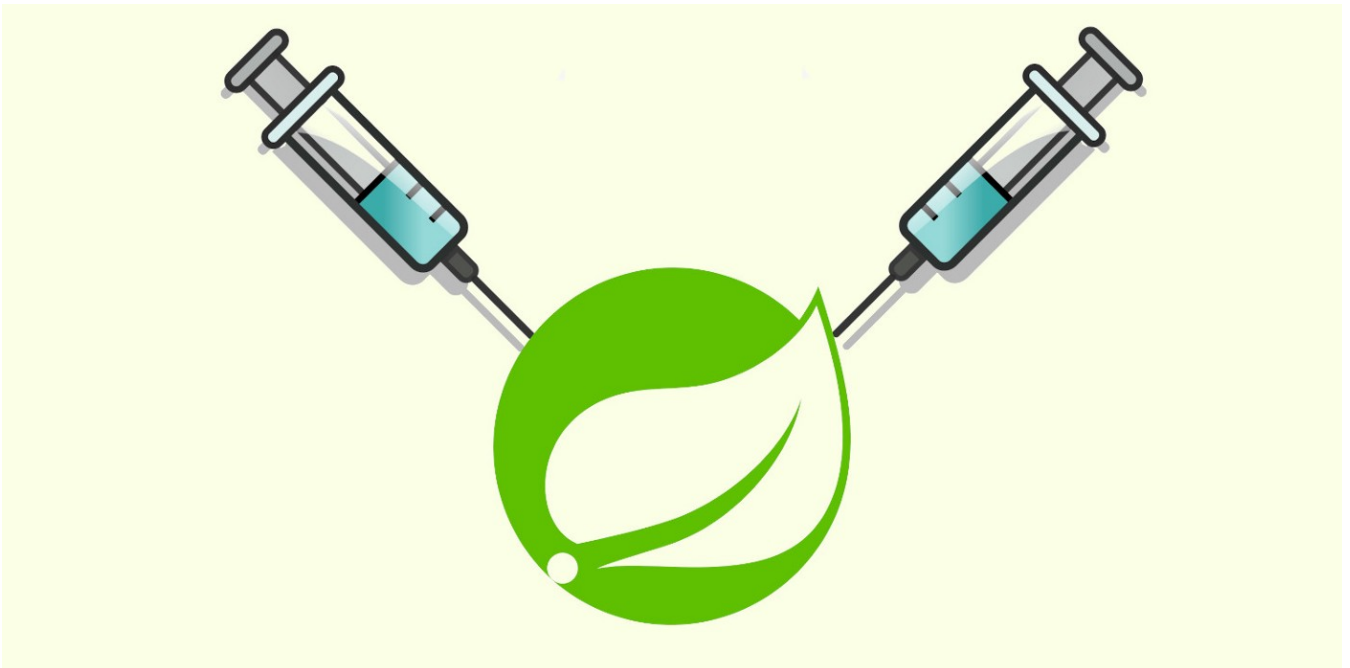
依赖注入很酷，但很多人误解了它的目的，最重要的是不了解如何正确使用它。



穆罕默德·阿蒂夫

跟随

5月2日 · 3分钟阅读



高级开发人员：为什么使用字段注入而不是构造函数注入？

初级开发人员：什么是字段注入？我正在使用@Autowired

一个经常发生的简单对话。听上去微不足道，却蕴含着深刻的含义。缺乏对事物运作的原因和方式的了解会导致灾难性的代码。

在我们讨论何时以及为什么使用哪种依赖注入之前，让我们深入研究不同类型的依赖注入

## 依赖注入的类型

1. 现场注入
2. 构造函数注入
3. 二传手注射

## 现场注入

简单地说，声明一个变量，希望是私有的，然后用 `@Autowired`

```
@Autowired
私有 UserService userService;
```

## 构造函数注入

简单来说就是让 **Spring Container** 直接通过构造函数注入依赖

```
@RestController
public class UserController {
    private final UserService userService;
    public UserController(UserService userService) {
        this.userService = userService;
    }
}
```

## Setter 注入 — 很少使用

它类似于字段注入，但这里的 **setter** 是用 `@Autowired` 标签注释的。

私人用户服务用户服务；

```
@Autowired
public void setUserService(UserService userService) {
    this.userService = userService;
}
```

## 它们有何不同？

保持简短，

- 字段注入使用反射来设置私有变量的值
- 构造函数注入发生在创建对象本身时

- Setter Injection 使用 setter 来设置值

现在您已经掌握了依赖注入的要点，您可以在本博客之外探索更多相关信息。

没有悬念了。直接跳到重点.....

[Download CSV](#)[View larger version](#)

## 判决书

构造函数注入始终是依赖注入的首选，因为它具有可靠和严格的性质。

Field Injection 也可以用于构造函数注入绝对不可能的场景（即使作为变通方法也必须避免循环依赖）。

## 其他注意事项

在谈论字段注入与构造函数注入时广泛讨论的一个关键点是**必需和可选依赖项**

- 许多争论我们可以对可选依赖使用字段注入，对必需依赖使用构造函数注入，但是在大多数情况下，具有可选依赖本身似乎是一个糟糕的设计。
- 由于 optional 依赖的范围是一个 *Unrestricted Territory*，任何开发者都可以通过调用它来随意添加多个依赖，最终降低整体代码的质量。

- 就我个人而言，我觉得如果您的代码中需要依赖项，那么这意味着它是必需的，如果它是可选的，那么为什么还要包含它。这种说法是有争议的，可能因人而异。

让我知道您如何在代码中使用依赖注入以及它如何提高项目的整体质量

在 ZeMoSo 技术中，我们遵循最佳设计和编码原则，确保每次都能获得最高质量的应用程序。要了解有关我们工程流程的更多信息，[请关注我们](#)

<https://medium.com/engineering-zemoso>

依赖注入   春天   弹簧靴   干净的代码   扎实的原则

[关于](#)   [写](#)   [帮助](#)   [合法的](#)

获取 Medium 应用

