* Việc liên kết giữa các Bean có thể được cấu hình bằng
  + File xml
  + Dùng annotation
  + Sử dụng code Java
* **Có 5 loại autowired**
  + By Type (default)
    - By Field
    - By Setter
    - By Constructor
  + By Name
  + By AutoDetect

Thứ tự thực hiện: By Contructor, nếu không thỏa mãn sẽ thực hiện tiếp By Type, …

* **Setup:**

**Enable @Autowired Annotations**

* + Use AnnotationConfigApplicationContext



* + Use XML

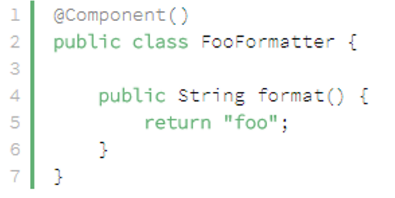
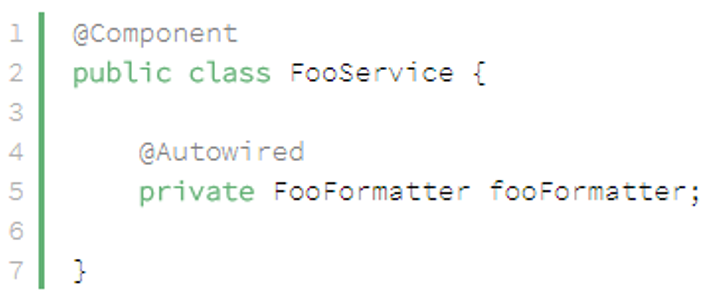


1. **@Autowired ByType (Datatype)**

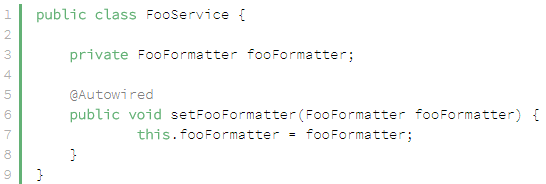
Default @Autowire là kiểu ByType. Nếu có nhiều hơn 1 Type được tìm thấy thì Spring sẽ throw ra 1 exception

* 1. **@Autowired on Field (Datatype)**

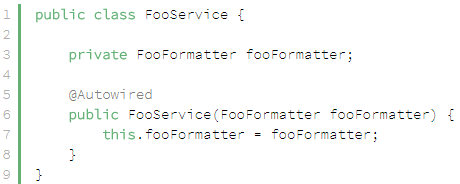
Autowired trực tiếp trên properties 🡺 không cần getter, setter

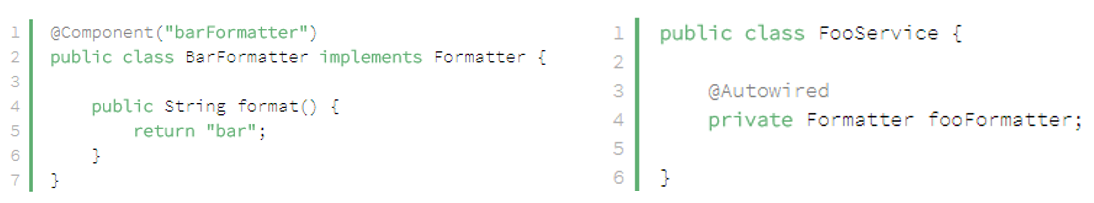
* 1. **@Autowired on Setters**



* 1. **@Autowired on Constructors**



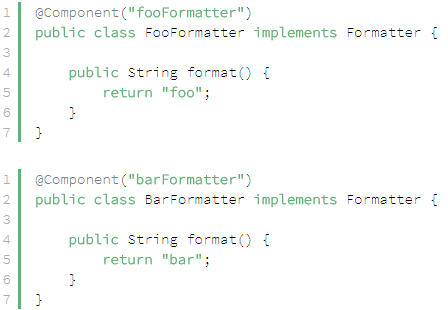
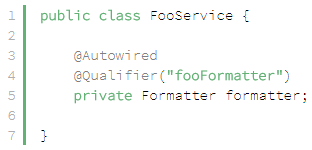
1. **@Autowrited ByName**



* + **@Qualifier**

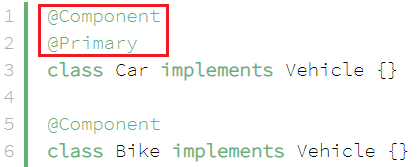
Nếu có nhiều hơn 1 Type được tìm thấy thì Spring sẽ throw ra 1 exception

Để tránh điều này thì cần phải phân biệt các Type bằng name

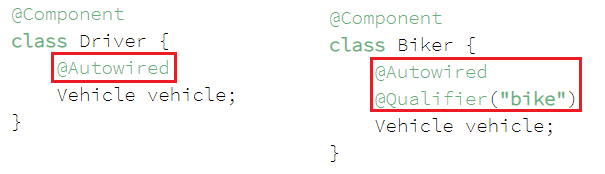
**** 

* + **@Primarykey**

Nếu có nhiều Type được tìm thấy, mà user không config rõ muốn autowired type name nào thì default sẽ lấy Type có field @Primary

****

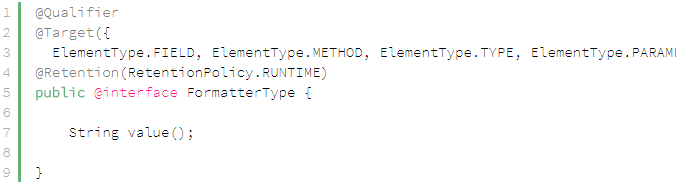
*Components*

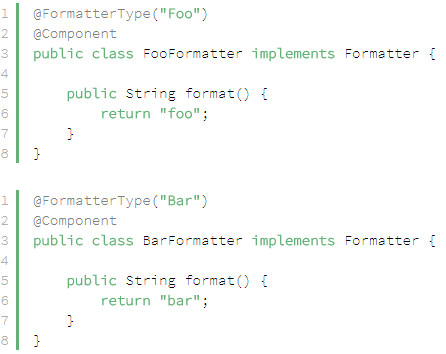
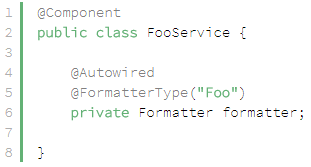
****

*Using*

* + **Custom Qualifier**

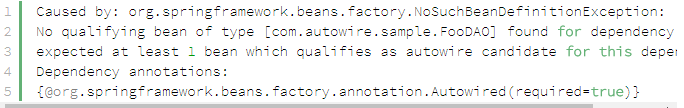
The value specified in the @Target annotation restrict where the qualifier can be used to mark injection points.



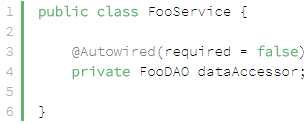
* **@Autowired(required = false)**

Nếu như @Autowired không tìm thấy Object thì sẽ bắn ra lỗi



Để tránh lỗi thì sử dụng thêm required = false

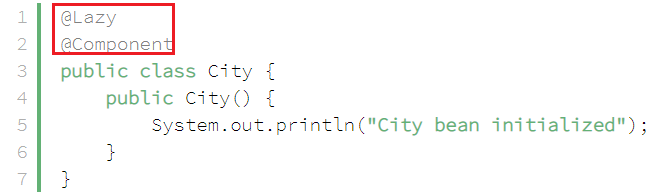
Nếu không tìm thấy Object thì Spring sẽ bỏ qua attribute đó



* **@Lazy**

initialize our bean lazily

VD:

[*https://www.baeldung.com/spring-lazy-annotation*](https://www.baeldung.com/spring-lazy-annotation)

* **Reference**

[http://www.baeldung.com/spring-autowire](http://www.baeldung.com/spring-autowirehttp://www.baeldung.com/spring-autowire)