Spring @Autowired y la inyección de dependencias

Gustavo García, 30 de mayo de 2022.

Esta explicación está basada en:

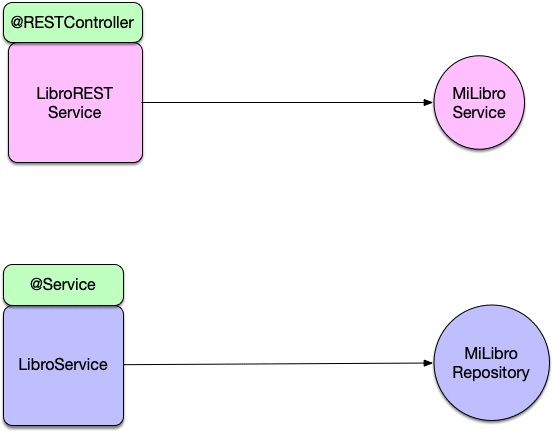
<https://www.arquitecturajava.com/spring-autowired-y-la-inyeccion-de-dependencias/>

pero la edité a mi gusto.

Spring @Autowired es una de las anotaciones más habituales cuando trabajamos con Spring Framework ya que se trata de la anotación que permite inyectar unas dependencias con otras dentro de Spring .

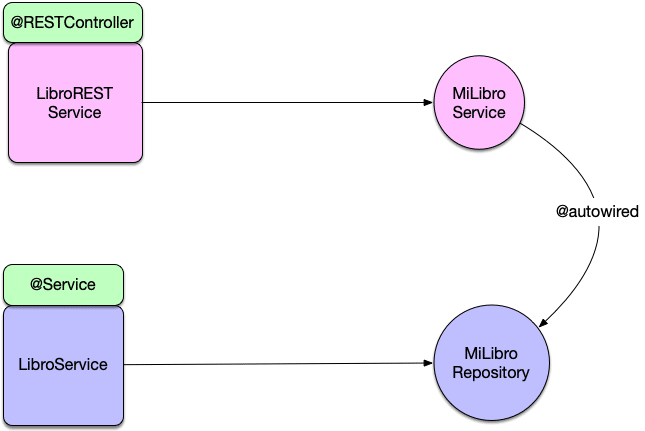
# Spring @Autowired y su funcionamiento

@Autowired sirve para inyectar en una clase destino una propiedad perteneciente a otra clase origen. En este sentido, Spring funciona como una factoría[[1]](#footnote-1) de objetos, es decir que implementa el patrón factoría.

Una clase se registra como origen de inyección con alguna de estas anotaciones:

* @Controller,
* @Service,
* @Repository, o
* @RestController.

En la clase origen tenemos la propiedad que deseamos inyectar.

La clase destino es la clase en la cual deseamos inyectar la propiedad. Para eso, necesitamos declarar la propiedad, y anotarla con @Autowired. SpringBoot se encarga de implementar todo lo necesario, es decir que inyecta las dependencias.

Este caso es habitual cuando tenemos un servicio y un repositorio.

1. <https://www.arquitecturajava.com/usando-el-patron-factory/> [↑](#footnote-ref-1)