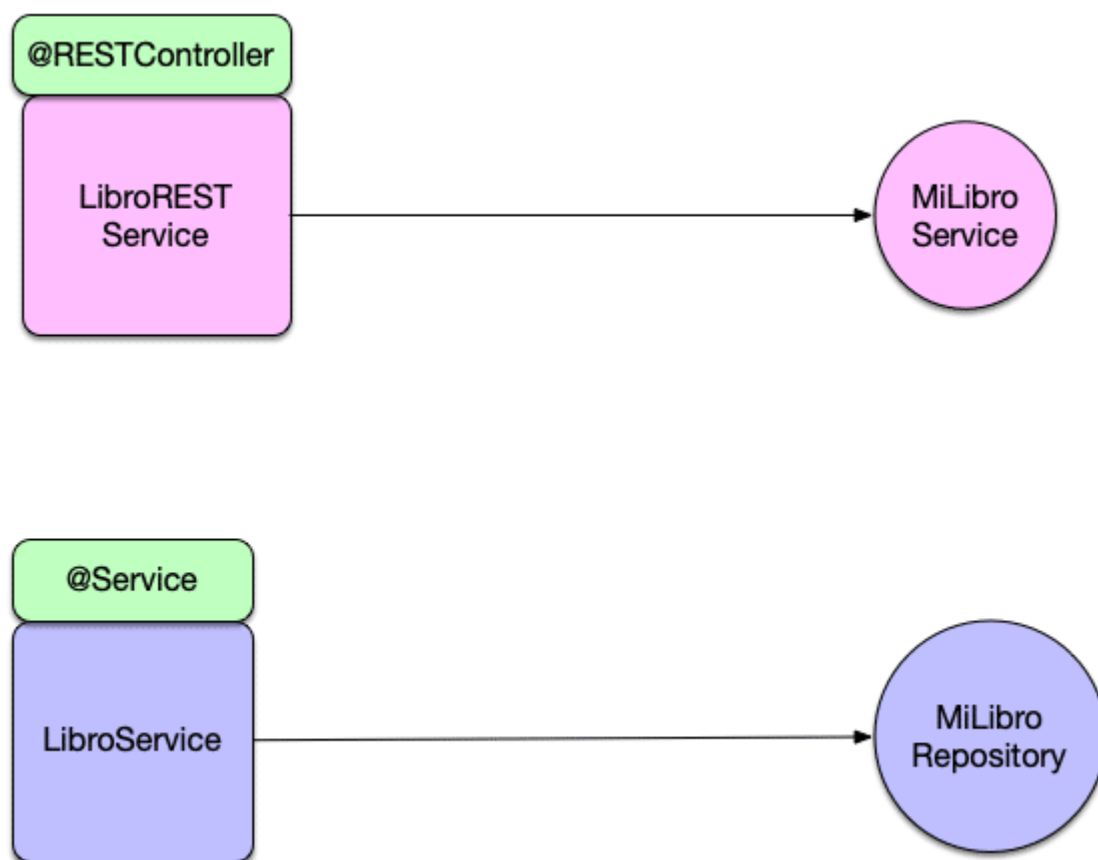


Acabamos de ver como usar Java equals y hashCode

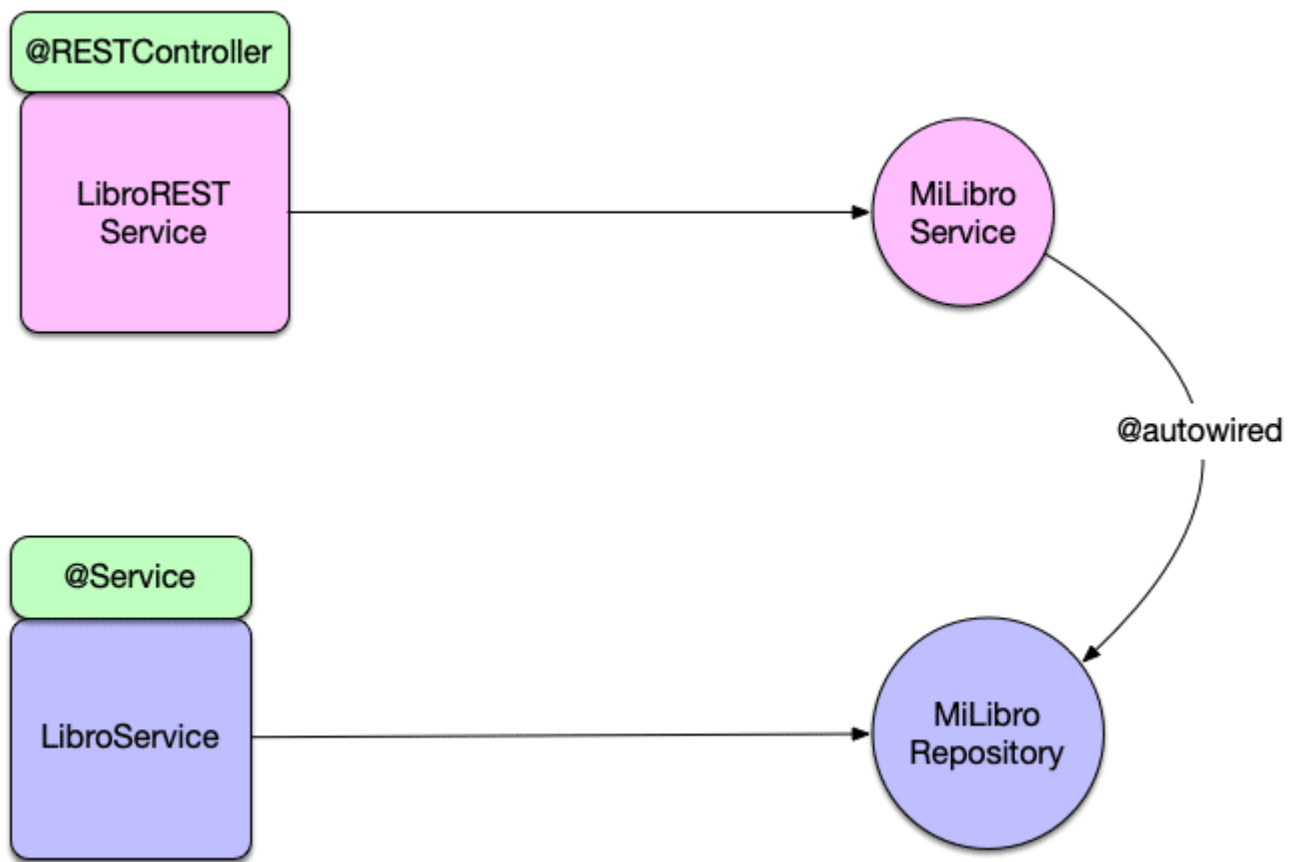
Spring @Autowired es una de las anotaciones más habituales cuando trabajamos con Spring Framework ya que se trata de la anotación que permite inyectar unas dependencias con otras dentro de Spring .

Spring @Autowired y su funcionamiento

Normalmente estamos acostumbrados a usar @AutoWired a nivel de la propiedad que deseamos inyectar. Spring funciona como **una mega factoria** de objetos. Cada clase se registra para instanciar objetos con alguna de las anotaciones @Controller ,@Service ,@repository o @RestController.



Una vez que los objetos estan creados la anotación Spring @Autowired se encarga de construir las ligazones entre los distintos elementos



Por ejemplo este caso es habitual cuando tenemos un servicio y un repositorio. Vamos a verlo usando Spring Boot

```
package com.arquitecturajava.wired;

import java.util.ArrayList;
import java.util.List;

import org.springframework.stereotype.Component;
import org.springframework.stereotype.Repository;

@Repository
public class LibroRepository {
```

```

        public List<Libro> buscarTodos() {
            List<Libro> lista= new ArrayList<Libro>();
            lista.add(new Libro ("1","java","pedro"));
            lista.add(new Libro ("2","python","gema"));
            return lista;
        }
    }
}

```

Como podemos ver tenemos el LibroRepository como primer componente y le queremos enlazar desde un Servicio.

```

package com.arquitecturajava.wired;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class LibroRestService {

    @Autowired
    LibroRepository repositorio;
    @RequestMapping("/libros")
    public List<Libro> buscarTodos() {
        return repositorio.buscarTodos();
    }
}

```

Ahora tenemos los dos elementos enlazados y un ServicioREST publicado por lo tanto si ejecutamos la aplicación de Spring Boot dispondremos de una Url con un listado de Libros

disponible

```
1 // 20200511171742
2 // http://localhost:8080/libros
3
4 ▾ [
5 ▾ {
6     "isbn": "1",
7     "titulo": "java",
8     "autor": "pedro"
9 },
10 ▾ {
11     "isbn": "2",
12     "titulo": "python",
13     "autor": "gema"
14 }
15 ]
```

Spring Testing y Wiring

Hoy por hoy las metodologías de TDD cada día se usan más y aunque este uso de la anotación `@Autowired` es el más habitual suele ser más práctico y cómodo para el manejo de test realizar un `@Autowired` a nivel de constructores por lo tanto el código quedaría así:

```
package com.arquitecturajava.wired;
```

```
import java.util.List;
```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
```

```
public class LibroRestService {  
  
    LibroRepository repositorio;  
    @Autowired  
    public LibroRestService(LibroRepository repositorio) {  
        this.repositorio=repositorio;  
    }  
    @RequestMapping("/libros")  
    public List<Libro> buscarTodos() {  
        return repositorio.buscarTodos();  
    }  
}
```

De esta manera el código queda más limpio , flexible y más sencillo de usar cuando utilizemos Test Driven Development.

Otros Artículos relacionados

- [¿Que es Spring Boot?](#)
- [Spring ComponentScan](#)
- [¿Que es REST?](#)
- [Spring Framework](#)