### Exercício 10.1 – Injeção de Dependências por Campo

**Objetivo:** Demonstrar como funciona a injeção de dependências por campo no Spring Boot utilizando a anotação @Autowired.

### Parte 1: Configuração do Projeto no Spring Initializr

1. Acesse o [Spring Initializr](https://start.spring.io/).
2. Configure o projeto com as seguintes opções:
   * **Project:** Maven
   * **Language:** Java
   * **Spring Boot:** Versão estável mais recente
   * **Group:** com.example
   * **Artifact:** demo
   * **Name:** demo
   * **Package Name:** com.example.demo
   * **Packaging:** Jar
   * **Java Version:** 21 (ou outra versão suportada pela sua máquina)
3. Adicione as seguintes dependências:
   * **Spring Web:** Para criar endpoints REST.
   * **Spring Boot DevTools (opcional):** Para facilitar o desenvolvimento com recarregamento automático.
   * **Spring Boot Actuator (opcional):** Para monitorizar a aplicação (não necessário para este exercício, mas útil para expandir o projeto futuramente).
4. Clique no botão **Generate** para descarregar o projeto como um ficheiro ZIP.
5. Extraia o ficheiro ZIP e abra o projeto na sua IDE (e.g., VS Code ou IntelliJ).

### Parte 2: Estrutura Inicial do Projeto

Ao abrir o projeto, a estrutura inicial será semelhante a esta:

src/  
├── main/  
│ ├── java/  
│ │ └── com.example.demo/  
│ │ ├── DemoApplication.java  
└── resources/  
 └── application.properties

**Nota:** As pastas service e controller ainda não existem. Terás que criá-las manualmente.

### Parte 3: Criar a Estrutura de Pastas

1. **Na IDE:**
   * Dentro da pasta com.example.demo, crie duas novas subpastas:
     + service
     + controller
2. Após criar estas pastas, a estrutura deverá ser semelhante a:

src/  
├── main/  
│ ├── java/  
│ │ └── com.example.demo/  
│ │ ├── DemoApplication.java  
│ │ ├── service/  
│ │ └── controller/  
└── resources/  
 └── application.properties

### Parte 4: Implementação do Serviço de Email

1. Na pasta service, crie o ficheiro ServicoEmail.java.
2. Insira o seguinte código:

package com.example.demo.service;  
  
import org.springframework.stereotype.Component;  
  
@Component  
public class ServicoEmail {  
  
 public void enviarEmail(String destinatario, String mensagem) {  
 System.out.println("Enviando e-mail para " + destinatario + ": " + mensagem);  
 }  
}

### Parte 5: Implementação do Serviço de Notificação

1. Na pasta service, crie o ficheiro ServicoNotificacao.java.
2. Insira o seguinte código:

package com.example.demo.service;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Component;  
  
@Component  
public class ServicoNotificacao {  
  
 @Autowired  
 private ServicoEmail servicoEmail;  
  
 public void notificar(String destinatario, String mensagem) {  
 servicoEmail.enviarEmail(destinatario, mensagem);  
 }  
}

### Parte 6: Criar o Controlador REST

1. Na pasta controller, crie o ficheiro NotificacaoController.java.
2. Insira o seguinte código:

package com.example.demo.controller;  
  
import com.example.demo.service.ServicoNotificacao;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RequestParam;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class NotificacaoController {  
  
 @Autowired  
 private ServicoNotificacao servicoNotificacao;  
  
 @GetMapping("/notificar")  
 public String notificar(@RequestParam String destinatario, @RequestParam String mensagem) {  
 servicoNotificacao.notificar(destinatario, mensagem);  
 return "Notificação enviada para " + destinatario;  
 }  
}

### Parte 7: Configuração do Projeto

1. No ficheiro application.properties (localizado em src/main/resources), adicione o seguinte para acompanhar os logs básicos:

logging.level.org.springframework=INFO

2. Certifique-se de que o ficheiro DemoApplication.java contém:

package com.example.demo;

import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class DemoApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.run(DemoApplication.class, args);  
 }  
}

### Parte 8: Testar o Projeto

**1. Iniciar o Spring Boot:**

Execute o projeto na sua IDE ou utilize o comando, no Windows:

./mvnw spring-boot:run

Se estiver a usar Linux ou Mac OS, o comando deve ser:

mvn spring-boot:run

Pode sempre correr também a aplicação carregando em “Run” por cima do método main, ou fazendo “play” no botão do canto superior direito.

**2. Aceder ao Endpoint:**

No browser ou Postman, aceda a:

http://localhost:8080/notificar?destinatario=joao@email.com&mensagem=Bem-vindo!

**3. Verificar a Saída no Terminal:**

O terminal deverá exibir:

* + Enviando e-mail para joao@email.com: Bem-vindo!

A página Web deverá exibir:

* + Notificação enviada para joao@email.com