

# Aula 4: Principais Dependências e Bibliotecas

## Abstrai a complexidade de configuração com o Spring Boot

# Objetivos

1. Swagger

2. Feign

## Aula 4 | Etapa 1:

# Swagger

Abstrai a complexidade de  
configuração com o Spring Boot

# Swagger

Swagger é um projeto open source com diversas ferramentas utilizadas para desenvolver APIs com a especificação OpenAPI Specification (OAS), uma especificação para descrever, produzir, consumir e visualizar serviços de uma API REST. Com o OAS você poderá descrever recursos, URIs, modelo de dados, métodos HTTP aceitos e códigos de resposta.

Tudo isso facilita a vida dos desenvolvedores de diversas linguagens de programação na criação, testes, consumo e documentação de APIs REST.

---



# Dependências

```
<dependency>  
  <groupId>io.springfox</groupId>  
  <artifactId>springfox-swagger2</artifactId>  
  <version>2.9.2</version>  
</dependency>  
<dependency>  
  <groupId>io.springfox</groupId>  
  <artifactId>springfox-swagger-ui</artifactId>  
  <version>2.9.2</version>  
</dependency>
```

# Desafio

Utilize o Swagger para documentar alguma API REST desenvolvida por você, seguindo as orientações da aula.

<https://github.com/Re04nan/dio-experts-spring-boot-java>

---

## Aula 4 | Etapa 2:

# Feign

Abstraia a complexidade de  
configuração com o Spring Boot

# Feign

O Feign é um cliente de serviço web declarativo (cliente HTTP) desenvolvido pela Netflix e um dos mais populares do Spring Cloud Component. Com ele é possível criar clientes API HTTP no java de forma mais simples para chamar/consumir os serviços REST, utilizando anotações.

Para usar o Feign, precisamos adicionar a dependência **spring-cloud-starter-openfeign** no pom.xml.

---



# Desafio

Utilize o Feign para integrar duas APIs ou simular esse tipo de requisição entre dois projetos, seguindo as orientações da aula.

<https://github.com/Re04nan/dio-experts-spring-boot-java>

---



DIGITAL  
INNOVATION  
ONE

# Dúvidas?

- > Fórum do curso
- > Comunidade online (discord)