



Spring Web

Aula 8 - Feign Client e Deploy no Heroku

Sumário

- Feign Client
- Heroku
- Desafio

Feign Client

- Facilita a escrita de clientes de serviço da web como chamada de APIs externas.
- O **Feign** foi criado justamente com o objetivo de **reduzir a complexidade** para consumir serviços.
- O Spring incorporou o Feign em sua stack de cloud, simplificando ainda mais a configuração e integração com nossa aplicação.

Dependências

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-openfeign</artifactId>
</dependency>

<dependency>
  <groupId>io.github.openfeign</groupId>
  <artifactId>feign-okhttp</artifactId>
</dependency>

<dependency>
  <groupId>io.github.openfeign</groupId>
  <artifactId>feign-jackson</artifactId>
</dependency>

<dependency>
  <groupId>io.github.openfeign</groupId>
  <artifactId>feign-slf4j</artifactId>
</dependency>

<dependency>
  <groupId>io.github.openfeign</groupId>
  <artifactId>feign-gson</artifactId>
</dependency>

<dependency>
  <groupId>io.github.openfeign</groupId>
  <artifactId>feign-httpclient</artifactId>
</dependency>
```

**

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-dependencies</artifactId>
      <version>2021.0.3</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
```

Feign Client

- Habilitar Feign
- Criar configuração do Feign
- Criar cliente que irá chamar um serviço terceiro
- Utilizar Serviço



Exercício

- Configurar o Feign Client no seu projeto.
- Criar um Client para a API dados pessoais (**<https://dados-pessoais-vemserdbc.herokuapp.com>**).
 - Esse cliente deve poder listar os dados pessoais e criar dados pessoais.
- Criar um novo Service para Dados Pessoais e utilizar o novo Client nele.
- Criar um novo Controller para Dados Pessoais e expor os métodos na API.

Heroku

- **Heroku** é uma plataforma na nuvem que faz **deploy** de várias aplicações back-end seja para hospedagem, testes em produção ou escalar as suas aplicações.
- Também tem integração com o GitHub, deixando o uso mais fácil e com **containers** denominados [Dynos](#).
- Aceita várias linguagens, inclusive **Java**.
- Utiliza o repositório do github para executar o deploy.

Heroku

- Para subir o app, criem um **repo novo** com o **projeto na raiz**;
- Adicionem um arquivo **system.properties** com o conteúdo **java.runtime.version=17** ;
- Verifiquem se todos os properties estão no **application.properties**;
- **Properties sensíveis** (senhas, VM Args) vão no **Heroku > Settings > Config Vars**;
- **NUNCA COMITEM SENHAS NO REPO!**

```
java.runtime.version=17
```

system.properties 1 KB

Config Vars

debug	true
spring.mail.password	
spring.mail.username	rafael
spring.profiles.active	prd



Homework

- Utilizando a API dados pessoais (<https://dados-pessoais-vemserdbc.herokuapp.com>):
- Disponibilizar todas as operações da API atual (Controllers), assim como a Dados Pessoais.
- Publicar o projeto no Heroku.

Desafio

- Criar novos endpoints/dtos na API da pessoa para (create, list, delete e update) com os dados dela + os dados pessoais.
- Ou seja, nesses novos endpoints informar também os dados pessoais e chamar o serviço para buscar/insérer/atualizar/remover essa informação também do serviço junto com os dados que a pessoa já tem na pessoa-api.

Obrigado!

DBC

DIGITAL BUSINESS COMPANY®



 /dbc.company

 /dbccompany

 /dbccompany.com.br

 /company/dbc-company