FuturoDEV

Exercícios da Semana

# COMO REALIZAR OS EXERCÍCIOS

Recomenda-se criar uma pasta no computador para armazenar as respostas dos exercícios semanais. No primeiro módulo do curso, os alunos irão receber os enunciados através de um documento texto, após (segundo módulo), os exercícios serão disponibilizados semanalmente em um quadro do *Trello*, onde os alunos irão movimentar os cartões de exercícios/tarefas, seguindo a metodologia *Kanban*, e armazenar as respostas em um repositório no *GitHub*.

# [M01S06] Ex 01 - Classe Produto como Entidade Spring

Crie uma classe com nome Produto, os atributos serão id, nome e preço. Com o uso das anotações necessárias para que o Spring realize as operações de criação de tabelas e colunas

# [M01S06] Ex 02 - API Calculador de IMC

Crie uma api que receba os dados através de métodos POST → peso (double) e altura (double) e retorne o cálculo do IMC referente os dados passados , deverá considerar erros de dados repassados para chamada, realizando então a devolução da chamada com status e mensagem de tratamento. O tipo de parâmetro que será passado pode ser de sua escolha.

# [M01S06] Ex 03 - API de Alunos pesquisa por nome

Aproveitando a estrutura já criada em aula com a classe Aluno, faça uma pesquisa de aluno por nome, o método será GET ← passando nome do aluno como Parâmetro da pesquisa. O retorno deverá ser uma lista de Alunos

# [M01S06] Ex 04 - API de Alunos pesquisa por nome e idade

Aproveitando a estrutura já criada em aula com a classe Aluno, realize uma alteração adicionando a informação de idade (int) do Aluno Agora através do método GET ← desenvolva um novo recurso para pesquisa de alunos por nome e idade. O retorno deverá ser uma lista de Alunos.

# [M01S06] Ex 05 - CRUD com Spring e Banco de Dados PostgreSQL

1. Conforme exemplo visto em aula, acesse o Spring Initializr e configure um projeto com as seguintes dependências:
   1. Spring Web;
   2. PostgreSQL Driver;
   3. Lombok;
   4. Spring Data JPA;
2. Baixe o arquivo zip e abra no IntelliJ;
3. Abra o programa PgAdmin 4 em seu computador e crie um database para a utilização na aplicação. Saiba a porta, usuário e senha do seu database para ser possível configurá-lo em sua aplicação Java.
4. Após isso, configure a conexão de sua aplicação com o seu banco de dados PostgreSQL no arquivo application.properties, deixando o arquivo com as seguintes configurações:

# datasource PostgreSQL

spring.datasource.url = jdbc:postgresql://localhost:<SuaPorta>/<SeuNomeDeDataBase>

spring.datasource.username=<SeuUsuario>

spring.datasource.password=<SuaSenha>

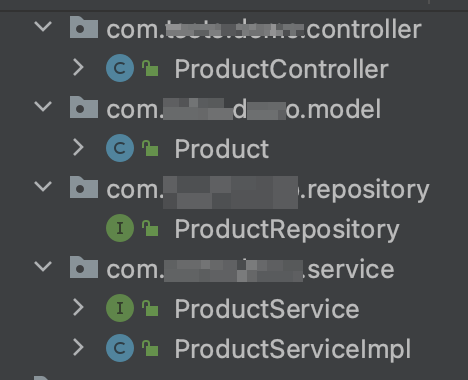
# jpa

spring.jpa.database-plataform=org.hibernate.dialect.PostgreSQLDialect

spring.jpa.hibernate.ddl-auto=update

spring.jpa.properties.hibernate.show\_sql=true

spring.jpa.properties.hibernate.format\_sql=true

1. Depois disso, recompile, recarregue o projeto e clique em play em sua aplicação. Verifique no terminal se não foi apresentado nenhum erro. Caso não tenha sido apresentado nenhum erro, significa que sua aplicação se conectou com seu database;
2. Agora, crie os pacotes model, controller, service e repository em sua aplicação para que fique nesta estrutura:
   1. 
3. Codifique as classes com suas devidas annotations:
   1. model > Product.java:
      1. Atributos: id, name, description, value, amount;
   2. repository > ProductRepository.java
      1. interface com herança de JpaRepository<Product, Long>
   3. service > ProductService.java:
      1. Devendo conter os comportamentos do JpaRepository
   4. service: ProductServiceImpl.java
      1. Devendo implementar ProductService
   5. controller > ProductController.java
      1. Esta classe deve controlar as requisições http. Necessário implementar as seguintes operações: criar um produto (POST), atualizar um produto (PUT), listar os produtos cadastrados (GET) e remover um produto (DELETE).
4. Configure no Insomnia as requisições, valide se estão todas funcionando;
5. Abre o PGAdmin e dê um “select \* from product” para verificar se foram inseridos produtos;
6. Com a aplicação Java sem erros, Insomnia devidamente funcionando, persistindo dados no banco de dados PostgreSQL. Export o JSON do Insomnia e coloque dentro de uma pasta do projeto, compacte o projeto para envio na entrega dos exercícios ao final do módulo.