

Documentação Teste JAVA Ws WORK

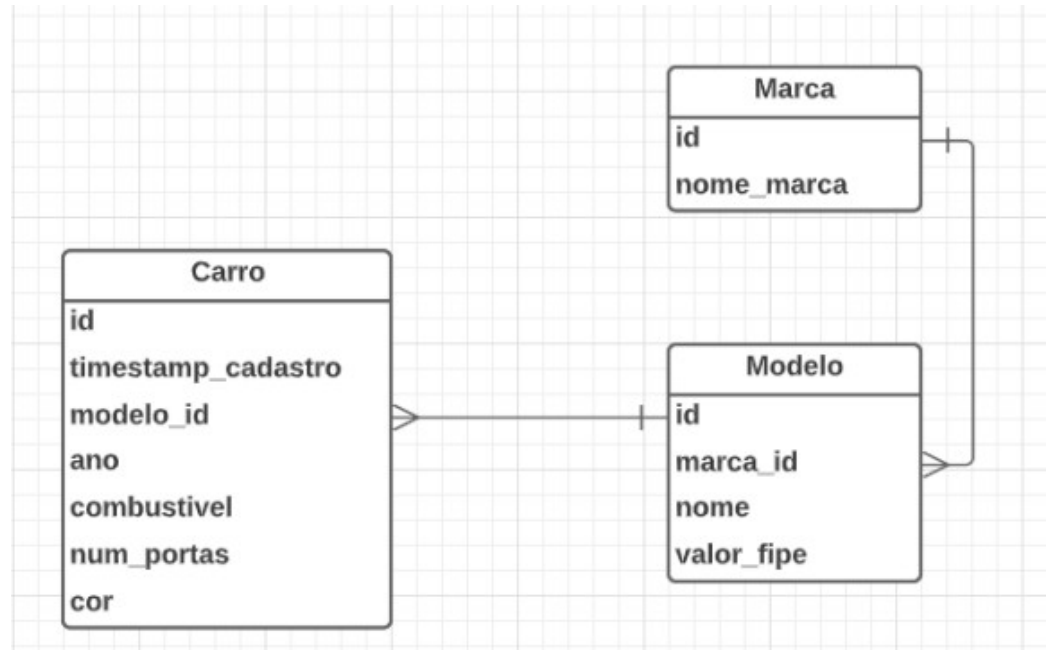
Thiago Vinícius Oliveira De Matos Rodrigues

Sobre o Projeto

Projeto desenvolvido durante teste para a Ws Work.

Objetivo:

Desenvolver Uma Aplicação SpringBoot , com acesso a Banco de dados (Spring-Data), endpoints Rest para listagem completa dos carros, com informações do modelo e da marca relacionados. Endpoints para CRUD de carros, modelos e marcas seguindo o modelo ER ao lado.



Tecnologias usadas

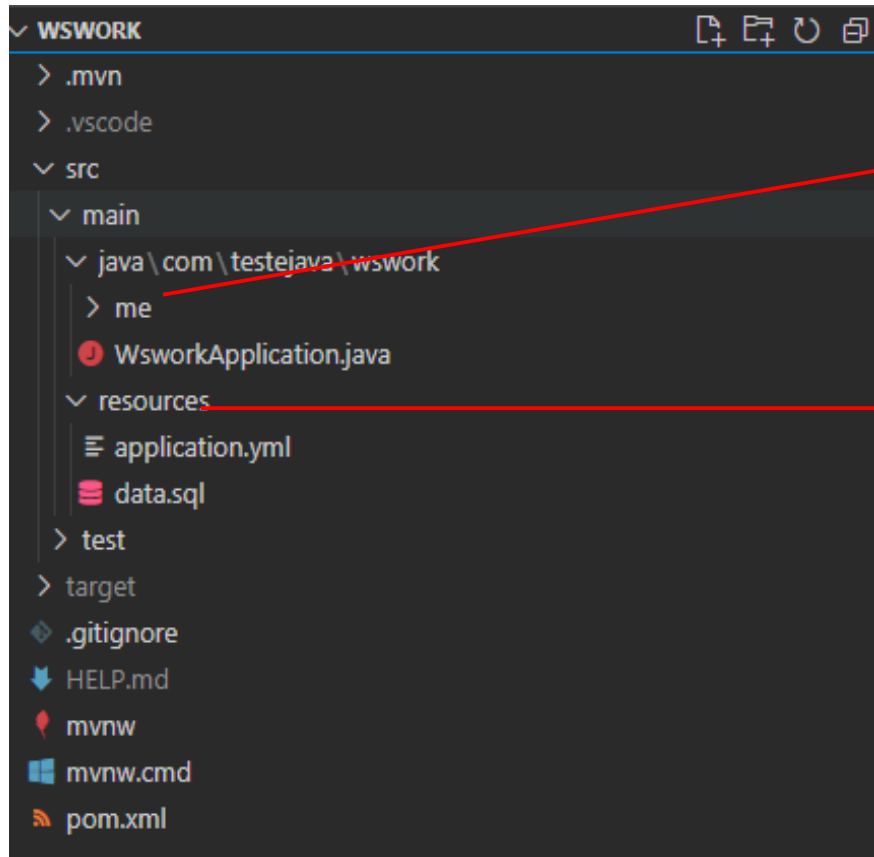
- Spring Initializr:
 - para gerar a estrutura inicial do projeto.
- Dependências:
 - Spring Web
 - Spring Data JPA
 - Validation (Hibernate Validator)
 - Lombok
 - PostgreSQLDriver:
 - Para conexão com o Banco de dados PostgreSQL, com o objetivo de aprofundar os conhecimentos nesse sistema gerenciador de banco de dados.
- Configurações:
 - Projeto Maven, Spring Boot(2.7.0), Java 18, pacote no formato de arquivos jar.

Como foi Desenvolvido

Pensando em separação de responsabilidades e escalabilidade, o projeto foi desenvolvido separado em camadas: Controller, service e repository. Onde o Controller chama o Service responsável pela regra de negócio , que chama o repository responsável pela persistência dos dados na base de dados.

Já sobre o endpoint de listagem dos carros com informações do seu modelo e marca. O controller HomeController chama o service do carro(CarroService), que por sua vez tem o método que instancia a classe criada exclusivamente para exposição dos dados desse endpoint, separando assim a funcionalidade de listar carros, desse endpoint. Então o carroService fica com essa responsabilidade já que a partir do carro, pode se acessar o modelo, e a partir do modelo acessar a marca , mesmo usando a técnica de carregamento lento(Por questões de performance). Assim tendo todos os dados necessários para a exibição desejada.

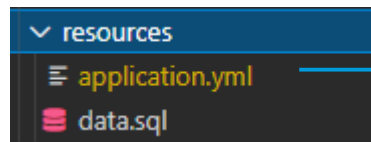
Estrutura do projeto



Pasta com os arquivos
Com a lógica da
aplicação.

Pasta com
configurações de
acesso a banco de
dados e aplicação.

Estrutura do projeto



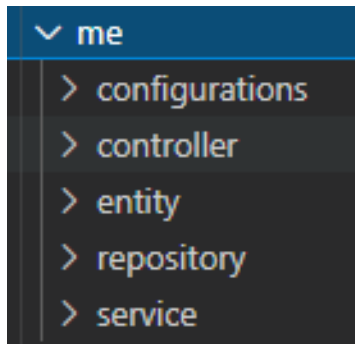
Arquivo para inserção de registros automaticamente, ao inicializar a aplicação.

Configuração para conexão com a base de dados, definições para gerar tabelas automaticamente e carregamento de scripts sql, ao iniciar a aplicação

```
spring:
  datasource:
    url: jdbc:postgresql://localhost:5432/wswork?useTimezone=true&serverTimezone=UTC&useLegacyDate
    username: postgres
    password: root

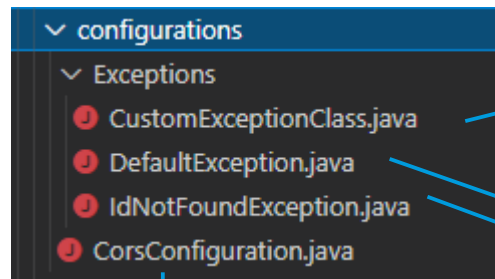
  jpa:
    show-sql: true
    hibernate:
      ddl-auto: create
      defer-datasource-initialization: true
      properties:
        hibernate:
          format_sql: true
    sql:
      init:
        mode: always
```

Estrutura do projeto



- Configurations:
 - reservada para arquivos relacionados a algum tipo de configuração que envolva a regra de negócio
- Controller:
 - Todos os controllers que configuram os endpoints da API REST.
- Entity:
 - Pasta onde está a definição das entidades, modelos que serão usados na manipulação dos dados.
- Repository:
 - Possui as classes de repositório, camada criada para separar as responsabilidades da persistência dos dados das camadas de regra de negócio.
- Service:
 - Camada onde a regra de negócio de acesso aos dados é implementada.

Estrutura do projeto

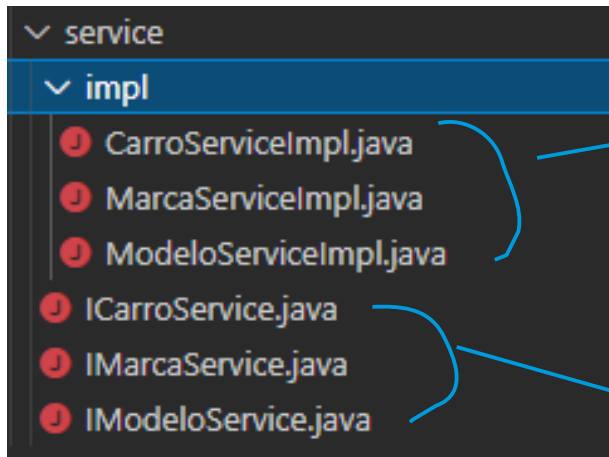


Configuração de
permissões de acesso
a API.

Classe de Tratamento de
exceções usando
`@RestControllerAdvice`

Classes de Exceções
personalizadas

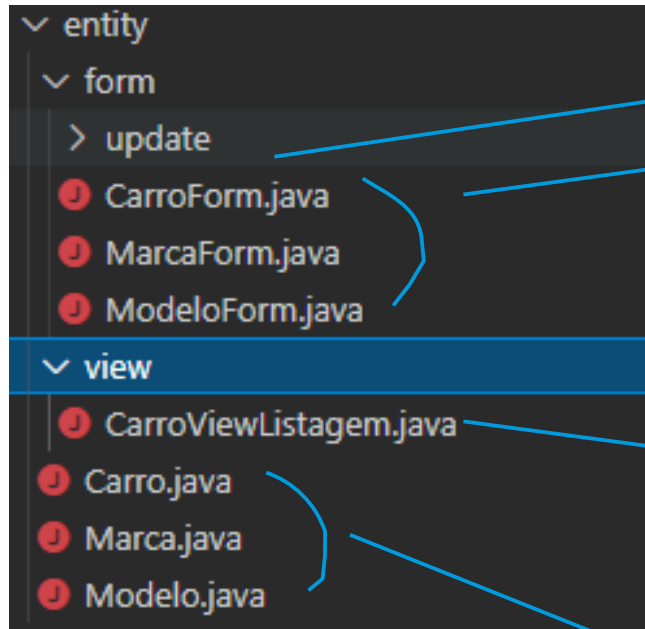
Estrutura do projeto



Implementação dos services de fato, usando as interfaces.

Interfaces que definem os métodos que os services vão possuir

Estrutura do projeto



Para um melhor tratamento de dados, foi criado classes para os dados recebidos tanto para criar registros quanto para atualizar, podendo definir melhor que dados podem ser atualizados , que dados podem ser usados para cadastro e etc...

Classe que reúne dados das três entidades relacionadas(carro, modelo e marca), usado no HomeController

Classes dos modelos bases