Vehicle.java

package com.example.vehicle\_service.model;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;

@Entity // JPA annotation to mark this class as a database entity

@AllArgsConstructor // Lombok annotation to generate an all-arguments constructor

@NoArgsConstructor // Lombok annotation to generate a no-argument constructor

@Data // Lombok annotation to generate getters, setters, toString, equals, and hashCode methods

@Table(name = "vehicles") // Specifies the table name in the database

public class Vehicle {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Long serviceId;

    private String vehicleType;

    private int serviceYear;

}

VehicleRepository.java

package com.example.vehicle\_service.repository;

import com.example.vehicle\_service.model.Vehicle;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import org.springframework.data.repository.query.Param;

import jakarta.transaction.Transactional;

import java.util.List;

import java.util.Optional;

public interface VehicleRepository extends JpaRepository<Vehicle, Long> {

    List<Vehicle> findByServiceYear(int serviceYear);

    @Query("SELECT v.vehicleType FROM Vehicle v WHERE v.serviceId = :serviceId")

    Optional<String> findVehicleTypeByServiceId(@Param("serviceId") Long serviceId);

    @Transactional

    void deleteByServiceYear(int serviceYear);

}

VehicleService.java

package com.example.vehicle\_service.service;

import com.example.vehicle\_service.model.Vehicle;

import java.util.List;

import java.util.Optional;

public interface VehicleService {

    List<Vehicle> getAllVehicles();

    List<Vehicle> getVehiclesByYear(int year);

    Optional<String> getVehicleTypeById(Long serviceId);

    void deleteVehiclesByYear(int year);

    Vehicle createVehicle(Vehicle vehicle);

}

VehicleServiceImpl.java

package com.example.vehicle\_service.service.impl;

import com.example.vehicle\_service.model.Vehicle;

import com.example.vehicle\_service.repository.VehicleRepository;

import com.example.vehicle\_service.service.VehicleService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.List;

import java.util.Optional;

@Service

public class VehicleServiceImpl implements VehicleService {

    private final VehicleRepository vehicleRepository;

    @Autowired

    public VehicleServiceImpl(VehicleRepository vehicleRepository) {

        this.vehicleRepository = vehicleRepository;

    }

    @Override

    public List<Vehicle> getAllVehicles() {

        return vehicleRepository.findAll();

    }

    @Override

    public List<Vehicle> getVehiclesByYear(int year) {

        return vehicleRepository.findByServiceYear(year);

    }

    @Override

    public Optional<String> getVehicleTypeById(Long serviceId) {

        return vehicleRepository.findVehicleTypeByServiceId(serviceId);

    }

    @Override

    public void deleteVehiclesByYear(int year) {

        vehicleRepository.deleteByServiceYear(year);

    }

    @Override

    public Vehicle createVehicle(Vehicle vehicle) {

        return vehicleRepository.save(vehicle);

    }

}

VehicleController.java

package com.example.vehicle\_service.controller;

import com.example.vehicle\_service.model.Vehicle;

import com.example.vehicle\_service.service.VehicleService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

@RequestMapping("/api/vehicles")

public class VehicleController {

    private final VehicleService vehicleService;

    @Autowired

    public VehicleController(VehicleService vehicleService) {

        this.vehicleService = vehicleService;

    }

    @GetMapping

    public ResponseEntity<List<Vehicle>> getAllVehicles() {

        List<Vehicle> vehicles = vehicleService.getAllVehicles();

        return ResponseEntity.ok(vehicles);

    }

    @GetMapping("/year/{year}")

    public ResponseEntity<List<Vehicle>> getVehiclesByServiceYear(@PathVariable int year) {

        List<Vehicle> vehicles = vehicleService.getVehiclesByYear(year);

        return ResponseEntity.ok(vehicles);

    }

    @GetMapping("/type/{id}")

    public ResponseEntity<String> getVehicleTypeById(@PathVariable Long id) {

        return vehicleService.getVehicleTypeById(id)

                .map(ResponseEntity::ok)

                .orElse(ResponseEntity.notFound().build());

    }

    @DeleteMapping("/year/{year}")

    public ResponseEntity<String> deleteVehiclesByYear(@PathVariable int year) {

        vehicleService.deleteVehiclesByYear(year);

        return ResponseEntity.ok("Successfully deleted all service records for the year: " + year);

    }

    @PostMapping

    public ResponseEntity<Vehicle> createVehicle(@RequestBody Vehicle vehicle) {

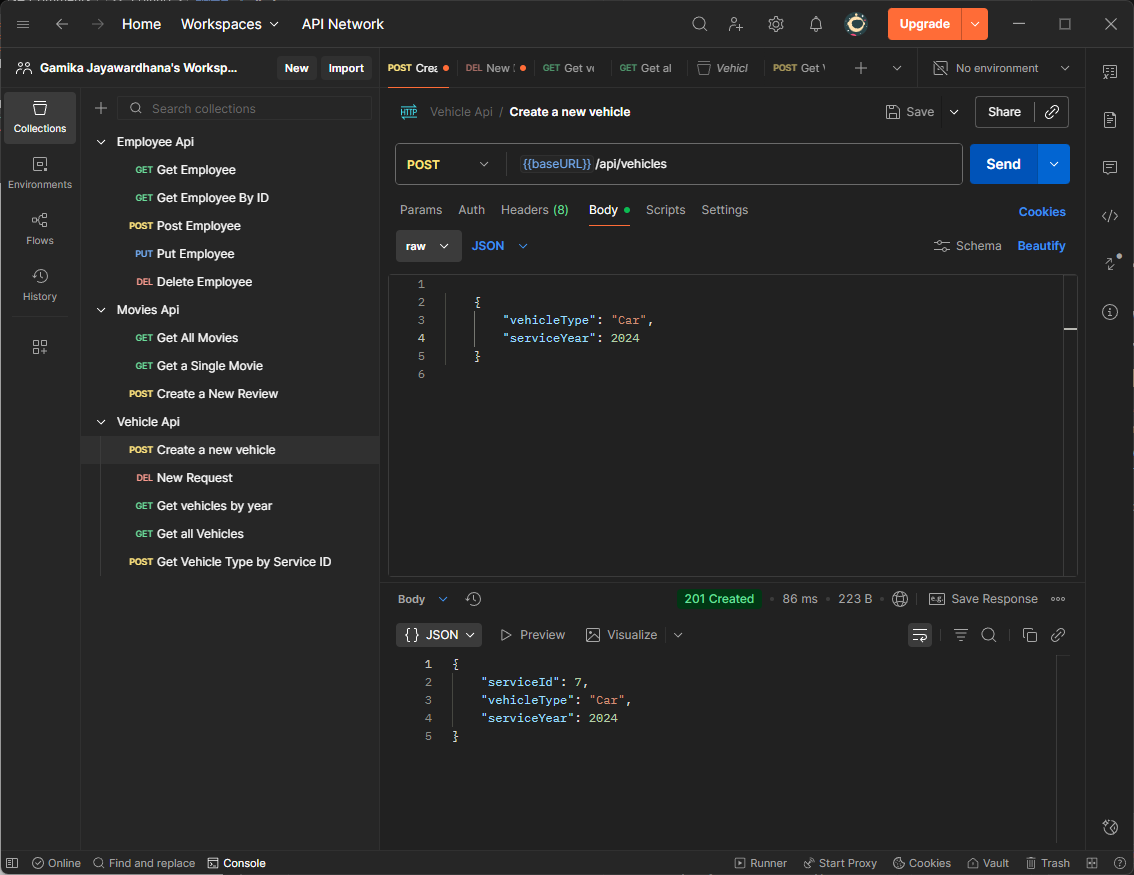
        Vehicle newVehicle = vehicleService.createVehicle(vehicle);

        return new ResponseEntity<>(newVehicle, HttpStatus.CREATED);

    }

}

GithubLink: [GamikaJayawardana/vehicle-service](https://github.com/GamikaJayawardana/vehicle-service)



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.