

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
```

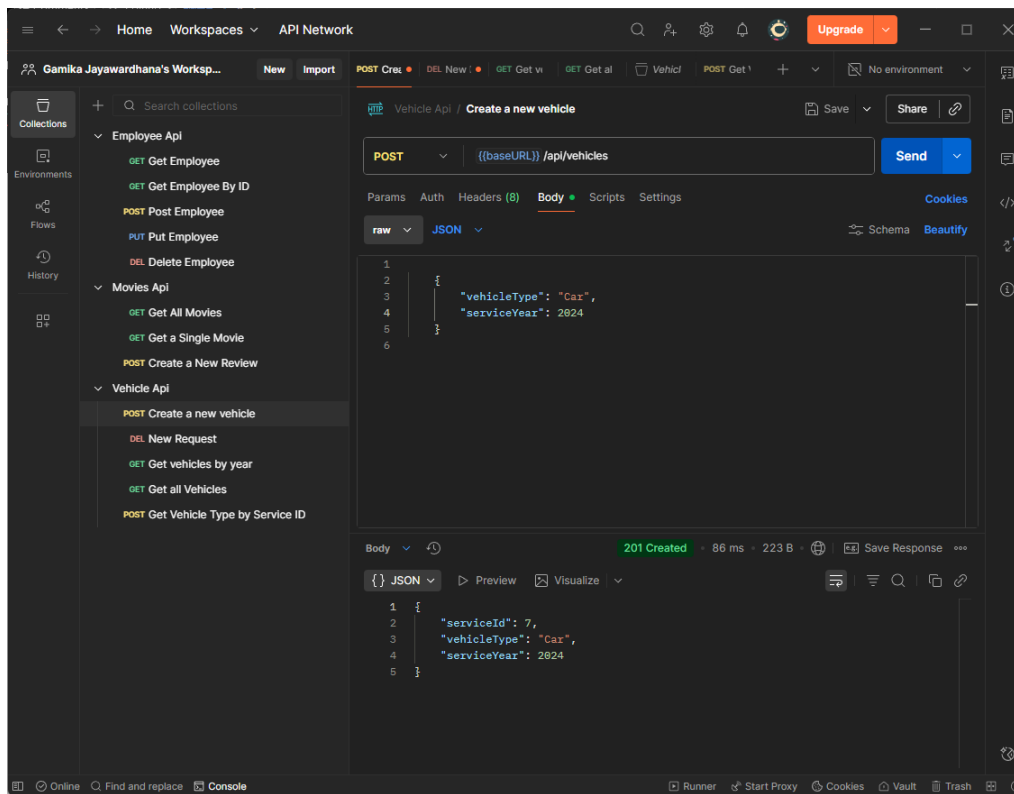
LAB 03

EC/2021/006

W.K.G.K JAYAWARDANA

```
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)



LAB 03

EC/2021/006

W.K.G.K JAYAWARDANA

The screenshot shows the API Network interface with the 'Vehicle Api' selected. The endpoint is 'Get all Vehicles' with a GET method. The URL is '[[baseUrl]] /api/vehicles'. The response is a 200 OK status with a 9 ms response time and 556 B of data. The response body is in JSON format, showing a list of four vehicles.

Key	Value	Description
Key	Value	Description

```

1 [
2   {
3     "serviceId": 1,
4     "vehicleType": "Truck",
5     "serviceYear": 2025
6   },
7   {
8     "serviceId": 2,
9     "vehicleType": "Bike",
10    "serviceYear": 2024
11  },
12  {
13    "serviceId": 3,
14    "vehicleType": "Truck",
15    "serviceYear": 2025
16  },
17  {
18    "serviceId": 4,
19    "vehicleType": "Van",
20    "serviceYear": 2025
21  },
22  {
23    "serviceId": 5
  }
]
  
```

The screenshot shows the API Network interface with the 'Vehicle Api' selected. The endpoint is 'Get vehicles by year' with a GET method. The URL is '[[baseUrl]] /api/vehicles/year/2023'. The response area is currently empty, showing a placeholder with a cartoon character and the text 'Click Send to get a response'.

LAB 03

EC/2021/006

W.K.G.K JAYAWARDANA

The screenshot shows the Postman interface with a workspace named 'Gamika Jayawardhana's Worksp...'. The left sidebar lists collections: 'Employee Api' (with GET, POST, PUT, and DELETE methods) and 'Movies Api' (with GET and POST methods). The 'Vehicle Api' collection is expanded, showing 'Get Vehicle Type by Service ID' selected. The main panel displays the request details for this endpoint: a GET method with a URL template '({baseUrl}) /api/vehicles/type/2'. The 'Query Params' section is empty. The response is shown as '200 OK' with a status of '24 ms' and a body of '1 Bike'.

Key	Value	Description
Key	Value	Description

The screenshot shows the Postman interface with the same workspace. The 'Vehicle Api' collection is expanded, and 'New Request' is selected. The main panel displays the request details for this endpoint: a DELETE method with a URL template '({baseUrl}) /api/vehicles/year/2024'. The 'Query Params' section is empty. The response is shown as '200 OK' with a status of '139 ms' and a body of '1 Successfully deleted all service records for the year: 2024'.

Key	Value	Description
Key	Value	Description