Load's APIS

This code in application.properties will connect API to PostgreSQL

#changing the server port

server.port = 9090

#database configuration:PostgreSQL

spring.datasource.url=jdbc:postgresql://localhost:5432/amigoscode

spring.datasource.username=root

spring.datasource.password=root

#hibernate configuration

spring.jpa.hibernate.ddl-auto=create-drop

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect

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

We will write this code in LoadServiceImpl.java file

package com.springrest.springrest.services;

import java.util.ArrayList;

import java.util.List;

import java.util.stream.Collectors;

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

import org.springframework.stereotype.Service;

import com.springrest.springrest.dao.LoadDao;

import com.springrest.springrest.entities.Load;

@Service

public class LoadServiceImpl implements LoadService {

@Autowired

private LoadDao LoadDao;

List<Load> list;

public LoadServiceImpl() {

list = new ArrayList<>();

list.add(newLoad("delhi","jaipur","chemicals","canter","1","100","","shipper:<UUID>","01-01-2023">

}

@Override

public List<Load> getLoads() {

// TODO Auto-generated method stub

return loadDao.findAll();

}

@Override

public Load getLoad(long loadId) {

Load l = null;

for(Load load:list)

{

if(load.getId() == loadId)

{

l = load;

break;

}

}

return loadDao.getOne(loadId);

}

@Override

public Load addLoad(Load load) {

list.add(load);

loadDao.save(load);

return load;

}

@Override

public Load updateLoad(Load load) {

list.forEach(e -> {

if(e.getId() == load.getId()) {

e.setTitle(load.getTitle());

e.setDescription(load.getDescription());

}

});

loadDao.save(load);

return load;

}

@Override

public void deleteLoad(long parseLong) {

list = this.list.stream().filter(e->e.getId()!=parseLong).collect(Collectors.toList());

Load entity = loadDao.getOne(parseLong);

loadDao.delete(entity);

}

}

We will write this code in Load.java

package com.springrest.springrest.entities;

import javax.persistence.Entity;

import javax.persistence.Id;

@Entity

public class Load {

@Id

private String loadingPoint;

private String unloadingPoint;

private String productType;

private String truckType;

private long noOfTrucks;

private long weight;

private String comment;

private int shipperId;

private date Date;

public Load(String loadingPoint, String unloadingPoint, String productType, String truckType, long noOfTrucks,

long weight,String comment,int shipperId,date Date) {

super();

this.loadingPoint = loadingPoint;

this.unloadingPoint = unloadingPoint;

this.productType = productType;

this.truckType = truckType;

this.noOfTrucks = noOfTrucks;

this.weight = weight;

this.comment = comment;

this.shipperId = shipperId;

this.date = date;

}

}

We will write this code in LoadService.java file

package com.springrest.springrest.services;

import java.util.List;

import com.springrest.springrest.entities.Load;

public interface LoadService {

public List<Load> getLoads();

public Load getLoad(long loadId);

public Load addLoad(Load load);

public Load updateLoad(Load load);

public void deleteLoad(long parseLong);

}

We will write this code in MyController.java file

package com.springrest.springrest.controller;

import java.util.List;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import com.springrest.springrest.entities.Load;

import com.springrest.springrest.services.LoadService;

@RestController

public class MyController {

@Autowired

private LoadService loadSerivice;

// get the loads

@GetMapping("/load/{shipperId}")

public List<Load> getLoads()

{

return this.loadSerivice.getLoads();

}

// single load get

@GetMapping("/load/{loadId}")

public Load getLoad(@PathVariable String loadId)

{

return this.loadSerivice.getLoad(Long.parseLong(loadId));

}

// load add

@PostMapping("/load")

public Load addLoad(@RequestBody Load load)

{

return this.loadSerivice.addLoad(load);

}

// update load using PUT request

@PutMapping("/load/{loadId}")

public Course update\load(@RequestBody Load load) {

return this.loadSerivice.updateLoad(load);

}

// delete the load

@DeleteMapping("/load/{loadId}")

public ResponseEntity<HttpStatus> deleteLoad(@PathVariable String loadId) {

try {

this.courseSerivice.deleteLoad(Long.parseLong(loadId));

return new ResponseEntity<>(HttpStatus.OK);

} catch(Exception e) {

return new ResponseEntity<>(HttpStatus.INTERNAL\_SERVER\_ERROR);

}

}

}