package com.ust.carapp.controller;

import java.util.List;

import java.util.Optional;

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

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.RequestMapping;

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

import com.ust.carapp.model.Car;

import com.ust.carapp.repository.CarRepo;

@RestController

@RequestMapping("/api/car")

public class CarController {

@Autowired

private CarRepo repo;

@PostMapping("/addcar")

public ResponseEntity<Car> addcar(@RequestBody Car car) {

repo.save(car);

return ResponseEntity.ok().body(car);

}

@PutMapping("/{id}")

public ResponseEntity<Car> updatecar(@PathVariable int id, @RequestBody Car car) {

Optional<Car> cari = repo.findById(id);

if (cari.isPresent()) {

return ResponseEntity.ok().body(repo.save(car));

}

else {

return ResponseEntity.noContent().build();

}

}

@GetMapping("/{id}")

public ResponseEntity<Car>getid (@PathVariable int id){

Optional<Car> cari=repo.findById(id);

if(cari.isPresent()) {

return ResponseEntity.ok().body(repo.findById(id).get());

}

else {

return ResponseEntity.noContent().build();

}

}

@DeleteMapping("/{id}")

public ResponseEntity<Void>delete(@PathVariable int id){

Optional<Car>cari=repo.findById(id);

if(cari.isPresent()) {

repo.deleteById(id);

return ResponseEntity.ok().build();

}

else {

return ResponseEntity.noContent().build();

}

}

@GetMapping

public ResponseEntity<List<Car>> getAll(){

List<Car> c=repo.findAll();

return ResponseEntity.ok().body(c);

}

}

int width=0;

        for(int i=0;i<heights.length;i++){

            if(heights[i]>H){

                width+=2;

            }

            else{

                width+=1;

            }

        }

        return width;

for(int i=0;i<K;i++){

            //while(K!=0){

               int t=N%10;

               if(t==0){

                   N=N/10;

               }

               else{

                   N=N-1;

               }

            }

        //}

        return N;

int out=0;

      Arrays.sort(Arr);

     return Arr[Arr.length-1]-Arr[0]+1-Arr.length;