create rest api using spring boot

1. Create a spring boot application with dependencies

Spring web

Spring data jpa

Mysql driver

1. Application.properties

spring.datasource.url=jdbc:mysql://localhost:3306/ust1

spring.datasource.username=root

spring.datasource.password=

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

1. Entity: Food

package com.ust.spring.entity;

import java.util.Arrays;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Lob;

@Entity

public class Food {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Integer id;

private String name;

private Double price;

@Lob

private byte[] photo;

public Food() {}

public Food(Integer id, String name, Double price, byte[] photo) {

super();

this.id = id;

this.name = name;

this.price = price;

this.photo = photo;

}

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Double getPrice() {

return price;

}

public void setPrice(Double price) {

this.price = price;

}

public byte[] getPhoto() {

return photo;

}

public void setPhoto(byte[] photo) {

this.photo = photo;

}

@Override

public String toString() {

return "Food [id=" + id + ", name=" + name + ", price=" + price + ", photo=" + photo.length + "]";

}

}

1. Repository:

package com.ust.spring.repository;

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

import org.springframework.stereotype.Repository;

import com.ust.spring.entity.Food;

@Repository

public interface FoodRepository extends JpaRepository<Food, Integer>

{

}

1. Service:

package com.ust.spring.service;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import com.ust.spring.entity.Food;

import com.ust.spring.repository.FoodRepository;

@Service

public class FoodService {

@Autowired

private FoodRepository fr;

public Food create(Food food) {

return fr.save(food);

}

public List<Food> read() {

return fr.findAll();

}

public Food read(Integer id) {

Optional<Food> temp = fr.findById(id);

Food food=null;

if(temp.isPresent())

{

food=temp.get();

}

return food;

}

public Food update(Food food) {

Food f=read(food.getId());

if(f!=null)

{

f=food;

fr.save(f);

}

return f;

}

public Food delete(Integer id) {

Food f=read(id);

if(f!=null)

{

fr.delete(f);

}

return f;

}

}

1. Controller:

package com.ust.spring.controller;

import java.util.List;

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

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.RequestMethod;

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

import com.ust.spring.entity.Food;

import com.ust.spring.service.FoodService;

@RestController

@RequestMapping("/food")

public class FoodController {

@Autowired

private FoodService fs;

// @RequestMapping(method = RequestMethod.POST, value = "/food", params = "add")

@PostMapping

public Food addFood(@RequestBody Food food) {

return fs.create(food);

}

@GetMapping

public List<Food> retrieveAllFoods() {

return fs.read();

}

@GetMapping("/{id}")

public Food findFoodById(@PathVariable("id") Integer id) {

return fs.read(id);

}

@PutMapping

public Food updateFood(@RequestBody Food food) {

return fs.update(food);

}

@DeleteMapping("/{id}")

public Food deleteFood(@PathVariable("id") Integer id) {

return fs.delete(id);

}

}

1. Test the food api using post man
2. Create angular project

ng new food-project --routing --style=css

1. Cd food-project
2. Code .
3. ng serve -o
4. create a food component

ng g c food

1. plug the component to app-component
2. go to app.module.ts and import the following components

FormsModule

ReactiveFormsModule

HttpClientModule

Access to XMLHttpRequest at 'http://localhost:8080/food' from origin 'http://localhost:4200' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.

CORS = Cross Origin Resource Sharing

To enable CORS, in Spring rest controller, use @CrossOrigin

To set max file size allowed for multipart: (in application.properties)

spring.servlet.multipart.max-file-size=500MB

spring.servlet.multipart.max-request-size=500MB

in mysql:

SET GLOBAL max\_allowed\_packet=1073741824;

Task:

CORS Angular + Spring REST