**BackEnd Code**

**Entity**

**Admin:**

package com.simplilearn.demo.entity;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Admin {

@Id

@Column(name="id")

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private String email;

private String password;

private String country;

private int login;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getCountry() {

return country;

}

public void setCountry(String country) {

this.country = country;

}

public int getLogin() {

return login;

}

public void setLogin(int login) {

this.login = login;

}

}

**User:**

**package com.example.demo.pojo;**

**import javax.persistence.Column;**

**import javax.persistence.Entity;**

**import javax.persistence.GeneratedValue;**

**import javax.persistence.GenerationType;**

**import javax.persistence.Id;**

**@Entity**

**public class User {**

**@Id**

**@Column(name="id")**

**@GeneratedValue(strategy = GenerationType.IDENTITY)**

**private int id;**

**private String name;**

**private String email;**

**private String password;**

**private String address;**

**private int login;**

**public int getId() {**

**return id;**

**}**

**public void setId(int id) {**

**this.id = id;**

**}**

**public String getName() {**

**return name;**

**}**

**public void setName(String name) {**

**this.name = name;**

**}**

**public String getEmail() {**

**return email;**

**}**

**public void setEmail(String email) {**

**this.email = email;**

**}**

**public String getPassword() {**

**return password;**

**}**

**public void setPassword(String password) {**

**this.password = password;**

**}**

**public String getAddress() {**

**return address;**

**}**

**public void setAddress(String address) {**

**this.address = address;**

**}**

**public int getLogin() {**

**return login;**

**}**

**public void setLogin(int login) {**

**this.login = login;**

**}**

**}**

**Food**

package com.example.demo.pojo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Food {

@Id

@Column(name="id")

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private String category;

private String description;

private float discount;

private float price;

private String available;

private String cuisine;

private String image;

public String getCuisine() {

return cuisine;

}

public void setCuisine(String cuisine) {

this.cuisine = cuisine;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getCategory() {

return category;

}

public void setCategory(String category) {

this.category = category;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public float getDiscount() {

return discount;

}

public void setDiscount(float discount) {

this.discount = discount;

}

public float getPrice() {

return price;

}

public void setPrice(float price) {

this.price = price;

}

public String getAvailable() {

return available;

}

public void setAvailable(String available) {

this.available = available;

}

public String getImage() {

return image;

}

public void setImage(String image) {

this.image = image;

}

}

**Cuisine**

package com.example.demo.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import lombok.Data;

@Entity

public class Cuisine {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int cid;

private String cuisine;

public int getCid() {

return cid;

}

public void setCid(int cid) {

this.cid = cid;

}

public String getCuisine() {

return cuisine;

}

public void setCuisine(String cuisine) {

this.cuisine = cuisine;

}

}

**Cart**

package com.example.demo.pojo;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToOne;

import lombok.Data;

@Entity

@Data

public class Cart {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private int id;

@OneToOne(cascade=CascadeType.ALL)

@JoinColumn(name="food\_id")

private Food fd;

}

**BankDetails**

package com.example.demo.pojo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import lombok.Data;

@Data

@Entity

public class BankDetails {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int bid;

private String bankname;

private int balance;

}

**Repository:-**

**AdminRepo:**

**package** com.simplilearn.demo.repo;

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

**import** com.simplilearn.demo.entity.Admin;

**public** **interface** AdminRepo **extends** JpaRepository<Admin,Integer>{

}

**UserRepo:**

**package** com.simplilearn.demo.repo;

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

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

**import** org.springframework.stereotype.Repository;

**import** com.simplilearn.demo.entity.User;

@Repository

**public** **interface** UserRepo **extends** JpaRepository<User,Integer>{

User findByEmail(String cust\_email);

// @Query("select u.Username from User u where u.userid=:userid")

// public String getById(int userid);

}

**FoodRepo:**

package com.example.demo.repo;

import java.util.List;

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

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

import org.springframework.stereotype.Repository;

import com.example.demo.pojo.Food;

@Repository

public interface FoodRepo extends JpaRepository<Food,Integer>{

// @Query("Select f from Food f where f.name LIKE %?1%"+"OR f.cuisine LIKE %?1%")

// public List<Food> findBySearch(String keyword);

//

@Query("Select f from Food f where f.cuisine=?1")

public List<Food> findByCuisine(String Keyword);

@Query("Select f from Food f where f.id=?1")

public Food findBbyId(int id);

@Query("Select f from Food f ORDER BY f.price ASC")

public List<Food> sortASC();

@Query("Select f from Food f ORDER BY f.price DESC")

public List<Food> sortDESC();

}

**CartRepo:**

package com.example.demo.repo;

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

import org.springframework.stereotype.Repository;

import com.example.demo.pojo.Cart;

@Repository

public interface CartRepo extends JpaRepository<Cart, Integer>{

Object deleteById(long id);

}

**CuisineRepo:**

package com.example.demo.repo;

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

import com.example.demo.pojo.Cuisine;

public interface CuisineRepo extends JpaRepository<Cuisine, Integer>{

}

**BankRepo:**

package com.example.demo.repo;

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

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

import com.example.demo.pojo.BankDetails;

public interface BankRepo extends JpaRepository<BankDetails, Integer> {

@Query("Select b.balance from BankDetails b where b.bankname=?1")

public int findBalanceByName(String bname);

@Query("Select bd.balance from BankDetails bd where bd.bid=?1")

public int findBalanceByID(int bid);

}

**Service:-**

**AdminService:**

**package** com.simplilearn.demo.service;

**import** java.util.List;

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

**import** org.springframework.stereotype.Service;

**import** com.simplilearn.demo.entity.Admin;

**import** com.simplilearn.demo.repo.AdminRepo;

@Service

**public** **class** AdminService {

@Autowired

**private** AdminRepo repo;

**public** Admin addAdmin(Admin a) {

**return** repo.save(a);

}

**public** List<Admin> getAllAdmin(){

**return** repo.findAll();

}

**public** Admin getAdminById(**int** id) {

**if**(repo.findById(id).isPresent()) {

**return** repo.findById(id).get();

}

**else** {

**return** **null**;

}

}

**public** Admin updateAdmin(**int** id, Admin newAdmin) {

**if**(repo.findById(id).isPresent()) {

Admin admin = repo.findById(id).get();

admin.setName(newAdmin.getName());

admin.setEmail(newAdmin.getEmail());

admin.setCountry(newAdmin.getCountry());

admin.setPassword(newAdmin.getPassword());

**return** repo.save(admin);

}

**else** {

**return** **null**;

}

}

**public** **boolean** deleteAdmin(**int** id) {

**if** (repo.findById(id).isPresent()){

repo.deleteById(id);

**return** **true**;

}

**else** {

**return** **false**;

}

}

}

**UserService**

package com.example.demo.service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.example.demo.pojo.User;

import com.example.demo.repo.UserRepo;

@Service

public class UserService {

@Autowired

private UserRepo repo;

public User addUser(User u) {

return repo.save(u);

}

public List<User> getAllUser(){

return repo.findAll();

}

public User getUserById(int id) {

if(repo.findById(id).isPresent()) {

return repo.findById(id).get();

}

else {

return null;

}

}

public User updateUser(int id, User newUser) {

if(repo.findById(id).isPresent()) {

User olduser= repo.findById(id).get();

olduser.setName(newUser.getName());

olduser.setEmail(newUser.getEmail());

olduser.setPassword(newUser.getPassword());

olduser.setAddress(newUser.getAddress());

olduser.setLogin(newUser.getLogin());

return repo.save(olduser);

}

else {

return null;

}

}

public boolean deleteUser(int id) {

if(repo.findById(id).isPresent()) {

repo.deleteById(id);

return true;

}

else {

return false;

}

}}

**FoodService**

package com.example.demo.service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.example.demo.pojo.Food;

import com.example.demo.repo.FoodRepo;

@Service

public class FoodService {

@Autowired

private FoodRepo repo;

public Food addFood(Food f) {

return repo.save(f);

}

public List<Food> getAllFood(){

return repo.findAll();

}

public Food getFoodById(int id) {

if(repo.findById(id).isPresent()) {

return repo.findById(id).get();

}

else {

return null;

}

}

public Food updateFood(int id, Food newFood) {

if(repo.findById(id).isPresent()) {

Food food = repo.findById(id).get();

food.setName(newFood.getName());

food.setCategory(newFood.getCategory());

food.setDescription(newFood.getDescription());

food.setAvailable(newFood.getAvailable());

food.setDiscount(newFood.getDiscount());

food.setPrice(newFood.getPrice());

food.setImage(newFood.getImage());

return repo.save(food);

}

else {

return null;

}

}

public boolean deleteFood(int id) {

if(repo.findById(id).isPresent()) {

repo.deleteById(id);

return true;

}

else {

return false;

}

}

//public List<Food> findBySearch(String keyword){

// return repo.findBySearch(keyword);

//}

//

public List<Food> findByCuisine(String keyword){

return repo.findByCuisine(keyword);

}

public List<Food> sortAsc(){

return repo.sortASC();

}

public List<Food> sortDESC(){

return repo.sortDESC();

}

}

**BankService**

package com.example.demo.service;

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

import org.springframework.stereotype.Service;

import com.example.demo.pojo.BankDetails;

import com.example.demo.repo.BankRepo;

@Service

public class BankService {

@Autowired

BankRepo repo;

public BankDetails insert(BankDetails bd) {

return repo.save(bd);

}

public int findBalanceById(int id) {

return repo.findBalanceByID(id);

}

public int findBalanceByBname(String bname) {

return repo.findBalanceByName(bname);

}

}

**CartService**

package com.example.demo.service;

import java.util.List;

import javax.servlet.http.HttpSession;

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

import org.springframework.http.ResponseEntity;

import org.springframework.stereotype.Service;

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

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

import com.example.demo.pojo.Cart;

import com.example.demo.repo.CartRepo;

@Service

public class CartService {

@Autowired

CartRepo repo;

public Cart insert(Cart c) {

return repo.save(c);

}

public List<Cart> findAll(){

return repo.findAll();

}

public boolean deleteAllFromCart() {

repo.deleteAll();

return true;

}

}

**CuisineService**

package com.example.demo.service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.example.demo.pojo.Cuisine;

import com.example.demo.repo.CuisineRepo;

@Service

public class CuisineService {

@Autowired

CuisineRepo repo;

public Cuisine insert(Cuisine c) {

return repo.save(c);

}

public List<Cuisine> findAll(){

return repo.findAll();

}

}

**Controller:-**

**AdminController:**

package com.simplilearn.demo;

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.CrossOrigin;

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.simplilearn.demo.entity.Admin;

import com.simplilearn.demo.entity.User;

import com.simplilearn.demo.service.AdminService;

@CrossOrigin("http://localhost:4200/")

@RestController

@RequestMapping("/api/admin")

public class AdminController {

@Autowired

private AdminService service;

@PostMapping("/")

public ResponseEntity<Admin> addAdmin(@RequestBody Admin a){

Admin admin= service.addAdmin(a);

if(admin!=null)

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

else

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

}

@GetMapping("/")

public List<Admin> getAllAdmin(){

return service.getAllAdmin();

}

@GetMapping("/{id}")

public ResponseEntity<Admin>getAdminById (@PathVariable int id){

Admin admin= service.getAdminById(id);

if(admin!=null)

return new ResponseEntity<Admin>(admin, HttpStatus.OK);

else

return new ResponseEntity<Admin>(admin, HttpStatus.NOT\_FOUND);

}

@PutMapping("/{id}")

public ResponseEntity<Object> updateAdmin(@PathVariable int id,@RequestBody Admin newAdmin){

Admin admin= service.updateAdmin(id, newAdmin);

if (admin!=null)

return new ResponseEntity<Object>(admin,HttpStatus.OK);

else

return new ResponseEntity<Object>("No User Available to Update",HttpStatus.NOT\_FOUND);

}

@DeleteMapping("/{id}")

public ResponseEntity<String>deleteAdmin(@PathVariable int id){

boolean result = service.deleteAdmin(id);

if(result)

return new ResponseEntity<String>("Object Deleted",HttpStatus.OK);

else

return new ResponseEntity<String>("NO user Found", HttpStatus.NOT\_FOUND);

}

}

**UserController:**

package com.simplilearn.demo;

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

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

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

import com.simplilearn.demo.entity.User;

import com.simplilearn.demo.service.UserService;

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.CrossOrigin;

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;

@CrossOrigin("http://localhost:4200/")

@RestController

@RequestMapping("/api/user")

public class UserController {

@Autowired

private UserService service;

@PostMapping("/")

public ResponseEntity<User> addUser(@RequestBody User u){

User user= service.addUser(u);

if(user!=null)

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

else

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

}

@GetMapping("/")

public List<User> getAllUser(){

return service.getAllUser();

}

@GetMapping("/{id}")

public ResponseEntity<User>getUserById (@PathVariable int id){

User user= service.getUserById(id);

if(user!=null)

return new ResponseEntity<User>(user, HttpStatus.OK);

else

return new ResponseEntity<User>(user, HttpStatus.NOT\_FOUND);

}

@PutMapping("/{id}")

public ResponseEntity<Object> updateUser(@PathVariable int id,@RequestBody User newUser){

User user= service.updateUser(id, newUser);

if (user!=null)

return new ResponseEntity<Object>(user,HttpStatus.OK);

else

return new ResponseEntity<Object>("No User Available to Update",HttpStatus.NOT\_FOUND);

}

@DeleteMapping("/{id}")

public ResponseEntity<String>deleteUser(@PathVariable int id){

boolean result = service.deleteUser(id);

if(result)

return new ResponseEntity<String>("Object Deleted",HttpStatus.OK);

else

return new ResponseEntity<String>("NO user Found", HttpStatus.NOT\_FOUND);

}

}

**BankController:**

package com.example.demo.controller;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

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.RequestBody;

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

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

import com.example.demo.pojo.BankDetails;

import com.example.demo.service.BankService;

@CrossOrigin("http://localhost:4200/")

@RestController

@RequestMapping("/api/bank")

public class BankController {

@Autowired

private BankService service;

@PostMapping("bank/addBank")

public ResponseEntity<BankDetails> addBank(@RequestBody BankDetails bd) {

BankDetails mm=service.insert(bd);

if(mm!=null) {

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

}

else {

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

}

}

@GetMapping("bank/getBankDetails/{id}")

public int getBankById(@PathVariable int id){

return service.findBalanceById(id);

}

@GetMapping("bank/getBankDetailsByBname/{bname}")

public int getBankByBname(@PathVariable String bname){

return service.findBalanceByBname(bname);

}

}

**FoodController:**

package com.example.demo.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.CrossOrigin;

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.example.demo.pojo.Food;

import com.example.demo.service.FoodService;

@CrossOrigin("http://localhost:4200/")

@RestController

@RequestMapping("/api/food")

public class FoodController {

@Autowired

private FoodService service;

@PostMapping("/")

public ResponseEntity<Food> addFood(@RequestBody Food f){

Food food= service.addFood(f);

if(food!=null)

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

else

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

}

@GetMapping("/")

public List<Food> getAllFood(){

return service.getAllFood();

}

@GetMapping("/{id}")

public ResponseEntity<Food>getFoodById (@PathVariable int id){

Food food= service.getFoodById(id);

if(food!=null)

return new ResponseEntity<Food>(food, HttpStatus.OK);

else

return new ResponseEntity<Food>(food, HttpStatus.NOT\_FOUND);

}

@PutMapping("/{id}")

public ResponseEntity<Object> updateFood(@PathVariable int id,@RequestBody Food newFood){

Food food = service.updateFood(id, newFood);

if (food!=null)

return new ResponseEntity<Object>(food,HttpStatus.OK);

else

return new ResponseEntity<Object>("No User Available to Update",HttpStatus.NOT\_FOUND);

}

@DeleteMapping("/{id}")

public ResponseEntity<String>deleteFood(@PathVariable int id){

boolean result = service.deleteFood(id);

if(result)

return new ResponseEntity<String>("Object Deleted",HttpStatus.OK);

else

return new ResponseEntity<String>("NO user Found", HttpStatus.NOT\_FOUND);

}

@GetMapping("food/getAllFoodByASC")

public List<Food> getAllFoodASC(){

return service.sortAsc();

}

@GetMapping("food/getAllFoodByDESC")

public List<Food> getAllFoodDESC(){

return service.sortDESC();

}

// @GetMapping("food/getAllFoodBySearch/{keyword}")

// public List<Food> getAllFoodBySearch(@PathVariable String keyword){

// return service.findBySearch(keyword);

// }

@GetMapping("food/getAllFoodByCuisine/{keyword}")

public List<Food> getAllFoodByCuisine(@PathVariable String keyword){

return service.findByCuisine(keyword);

}

}

**CartController:**

package com.example.demo.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.CrossOrigin;

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

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

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

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

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

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

import com.example.demo.pojo.Cart;

import com.example.demo.service.CartService;

@CrossOrigin("http://localhost:4200/")

@RestController

@RequestMapping("/api/cart")

public class CartController {

@Autowired

private CartService service;

@PostMapping("cart/addToCart")

public ResponseEntity<Cart> addToCart(@RequestBody Cart bd) {

Cart mm=service.insert(bd);

if(mm!=null) {

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

}

else {

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

}

}

@GetMapping("cart/getAllCart")

public List<Cart> getCart(){

return service.findAll();

}

@DeleteMapping("cart/flush")

public ResponseEntity<String>deleteAllFromCart(){

boolean result = service.deleteAllFromCart();

if(result)

return new ResponseEntity<String>("Object Deleted",HttpStatus.OK);

else

return new ResponseEntity<String>("NO movie Found", HttpStatus.NOT\_FOUND);

}

}

**CuisineController:**

package com.example.demo.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.CrossOrigin;

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

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

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

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

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

import com.example.demo.pojo.Cuisine;

import com.example.demo.service.CuisineService;

@CrossOrigin("http://localhost:4200/")

@RestController

@RequestMapping("/api/cuisine")

public class CuisineController {

@Autowired

CuisineService service;

@GetMapping("cuisine/getAllCuisine")

public List<Cuisine> getAllCuisine(){

return service.findAll();

}

@PostMapping("cuisine/addCuisine")

public ResponseEntity<Cuisine> addCuisine(@RequestBody Cuisine f) {

Cuisine mm=service.insert(f);

if(mm!=null) {

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

}

else {

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

}

}

}

**config**

**SimpleCORSFilter**

package com.example.demo.config;

import java.io.IOException;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.stereotype.Component;

public class SimpleCORSFilter implements Filter{

private final Logger log = LoggerFactory.getLogger(SimpleCORSFilter.class);

public SimpleCORSFilter() {

log.info("SimpleCORSFilter init");

}

@Override

public void doFilter(ServletRequest req, ServletResponse res, FilterChain chain)

throws IOException, ServletException {

HttpServletRequest request = (HttpServletRequest) req;

HttpServletResponse response = (HttpServletResponse) res;

response.setHeader("Access-Control-Allow-Origin", request.getHeader("Origin"));

response.setHeader("Access-Control-Allow-Credentials", "true");

response.setHeader("Access-Control-Allow-Methods", "POST, PUT, GET, OPTIONS, DELETE");

response.setHeader("Access-Control-Max-Age", "3600");

response.setHeader("Access-Control-Allow-Headers", "Content-Type, Accept, X-Requested-With, remember-me");

chain.doFilter(req, res);

}

@Override

public void init(FilterConfig filterConfig) {

}

@Override

public void destroy() {

}

}