**Microservices Communication in Spring Boot**

package com.example.test;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class RestApiOneApplication {

public static void main(String[] args) {

SpringApplication.run(RestApiOneApplication.class, args);

}

}

Creating an Entity Class

package com.example.test;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class PersonEntity {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

@Column(name = "id", updatable = false, nullable = false)

private Integer personId;

@Column

private String name;

@Column

private Integer age;

public PersonEntity() {

super();

}

public PersonEntity(Integer personId, String name, Integer age) {

super();

this.personId = personId;

this.name = name;

this.age = age;

}

public Integer getPersonId() {

return personId;

}

public void setPersonId(Integer personId) {

this.personId = personId;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Integer getAge() {

return age;

}

public void setAge(Integer age) {

this.age = age;

}

}

Creating a Repository Class

package com.example.test;

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

import org.springframework.stereotype.Repository;

@Repository

public interface PersonRepository extends JpaRepository<PersonEntity, Integer> {

}

Creating a Service Class

package com.example.test;

import java.util.HashMap;

import java.util.Map;

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

import org.springframework.stereotype.Service;

import org.springframework.web.client.RestTemplate;

@Service

public class PersonService {

@Autowired

PersonRepository personRepository;

RestTemplate restTemplate = new RestTemplate();

public PersonResonse getPerson(int personId){

final String uri = "http://localhost:8082/webapitwo/hobby/{personId}";

Map<String, Integer> params = new HashMap<String, Integer>();

params.put("personId", personId);

String result = restTemplate.getForObject(uri, String.class, params);

PersonEntity pe=personRepository.findById(personId).get();

PersonResonse pr=new PersonResonse();

pr.setPersonId(pe.getPersonId());

pr.setName(pe.getName());

pr.setAge(pe.getAge());

pr.setHobby(result);

return pr;

}

public void addPerson(PersonEntity pe){

personRepository.save(pe);

}

}

Creating a Controller Class

package com.example.test;

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

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

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;

@RestController

@RequestMapping(path = "/webapione")

public class PersonControlller {

@Autowired

PersonService personService;

@RequestMapping("/person/{personId}")

public PersonResonse getPerson(@PathVariable int personId){

return personService.getPerson(personId);

}

@RequestMapping(method=RequestMethod.POST, value="/person")

public void addPerson(@RequestBody PersonEntity pe ) {

personService.addPerson(pe);

}

}

Creating a Response Class

package com.example.test;

public class PersonResonse {

private Integer personId;

private String name;

private Integer age;

private String hobby;

public Integer getPersonId() {

return personId;

}

public void setPersonId(Integer personId) {

this.personId = personId;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Integer getAge() {

return age;

}

public void setAge(Integer age) {

this.age = age;

}

public String getHobby() {

return hobby;

}

public void setHobby(String result) {

this.hobby = result;

}

}

Setting the port number for the project in the application properties

server.port=8081

spring.application.name=RestApiOne