1. Iniciar la configuración del sistema utilizando ctrl+shift+p. Agregar el código “Sprig initilizr crate a Maven Project”

Group id:usuario.service

Artifact id: usuario-service

1. Instalar las siguientes dependencias :
2. Spring web
3. Spring dev tolos
4. H2
5. JPA

3.- Crear el package entidades

4.- Agregar la clase Usuario con el siguiente código :

package com.usuario.service.usuario\_service.entidades;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

@Entity

public class Usuario {

    private int id;

    private String nombre;

    private String email;

    @Id /\*Se le indica al programa que la variable id

    se reconocerá como pk o identificador.  \*/

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    /\*Se establece que los valores del id se genrarán

     de forma automatica \*/

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public String getNombre() {

        return nombre;

    }

    public void setNombre(String nombre) {

        this.nombre = nombre;

    }

    public String getEmail() {

        return email;

    }

    public void setEmail(String email) {

        this.email = email;

    }

    public Usuario() {

    }

}

5.- Crear el package repositorio

6.- Agregar la clase de tipo interface UsuarioRepository con el siguiente código :

package com.usuario.service.usuario\_service.repositorio;

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

import org.springframework.stereotype.Repository;

import com.usuario.service.usuario\_service.entidades.Usuario;

@Repository

public interface UsuarioRepository extends JpaRepository<Usuario,Integer>{

/\*

 \* findAll() --> Retorna todos los usuarios

 \* findById(Integer id) -->Retorna un usuario según su id

 \* save(Usuario usuario) --> Guarda o agrega un nuevo usuario.

 \*  Si el id ya existe, actualiza al usuario

 \* deleteById(Integer id)--> Elimina a un usuario según si id

 \*/

}

7.- Crear el package servicio

8.- Crear la clase UsuarioServicio que tiene el siguiente código :

package com.usuario.service.usuario\_service.servicio;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.usuario.service.usuario\_service.entidades.Usuario;

import com.usuario.service.usuario\_service.repositorio.UsuarioRepository;

@Service

public class UsuarioServicio {

    @Autowired

    private UsuarioRepository usuarioRepository;

    public List<Usuario>getAll(){

        return usuarioRepository.findAll();

    }

    public Usuario getUsuarioById(int id){

        return usuarioRepository.findById(id).orElse(null);

    }

    public Usuario save(Usuario usuario){

        Usuario nuevoUsuario = usuarioRepository.save(usuario);

        return nuevoUsuario;

    }

}

9-. Crear el paquete controlador

10.- Agregar la clase UsuarioController con el siguiente código :

package com.usuario.service.usuario\_service.controlador;

import java.util.List;

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

import org.springframework.http.ResponseEntity;

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.usuario.service.usuario\_service.entidades.Usuario;

import com.usuario.service.usuario\_service.servicio.UsuarioServicio;

@RestController

@RequestMapping("/usuario")

public class UsuarioController {

    @Autowired

    private UsuarioServicio usuarioService;

    @GetMapping //Permite retornar a todos los usuarios

    public ResponseEntity<List<Usuario>> listarUsuario(){

        List<Usuario> usuarios = usuarioService.getAll();

        if(usuarios.isEmpty()){

            return ResponseEntity.noContent().build();

        }

        return ResponseEntity.ok(usuarios);

    }

    @GetMapping("/{id}")

    public ResponseEntity<Usuario> obtenerUsuario(@PathVariable("id") int id){

        Usuario usuario = usuarioService.getUsuarioById(id);

        if(usuario == null){

            return ResponseEntity.notFound().build();

        }

        return ResponseEntity.ok(usuario);

    }

    @PostMapping

    public ResponseEntity<Usuario> guardarUsuario(@RequestBody Usuario usuario){

        Usuario nuevoUsuario = usuarioService.save(usuario);

        return ResponseEntity.ok(nuevoUsuario);

    }

}

* - - - s e g u n d a p a r t e - - - - - - - - - - -

1.- Trabajando en el microservicio del usuario, crear un nuevo package con el nombre configuración. Dentro crear la clase RestTemplateConfig con el siguiente código :

package com.usuario.service.usuario\_service.configuracion;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.web.client.RestTemplate;

@Configuration

public class RestTemplateConfig {

    @Bean

    public RestTemplate restTemplate(){

        return new RestTemplate();

    }

}

2.- Crear el package modelos con la clase Producto que tiene el siguiente código : (Recordar que en este package va una clase por microservicio adicional):

package com.usuario.service.usuario\_service.modelos;

public class Producto {

    public String modelo;

    public String marca;

    public Producto() {

    }

    public String getModelo() {

        return modelo;

    }

    public void setModelo(String modelo) {

        this.modelo = modelo;

    }

    public String getMarca() {

        return marca;

    }

    public void setMarca(String marca) {

        this.marca = marca;

    }

}

3.- En la clase UsuarioServicio agregar al código existente lo siguiente:

    @Autowired

    private RestTemplate restTemplate;

    public List<Producto> getProductos(int usuarioId){

        List<Producto>productos = restTemplate.getForObject("http://localhost:8002/producto/usuario/"+usuarioId, List.class);

        return productos;

    }

4.-En la clase controlador agregar el siguente código :

    @GetMapping("/productos/{usuarioId}")

    public ResponseEntity<List<Producto>>listarProductos(@PathVariable("usuarioId")int id){

        Usuario usuario = usuarioService.getUsuarioById(id);

        if(usuario==null){

            return ResponseEntity.notFound().build();

        }

        List<Producto> productos = usuarioService.getProductos(id);

        return ResponseEntity.ok(productos);

    }