GROUP-4

To deploy a sample Spring Boot application in rest assured API

Codes

DemoApplication.java:

package com.example.test;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class DemoApplication

{

public static void main(String[] args)

{

SpringApplication.run(DemoApplication.class, args);

}

}

Product.java:

package com.example.test;

public class Product

{

private String id;

private String name;

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

ProductServiceController.java:

package com.example.test;

import java.util.Collection;

import java.util.HashMap;

import java.util.Map;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

public class ProductServiceController

{

private static Map<String, Product> productRepo = new HashMap<>();

static {

Product Vanilla = new Product();

Vanilla.setId("1");

Vanilla.setName("VANILLA");

productRepo.put(Vanilla.getId(), Vanilla);

Product Chocolate = new Product();

Chocolate.setId("2");

Chocolate.setName("CHOCOLATE");

productRepo.put(Chocolate.getId(), Chocolate);

}

@RequestMapping(value = "/products/{id}", method = RequestMethod.DELETE)

public ResponseEntity<String> delete(@PathVariable("id") String id) {

productRepo.remove(id);

return new ResponseEntity<>("Product is deleted successsfully", HttpStatus.OK);

}

@RequestMapping(value = "/products/{id}", method = RequestMethod.PUT)

public ResponseEntity<String> updateProduct(@PathVariable("id") String id, @RequestBody Product product) {

productRepo.remove(id);

product.setId(id);

productRepo.put(id, product);

return new ResponseEntity<>("Product is updated successsfully", HttpStatus.OK);

}

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

public ResponseEntity<String> createProduct(@RequestBody Product product) {

productRepo.put(product.getId(), product);

return new ResponseEntity<>("Product is created successfully", HttpStatus.CREATED);

}

@RequestMapping(value = "/products")

public ResponseEntity<Collection<Product>> getProduct() {

return new ResponseEntity<>(productRepo.values(), HttpStatus.OK);

}

}

Steps

* Install Spring tools 4.0 from the eclipse marketplace and restart the eclipse.
* Right click on the project🡪Run as🡪Spring Boot application
* Open 8080 port and view the result.



