**TASK 7**

To deploy a sample Springboot application on Rest assured API (testing).

Product.java

package com.example.test;

public class Product {

private String id;

private String model;

private String price;

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getModel() {

return model;

}

public void setModel(String model) {

this.model = model;

}

public String getPrice() {

return price;

}

public void setPrice(String price) {

this.price = price;

}

}

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 mobile1 = new Product();

mobile1.setId("1");

mobile1.setModel("Redmi");

productRepo.put(mobile1.getId(), mobile1);

mobile1.setPrice("10000");

productRepo.put(mobile1.getId(), mobile1);

Product mobile2 = new Product();

mobile2.setId("2");

mobile2 .setModel("Oppo");

productRepo.put(mobile2.getId(), mobile2);

mobile2 .setPrice("12000");

productRepo.put(mobile2.getId(), mobile2);

}

/\* @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);

}

}

DemoApplication.java

package com.example.test;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.ComponentScan;

@ComponentScan(basePackages="com.example.test")

@SpringBootApplication

public class DemoApplication {

public static void main(String[] args)

{

SpringApplication.run(DemoApplication.class, args);

}

}

A picture containing scatter chart

Description automatically generated**RESULTS**

Graphical user interface, application

Description automatically generated