Hellow Controller///////////////////////////////////////////////////////

package com.example.springJava;  
  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class HelloController {  
 @RequestMapping("/hello")  
 public String sayHi(){  
 return "hiiiiiiiiiii";  
 }  
}

product Exception Controller//////////////////////////////////

package com.example.springJava;  
  
import org.springframework.web.bind.annotation.ControllerAdvice;  
import org.springframework.http.HttpStatus;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.ExceptionHandler;  
import org.springframework.web.context.request.WebRequest;  
  
@ControllerAdvice  
  
public class ProductExceptionController {  
 @ExceptionHandler(value = ProductNotfoundException.class)  
 public ResponseEntity<Object> exception(ProductNotfoundException exception) {  
 return new ResponseEntity<>("Product not found", HttpStatus.*NOT\_FOUND*);  
 //return new ResponseEntity<>("Product not found", HttpStatus.BAD\_REQUEST);  
 }  
/\*/////////////////////////////////////////////\*/  
@ExceptionHandler(Exception.class)  
public final ResponseEntity<Object> handleAllExceptions(Exception ex, WebRequest request) {  
  
 return new ResponseEntity<>("product not found", HttpStatus.*INTERNAL\_SERVER\_ERROR*);  
}  
  
  
  
}

Product NotFound Exception/////////////////////////////////////////////////////////////////

package com.example.springJava;  
  
public class ProductNotfoundException extends RuntimeException {  
 private static final long *serialVersionUID* = 1L;  
}

Topic////////////////////////////////////////////

package com.example.springJava;  
  
  
  
  
  
public class Topic {  
 private String id;  
 private String name;  
 private String description;  
  
  
 public Topic() {  
 }  
  
 public Topic(String id, String name, String description) {  
 this.id = id;  
 this.name = name;  
 this.description = description;  
 }  
  
 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;  
 }  
  
 public String getDescription() {  
 return description;  
 }  
  
 public void setDescription(String description) {  
 this.description = description;  
 }  
}

Topic Controller

package com.example.springJava;  
  
import org.hibernate.validator.constraints.EAN;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.Arrays;  
import java.util.List;  
  
@RestController  
public class TopicController {  
  
 @Autowired  
 private TopicService topicService;  
  
 @RequestMapping("/topics")  
 public List<Topic>getAllTopic(){  
 return topicService.getAllTopics();  
 }  
  
 @RequestMapping("/topics/{id}")  
 public Topic getTopic(@PathVariable String id){  
 return topicService.getTopic(id);  
  
 }  
  
 @RequestMapping(method = RequestMethod.*POST*,value = "/topics")  
 public void addTopic(@RequestBody Topic topic){  
 topicService.addTopic(topic);  
 }  
  
  
  
 @RequestMapping(method = RequestMethod.*PUT*,value = "/topics/{id}")  
 public void updateTopic(@RequestBody Topic topic, @PathVariable String id)  
  
 {  
 topicService.updateTopic(id,topic);  
 }  
  
  
  
 @RequestMapping(method = RequestMethod.*DELETE*,value = "/topics/{id}")  
 public void deleteTopic(@PathVariable String id){  
 topicService.deleteTopic(id);  
  
 }  
  
  
}

Topic Service///////////////////////////////////////////////////////

package com.example.springJava;  
  
import org.springframework.http.HttpStatus;  
import org.springframework.http.ResponseEntity;  
import org.springframework.stereotype.Service;  
  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.List;  
  
@Service  
public class TopicService {  
  
 private List<Topic> topics = new ArrayList<>(Arrays.*asList*(  
 new Topic("Spring", "Spring Framework", "Spring Framework Description"),  
 new Topic("java", "core java", "core java description"),  
 new Topic("javaScript", "javaScript", "javaScript Description")  
  
  
 ));  
  
 public List<Topic> getAllTopics() {  
 return topics;  
 }  
 //return topics.stream().filter(t -> t.getId().equals(id)).findFirst().get();  
  
 public Topic getTopic(String id) {  
 Topic topic = topics.stream().filter(t -> t.getId().equals(id)).findFirst().get();  
 if (topic==null){  
 throw new ProductNotfoundException();  
 }  
 return topic;  
  
 }  
  
  
 public void updateTopic(String id,Topic topic){  
 for (int i=0;i<topics.size();i++){  
  
 Topic t=topics.get(i);  
 if(t.getId().equals(id))  
 { topics.set(i,topic);  
  
 return;}  
 else if (!t.getId().equals(id))  
 { throw new ProductNotfoundException(); }  
 // ResponseEntity.notFound().build();}  
  
 }  
  
 }  
  
 public void addTopic(Topic topic) {  
 topics.add(topic);  
 }  
  
  
  
  
  
  
 public void deleteTopic(String id){  
 topics.removeIf(t-> t.getId().equals(id));  
  
  
  
 }  
  
  
  
  
  
}

////// Application properties ///////////////////////////////////

spring.datasource.url=jdbc:mysql://localhost:3306/cardb  
  
spring.datasource.username=monir  
spring.datasource.password=1234  
spring.jpa.show-sql = true  
  
  
spring.jpa.hibernate.ddl-auto=update  
  
#spring.jpa.hibernate.naming-strategy = org.hibernate.cfg.ImprovedNamingStrategy  
  
# Allows Hibernate to generate SQL optimized for a particular DBMS  
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect