Spring Boot Exception Handling ?

MainController :-

**package** com.ecommerce.controllers;

**import** org.springframework.http.HttpStatus;

**import** org.springframework.http.ResponseEntity;

**import** org.springframework.stereotype.Controller;

**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.ResponseBody;

**import** com.ecommerce.entity.EProduct;

**import** com.ecommerce.exceptions.ProductNotFoundException;

**@Controller**

**public** **class** MainController {

**@RequestMapping(value = "/product/{id}", method = RequestMethod.GET)**

**@ResponseBody**

**public** **String** getProduct(**@PathVariable("id")** **String** id) {

**if** (id.contentEquals("0"))

**throw** **new** ProductNotFoundException();

**return** "Product was found";

}

}

EProductExceptionController :-

**package** com.ecommerce.controllers;

**import** org.springframework.http.HttpStatus;

**import** org.springframework.http.ResponseEntity;

**import** org.springframework.web.bind.annotation.ControllerAdvice;

**import** org.springframework.web.bind.annotation.ExceptionHandler;

**import** com.ecommerce.exceptions.ProductNotFoundException;

**@ControllerAdvice**

**public** **class** EProductExceptionController {

**@ExceptionHandler(value = ProductNotFoundException.class)**

**public** ResponseEntity<**Object**> exception(ProductNotFoundException exception) {

**return** **new** ResponseEntity<>("Product not found", HttpStatus.NOT\_FOUND);

}

}

Create a project to Consume a RESTful web service ?

Quote :-

**package** com.ecommerce.beans;

**import** com.fasterxml.jackson.annotation.JsonIgnoreProperties;

**@JsonIgnoreProperties(ignoreUnknown = true)**

**public** **class** Value {

**private** **Long** id;

**private** **String** quote;

**public** Value() {

}

**public** **Long** getId() {

**return** **this**.id;

}

**public** **String** getQuote() {

**return** **this**.quote;

}

**public** void setId(**Long** id) {

**this**.id = id;

}

**public** void setQuote(**String** quote) {

**this**.quote = quote;

}

**@Override**

**public** **String** toString() {

**return** "Value{" +

"id=" + id +

", quote='" + quote + '\'' +

'}';

}

}

Ecommerce :-

**package** com.ecommerce.controllers;

**import** org.springframework.http.HttpStatus;

**import** org.springframework.http.ResponseEntity;

**import** org.springframework.stereotype.Controller;

**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.ResponseBody;

**import** org.springframework.web.client.RestTemplate;

**import** com.ecommerce.beans.Quote;

**@Controller**

**public** **class** MainController {

**@RequestMapping("/")**

**@ResponseBody**

**public** **String** index() {

RestTemplate restTemplate = **new** RestTemplate();

Quote quote = restTemplate.getForObject("https://gturnquist-quoters.cfapps.io/api/random", Quote.class);

**return** quote.toString();

}

}

Create a project to upload and download a file in Spring Boot ?

KafkaProducer :-

**package** com.ecommerce;

**import** java.util.HashMap;

**import** java.util.Map;

**import** org.apache.kafka.clients.producer.ProducerConfig;

**import** org.apache.kafka.common.serialization.StringSerializer;

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.context.annotation.Configuration;

**import** org.springframework.kafka.core.DefaultKafkaProducerFactory;

**import** org.springframework.kafka.core.KafkaTemplate;

**import** org.springframework.kafka.core.ProducerFactory;

**@Configuration**

**public** **class** KafkaProducerConfig {

**@Bean**

**public** ProducerFactory<**String**, **String**> producerFactory() {

**Map**<**String**, **Object**> configProps = **new** **HashMap**<>();

configProps.put(ProducerConfig.BOOTSTRAP\_SERVERS\_CONFIG, "localhost:9092");

configProps.put(ProducerConfig.KEY\_SERIALIZER\_CLASS\_CONFIG, StringSerializer.class);

configProps.put(ProducerConfig.VALUE\_SERIALIZER\_CLASS\_CONFIG, StringSerializer.class);

**return** **new** DefaultKafkaProducerFactory<>(configProps);

}

**@Bean**

**public** KafkaTemplate<**String**, **String**> kafkaTemplate() {

**return** **new** KafkaTemplate<>(producerFactory());

}

}

Wizard :-

**package** com.ecommerce;

**import** java.util.HashMap;

**import** java.util.Map;

**import** org.apache.kafka.clients.consumer.ConsumerConfig;

**import** org.apache.kafka.common.serialization.StringDeserializer;

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.context.annotation.Configuration;

**import** org.springframework.kafka.annotation.EnableKafka;

**import** org.springframework.kafka.config.ConcurrentKafkaListenerContainerFactory;

**import** org.springframework.kafka.core.ConsumerFactory;

**import** org.springframework.kafka.core.DefaultKafkaConsumerFactory;

**@EnableKafka**

**@Configuration**

**public** **class** KafkaConsumerConfig {

**@Bean**

**public** ConsumerFactory<**String**, **String**> consumerFactory() {

**Map**<**String**, **Object**> props = **new** **HashMap**<>();

props.put(ConsumerConfig.BOOTSTRAP\_SERVERS\_CONFIG, "localhost:2181");

props.put(ConsumerConfig.GROUP\_ID\_CONFIG, "group-id");

props.put(ConsumerConfig.KEY\_DESERIALIZER\_CLASS\_CONFIG, StringDeserializer.class);

props.put(ConsumerConfig.VALUE\_DESERIALIZER\_CLASS\_CONFIG, StringDeserializer.class);

**return** **new** DefaultKafkaConsumerFactory<>(props);

}

**@Bean**

**public** ConcurrentKafkaListenerContainerFactory<**String**, **String**> kafkaListenerContainerFactory() {

ConcurrentKafkaListenerContainerFactory<**String**, **String**>

factory = **new** ConcurrentKafkaListenerContainerFactory<>();

factory.setConsumerFactory(consumerFactory());

**return** factory;

}

}

MainController :-

**package** com.commerce.controllers;

**import** java.util.Calendar;

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

**import** org.springframework.kafka.core.DefaultKafkaProducerFactory;

**import** org.springframework.kafka.core.KafkaTemplate;

**import** org.springframework.kafka.core.ProducerFactory;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.web.bind.annotation.RequestMapping;

**@Controller**

**public** **class** MainController {

**@Autowired**

**private** KafkaTemplate<**String**, **String**> kafkaTemplate;

**@RequestMapping(value = "/")**

**public** **String** index() {

**this**.sendMessage("This is a message sent at " + **Calendar**.getInstance().getTime());

**return** "Check Eclipse console for kafka output";

}

**private** void sendMessage(**String** msg) {

kafkaTemplate.send("ecommerce", msg);

}

}

Ecommerce :-

**package** com.ecommerce;

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

**import** org.springframework.boot.ApplicationArguments;

**import** org.springframework.boot.ApplicationRunner;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.kafka.annotation.KafkaListener;

**import** org.springframework.kafka.core.KafkaTemplate;

**@SpringBootApplication**

**public** **class** SpringBootKafkaApplication {

**@Autowired**

**private** KafkaTemplate<**String**, **String**> kafkaTemplate;

**public** **static** void main(**String**[] args) {

SpringApplication.run(SpringBootKafkaApplication.class, args);

}

**@KafkaListener(topics = "ecommerce", groupId = "group-id")**

**public** void listen(**String** message) {

**System**.out.println("Received Message in group - group-id: " + message);

}

}

Create a project to enable HTTPS and display in browser ?

Wizard :-

**package** com.ecommerce.controllers;

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

**import** org.springframework.http.HttpStatus;

**import** org.springframework.http.ResponseEntity;

**import** org.springframework.stereotype.Controller;

**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.ResponseBody;

**@Controller**

**public** **class** MainController {

**@Autowired**

**private** ProductRepository repository;

**@RequestMapping("/")**

**@ResponseBody**

**public** **String** index() {

**return** “This is running under SSL”;

}

}

Application.properties :-

server.port=8443

server.ssl.key-alias=selfsigned\_localhost\_sslserver

server.ssl.key-password=changeit

server.ssl.key-store=classpath:ssl-server.jks

server.ssl.key-store-provider=SUN

server.ssl.key-store-type=JKS