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);

           }

}

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";

           }

}

package com.ecommerce.entity;

import java.math.BigDecimal;

import java.util.Collection;

import java.util.Date;

import java.util.List;

import java.util.Set;

import java.util.Map;

public class EProduct {

        private *long* ID;

        private String name;

        private BigDecimal price;

        private Date dateAdded;

        public EProduct() {

        }

        public *long* getID() {return this.ID; }

        public String getName() { return this.name;}

        public BigDecimal getPrice() { return this.price;}

        public Date getDateAdded() { return this.dateAdded;}

        public *void* setID(*long* *id*) { this.ID = *id*;}

        public *void* setName(String *name*) { this.name = *name*;}

        public *void* setPrice(BigDecimal *price*) { this.price = *price*;}

        public *void* setDateAdded(Date *date*) { this.dateAdded = *date*;}

}

package com.ecommerce.exceptions;

public class ProductNotFoundException extends RuntimeException {

           private static final *long* serialVersionUID = 1L;

}

package com.ecommerce;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringBootStarterApplication {

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

        SpringApplication.run(SpringBootStarterApplication.class, *args*);

    }

}

package com.ecommerce;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

class SpringBootStarterApplicationTests {

    @Test

*void* contextLoads() {

    }

}

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <parent>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-parent</artifactId>

        <version>2.6.3</version>

        <relativePath/> <!-- lookup parent from repository -->

    </parent>

    <groupId>com.ecommerce</groupId>

    <artifactId>SpringBootStarter</artifactId>

    <version>0.0.1-SNAPSHOT</version>

    <name>SpringBootStarter</name>

    <description>Demo project for Spring Boot</description>

    <properties>

        <java.version>11</java.version>

    </properties>

    <dependencies>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-web</artifactId>

        </dependency>

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-test</artifactId>

            <scope>test</scope>

        </dependency>

    </dependencies>

    <build>

        <plugins>

            <plugin>

                <groupId>org.springframework.boot</groupId>

                <artifactId>spring-boot-maven-plugin</artifactId>

            </plugin>

        </plugins>

    </build>

</project>