|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
| package com.example.springapp.model;  public class Product {      private long id;      private String name;      private double price;      public Product(){}      public Product(long id, String name, double price) {          this.id = id;          this.name = name;          this.price = price;      }      public long getId() {          return id;      }      public void setId(long id) {          this.id = id;      }      public String getName() {          return name;      }      public void setName(String name) {          this.name = name;      }      public double getPrice() {          return price;      }      public void setPrice(double price) {          this.price = price;      }      @Override      public String toString() {          return "Product [id=" + id + ", name=" + name + ", price=" + price + "]";      }  } |

|  |
| --- |
|  |
| package com.example.springapp.controller;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.http.ResponseEntity;  import org.springframework.web.bind.annotation.GetMapping;  import org.springframework.web.bind.annotation.PathVariable;  import org.springframework.web.bind.annotation.PostMapping;  import org.springframework.web.bind.annotation.RequestBody;  import org.springframework.web.bind.annotation.RestController;  import com.example.springapp.exceptions.\*;  import com.example.springapp.model.Product;  import com.example.springapp.service.ProductService;  @RestController  @RequestMapping(“/api/product”)  public class ProductController {        @Autowired      private ProductService service;      @PostMapping      public ResponseEntity<?> addProduct(@RequestBody Product p) throws ProductExistsException {          Product product=service.addProduct(p);          return ResponseEntity.status(201).body("Successfully added the product.");      }        @GetMapping("/{id}")      public ResponseEntity<?> getProductById(@PathVariable int id) throws Exception {          Product product=service.getProductById(id);          return ResponseEntity.status(200).body(product);      }  } |
|  |

|  |
| --- |
|  |
| package com.example.springapp.exceptions;  import org.springframework.http.HttpStatus;  import org.springframework.http.ResponseEntity;  import org.springframework.web.bind.annotation.ControllerAdvice;  import org.springframework.web.bind.annotation.ExceptionHandler;  @ControllerAdvice  public class ProductExceptionHandler {      @ExceptionHandler(ProductExistsException.class)      public ResponseEntity<String> handleCustomException(ProductExistsException e){          return ResponseEntity.status(400).body(e.getMessage());      }      @ExceptionHandler(ProductNotFoundException.class)      public ResponseEntity<String> handleCustomException(ProductNotFoundException e){          return ResponseEntity.status(404).body(e.getMessage());      }  } |

|  |
| --- |
|  |
| package com.example.springapp.exceptions;  public class ProductExistsException extends Exception{      public ProductExistsException(){          super();      }      public ProductExistsException(String msg){          super(msg);      }  } |

|  |
| --- |
|  |
| package com.example.springapp.exceptions;  public class ProductNotFoundException extends Exception{      public ProductNotFoundException(){          super();      }      public ProductNotFoundException(String msg){          super(msg);      }  } |

|  |
| --- |
|  |
| package com.example.springapp.service;  import java.util.ArrayList;  import java.util.List;  import org.springframework.stereotype.Service;  import com.example.springapp.model.Product;  import com.example.springapp.exceptions.ProductExistsException;  import com.example.springapp.exceptions.ProductNotFoundException;  @Service  public class ProductService {        List<Product> list=new ArrayList<>();      public Product addProduct(Product p) throws ProductExistsException {          for(Product x: list){              if(x.getId()==p.getId()){                  throw new ProductExistsException("Invalid product details.");              }          }       list.add(p);       return p;      }      public Product getProductById(int id) throws Exception {          boolean found=false;          Product prod=null;          for(Product x: list){              if(x.getId()==id){                  found=true;                  prod=x;                  break;              }          }          if(!found){              throw new ProductNotFoundException(null);          }          return prod;      }  } |

    public Product addProduct(Product p) throws ProductExistsException {

id=p.getProdId();

       Product prod=repo.findById(id).orElse(null);

If(prod!=null){

                throw new ProductExistsException("Invalid product details.");

            }

        }

     Repo.save(p);

     return p;

    }

@RequestMapping(“/order”)

@GetMapping(“/id”)