

1.Spring boot exception handling

EProductExceptionHandler.java:

```
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 EProductExceptionHandler {

    @ExceptionHandler(value = ProductNotFoundException.class)
    public ResponseEntity<Object> exception(ProductNotFoundException
exception) {
        return new ResponseEntity<>("Product not found",
HttpStatus.NOT_FOUND);
    }
}
```

MainController.java:

```
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";
    }
}
```

EProduct.java:

```
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;}

}
```

ProductNotFoundException.java:

```
package com.ecommerce.exceptions;

public class ProductNotFoundException extends RuntimeException {
    private static final long serialVersionUID = 1L;

}
```