Exception Handling in Spring Rest

Sometimes we know that we will be getting the error by putting some code in our application but we need that code so in that case we can declare exceptions and handle that exception in our code. Then we will not get the error and our code will also be used in the application.

Exceptions will be handled by the rest controller by creating a method in that class.

Create fields (status, message, timestamp) for exception POJO class with constructors using fields and generate getter and setter methods.

Create a class for specific exceptions (CustomerNotFoundException) and change its superclass to RuntimeException, super() will automatically inherit its parent class of RuntimeException extension.

While getting the data from DB if there is no data so make an exception logic in controller method that if id equals null then throw CustomerNotFoundException with the custom error message.

While deleting the data if id is not found then make an exception logic in controller method that if id equals null then throw CustomerNotFoundException with the custom error message.

Now, if a user enters a String value which is not supported then it will throw an error so in that case create a global exception class for handling all kinds of errors.

use @ControllerAdvice in global exception class and define the methods like this: -

```
@ExceptionHandler
public ResponseEntity<CustomerErrorResponse>
handleException(CustomerNotFoundException exc){

CustomerErrorResponse error = new CustomerErrorResponse(
HttpStatus.NOT_FOUND.value(),
exc.getMessage(),
System.currentTimeMillis());
return new ResponseEntity<>(error, HttpStatus.NOT_FOUND);
}

@ExceptionHandler
public ResponseEntity<CustomerErrorResponse> handleException(Exception exc){

CustomerErrorResponse error = new CustomerErrorResponse(
HttpStatus.BAD_REQUEST.value(),
exc.getMessage(),
System.currentTimeMillis());
return new ResponseEntity<>(error, HttpStatus.BAD_REQUEST);
}
```