

-----  
Email: javabyraghu@gmail.com  
FB: <https://www.facebook.com/groups/thejavatemple>

1. `ExceptionHandler` with path `/error`
  - | - `ExceptionHandler`
  - | - `ExceptionHandler`
    - `errorHtml()`
    - `error()`
2. `ExceptionHandlerAttributes`
3. Custom Error Pages
  - `error.html` page under `/templates` folder
  - `404.html`
  - `500.html`
  - `5xx.html` with `@ResponseStatus(HttpStatus.NOT_IMPLEMENTED)`
  - Custom Exception
4. Custom `ExceptionHandler`
5. Using Error Attributes in Custom Controller
6. Return Error Response as JSON Type
7. Customize `ExceptionHandlerAttributes` (extends `ExceptionHandlerAttributes`)
8. `** Rest @ExceptionHandlerAdvice` and `@ExceptionHandler(T.class)`

=====

### Spring Boot (MVC and ReST)

=====

Work flow (on success)

Client -> Request -> `DispatcherServlet` -> `doDispatch(req,resp)` -> HM -> Controller  
DS -> `ViewResolver` -> View -> Model (read) -> DS -> UI

Work flow (on failure)

Client -> Request -> `DispatcherServlet` -> `doDispatch(req,resp)` -> HM -> Controller  
-> DS -> `ExceptionHandler` (`error()`/`errorHtml()`) -> Response

Q) Is Spring/Spring Boot handling Errors Internally? How?

A) Yes. By using `ExceptionHandler(I)` which has on impl class `ExceptionHandler(C)`.  
that provides 2 methods

`errorHtml(): ModelAndView` (call made from browser)  
`error(): ResponseEntity<T>` (call from non-browser/client app/postman tool)

`ExceptionHandlerAttributes`:-

`ExceptionHandler` collects all error details as `Map<String, Object>`  
which holds details like 'status, error, path, timestamp..etc'.

This `ExceptionHandler` only called as `ExceptionHandlerAttributes`, this is given as final respons to client.

-----

### 3. Custom Error Pages

-----

#1. `error.html` page under `/templates` folder

We can create `error.html` file under 'templates' folder and design based on requirement. This gets executed on any problem (Http Status : 4xx and 5xx)

ie for 400, 401 , 403 , 404 , .. 500 , 501 ..etc only one page is executed : `error.html`

#2. Specific Error page create under "error" folder with ErrorCode.html format  
ex: 404.html  
500.html

If given code is not matche with above pages then again back to error.html

#3 Page with Generic

4xx.html <-- 400 , 401 , 402 ,...<-----

5xx.html

with @ResponseStatus(HttpStatus.NOT\_IMPLEMENTED)  
Custom Exception

=> If controller is throwing any exception then default HTTP Staus is -500  
INTERNAL SERVER ERROR. To change error code to any other number use  
@ResponseStatus(HttpStatus.NOT\_IMPLEMENTED) (on top of your custom exception).

=====CODE FILES=====

Dependecnies: web, devtools, thymeleaf

\*) Custom Exception

```
package in.nareshit.raghu.controller;
```

```
import org.springframework.http.HttpStatus;
```

```
import org.springframework.web.bind.annotation.ResponseStatus;
```

```
@ResponseStatus(code = HttpStatus.NOT_IMPLEMENTED)
```

```
public class NotImplementedException extends RuntimeException {
```

```
    /**
```

```
     *
```

```
    */
```

```
    private static final long serialVersionUID = 1L;
```

```
    public NotImplementedException() {  
        super();  
    }
```

```
    public NotImplementedException(String message) {  
        super(message);  
    }
```

```
}
```

=====

```
package in.nareshit.raghu.controller;
```

```
import org.springframework.http.HttpStatus;
```

```
import org.springframework.web.bind.annotation.ResponseStatus;
```

```
@ResponseStatus(code = HttpStatus.NOT_FOUND)
```

```
public class UserNotFoundException extends RuntimeException {
```

```
    /**
```

```
     *
```

```
    */
```

```
    private static final long serialVersionUID = 1L;
```

```
    public UserNotFoundException() {  
        super();  
    }
```

```
    public UserNotFoundException(String message) {  
        super(message);  
    }
```

```
}
```

```

}
=====
*)Controller
package in.nareshit.raghu.controller;

import java.util.Random;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class EmployeeController {

    @GetMapping("/showa")
    public String showA() {
        if(new Random().nextInt(10)<=10)
            throw new RuntimeException("SAMPLE");
        return "Welcome";
    }

    @GetMapping("/showb")
    public String showB() {
        if(new Random().nextInt(10)<=10)
            throw new NotImplementedException("SAMPLE");
        return "Welcome";
    }

    @GetMapping("/showc")
    public String showC() {
        if(new Random().nextInt(10)<=10)
            throw new UserNotFoundException("SAMPLE");
        return "Welcome";
    }
}

*) UI PAGES
error.html
<!DOCTYPE html>
<html xmlns:th="https://www.thymeleaf.org/">
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" >
</head>
<body>
<div class="container">
<h3>SOMTHING WENT WRONG! PLZ CONTACT ADMIN</h3>
<table class="table table-bordered">
    <tr>
        <td>DATE</td>
        <td th:text="${timestamp}"></td>
    </tr>

    <tr>
        <td>status</td>
        <td th:text="${status}"></td>
    </tr>

    <tr>
        <td>error</td>
        <td th:text="${error}"></td>
    </tr>

    <tr>

```

```

        <td>path</td>
        <td th:text="${path}"></td>
    </tr>

    <tr>
        <td>Trace</td>
        <td th:text="${trace}"></td>
    </tr>
</table>
</div>
</body>
</html>

```

```

-----
400.html
<!DOCTYPE html>
<html xmlns:th="https://www.thymeleaf.org/">
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<H3>NOT FOUND PROBLEM, CHECK AGAIN</H3>

</body>
</html>

```

```

-----
5xx.html
<!DOCTYPE html>
<html xmlns:th="https://www.thymeleaf.org/">
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" >
</head>
<body>
<div class="container">
<h3>FOR 5xx ONLY</h3>
<table class="table table-bordered">
    <tr>
        <td>DATE</td>
        <td th:text="${timestamp}"></td>
    </tr>

    <tr>
        <td>status</td>
        <td th:text="${status}"></td>
    </tr>

    <tr>
        <td>error</td>
        <td th:text="${error}"></td>
    </tr>

    <tr>
        <td>path</td>
        <td th:text="${path}"></td>
    </tr>

    <tr>
        <td>Trace</td>
        <td th:text="${trace}"></td>
    </tr>
</table>

```

```
</div>
</body>
</html>
```

```
-----
Welcome.html
```

```
<!DOCTYPE html>
<html xmlns:th="https://www.thymeleaf.org/">
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h3>WELCOME TO UI PAGE!!</h3>
</body>
</html>
```