



[Home](#) » [Spring](#) » Spring MVC – Code ví dụ xử lý Exception, Exception Handling Spring MVC

Spring MVC – Code ví dụ xử lý Exception, Exception Handling Spring MVC

Posted on Tháng Mười Hai 28, 2017

Spring MVC – Code ví dụ xử lý Exception, Exception Handling Spring MVC

Trong Spring có khá nhiều cách xử lý exception khác nhau, xử lý exception theo loại exception, theo error code, xử lý chung theo class, theo package hay cho cả project.

Trong bài trước về [Spring Form Validation](#) chúng ta xử lý validate object bằng BindingResult ở mỗi method, bạn hoàn toàn có thể gộp chung tất cả các exception validate form vào chung một xử lý.

Dưới đây mình sẽ giới thiệu một số cách xử lý exception khác nhau, các bạn có thể tìm hiểu để áp dụng vào từng trường hợp của mình.

Lưu ý: để xử lý global exception cần khai báo `<mvc:annotation-driven />` trong file spring config .xml

1. Xử lý Exception sử dụng HTTP Status Codes

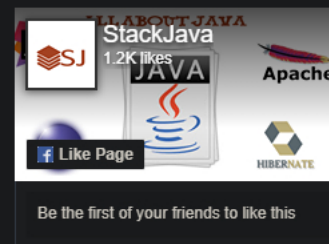
Sử dụng annotation `@ResponseStatus` trước class định nghĩa exception sẽ chỉ dẫn HTTP Code trả về khi exception này xảy ra.

Ví dụ:

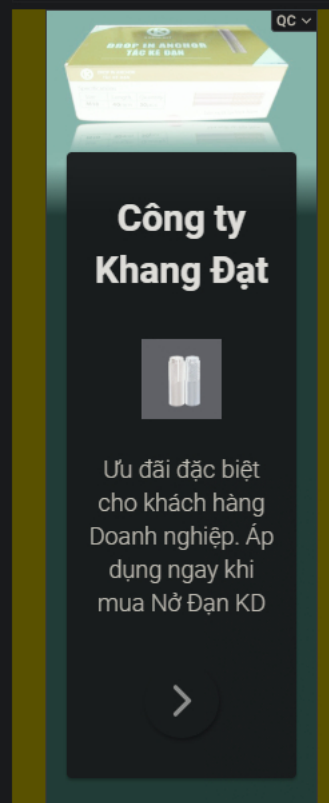
```
@ResponseStatus(value = HttpStatus.UNAUTHORIZED, reason = "User not logged in") // 401
public class UnauthorizedException extends RuntimeException {
    private static final long serialVersionUID = 1L;
}
```

```
@RequestMapping(value = "/unauthorizedException")
public String testUnauthorizedException() throws IOException {
    throw new UnauthorizedException();
}
```

url `/unauthorizedException` ném ra exception `UnauthorizedException`, exception `UnauthorizedException` sẽ trả về trang error 401 với message là "User not logged in".

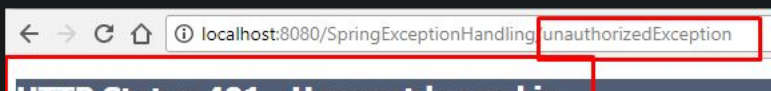


Tìm kiếm ...



SPRING MVC

Thymeleaf là gì? So sánh JSP, JSE



HTTP Status 401 - User not logged in	
type	Status report
message	User not logged in
description	This request requires HTTP authentication.
Apache Tomcat/8.0.45	

Khi xảy ra exception **UnauthorizedException** ứng dụng web sẽ xử lý đúng như trường hợp xảy ra lỗi 401.

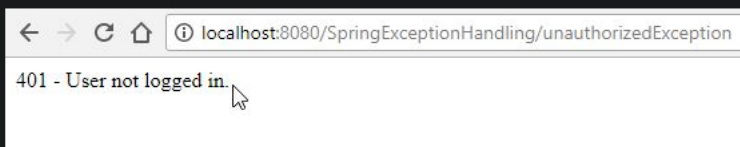
Ví dụ mình config error page cho 401 là trang 401.jsp

```
<error-page>
  <error-code>401</error-code>
  <location>/WEB-INF/error-pages/401.jsp</location>
</error-page>
```

401.jsp

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>401</title>
</head>
<body>
401 - User not logged in.
</body>
</html>
```

Kết quả:



2. Xử lý exception dựa trên Controller

Sử dụng Annotation @ExceptionHandler

Bạn có thể định nghĩa các method xử lý lỗi bằng cách thêm annotation `@ExceptionHandler` trước method đó để chỉ dẫn rằng method đó sẽ xử lý cho exception nào.

Các Exception được ném ra từ các method được đánh dấu `@RequestMapping` sẽ được các method đó xử lý.

Với cách này bạn có thể trả về view tùy ý khi exception xảy ra.

Ví dụ:

với Thymeleaf

CHUYÊN MỤC

Algorithm

Apache

Apache JMeter

Apache Kafka

AWS

C/C++

CDI

Clean Code

Demo

Design Pattern

Docker

Eclipse

Elasticsearch

Excel

FAQ

Framework

Freemaker

FreeMarker

Gradle

Hibernate

HttpClient

HttpComponents

Install

IntelliJ IDEA

Java

Java Basic

Java Core

```

@ExceptionHandler({NullPointerException.class})
public String processNullPointerException() {
    return "null-pointer-exception";
}

@RequestMapping(value = "/nullPointerException")
public String testNullPointerException() throws NullPointerException {
    throw new NullPointerException("this is null pointer exception");
}

```

Method **processNullPointerException** sẽ xử lý tất cả các exception **NullPointerException** ném ra từ các **@RequestMapping** trong cùng class.

Url **"/nullPointerException"** ném ra **NullPointerException** nên sẽ được xử lý bởi method **processNullPointerException**

null-pointer-exception.jsp

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>NullPointerException</title>
</head>
<body>
<h1>null-pointer-exception.jsp</h1><br/>
    Null pointer exception
</body>
</html>

```

Kết quả:



Sử dụng Annotation **@ControllerAdvice**

Khi sử dụng annotation **@ControllerAdvice** cho class, thì các method được đánh dấu **@ExceptionHandler** có thể xử lý exception cho cả các method của các class khác, package khác.

```

// Xử lý Exception cho tất cả các class
@ControllerAdvice
public class GlobalExceptionHandler {
    //...
}

```

Java8

JavaScript

jooq

JSF

JSP-Servlet

JUnit

Library

Linux

Maven

MongoDB

MySQL

Network Programming

Node.js

OOP

PostgreSQL

PrimeFaces

Principle

Python

quartz

Redis

SDKMan

Security

SocketCluster

Spring

Spring Boot

Spring Core

Spring Data

Spring Hibernate

Spring JDBC

```
// Xử lý các exception cho các class trong package:
@ControllerAdvice(basePackages="stackjava.com.springexceptionhandling.controller")
public class GlobalExceptionHandler {
    //...
}
```

Ví dụ:

```
@ControllerAdvice
public class GlobalExceptionHandler {

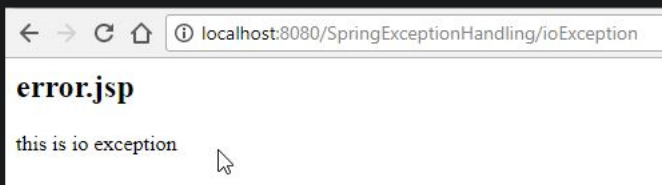
    @ExceptionHandler(IOException.class)
    private ModelAndView processIOException(IOException ex) {
        ModelAndView model = new ModelAndView("error");
        model.addObject("error", ex.getMessage());
        return model;
    }
}
```

```
@RequestMapping(value = "/ioException")
public String testIOException() throws IOException {
    throw new IOException("this is io exception");
}
```

error.jsp

```
<html>
<head>
<title>Error</title>
</head>
<body>
    <h2>error.jsp</h2>
    ${error}
</body>
</html>
```

Kết quả:



Ví dụ xử lý exception khi validate form:

User.java

```
public class User {
```

Spring MVC

Spring Security

Thymeleaf

Tomcat

Uncategorized

Web Service

WebSocket

Wordpress



```

@NotNull(message = "Id is required")
private Integer id;

@NotBlank(message = "Name is required")
@Length(min = 5, max = 10)
private String name;

@NotBlank(message = "Email is required")
@email
private String email;

//getter/setter
}

```

```

@ExceptionHandler(org.springframework.validation.BindException.class)
private ModelAndView processInvalidData(BindException ex) {
    StringBuilder error = new StringBuilder();
    for (ObjectError objectError: ex.getAllErrors()) {
        error.append(objectError.getDefaultMessage());
        error.append("<br/>");
    }
    ModelAndView model = new ModelAndView("error");
    model.addObject("error", error);
    return model;
}

```

```

@RequestMapping(value = "/addUser", method = RequestMethod.POST)
public String doPostAddUser(@ModelAttribute("user") @Validated User user) {
    return "view-user";
}

```

*Lưu ý: trong method doPostAddUser sẽ không có tham số BindingResult sau tham số được đánh dấu @Validate/@Valid

Kết quả:

add-user.jsp

Id:

Name:

Email:

localhost:8080/SpringExceptionHandler/addUser

error.jsp

length must be between 5 and 10
Name is required
Id is required



Code ví dụ xử lý Exception, Exception Handling Spring MVC

Okay, Done!

Download code tất cả các ví dụ trên [tại đây](#)

References: <https://spring.io/blog/2013/11/01/exception-handling-in-spring-mvc>

This entry was posted in [Spring](#), [Spring MVC](#) and tagged [spring](#), [spring mvc](#). Bookmark the [permalink](#).

← [Spring MVC FAQ: Phân biệt sự khác nhau giữa component-scan, annotation-config, annotation-driven](#)

[Hướng dẫn lập trình Python trên Eclipse với plugin PyDev](#)



Học Lập Trình Web PHP Từ A-Z - Viện Công
Nghệ Và Đào Tạo DEV

VIỆN CÔNG NGHỆ VÀ ĐÀO TẠO DEVMASTER

Học Lập Trình Web PHP Từ Cơ Bản Đến Chuyên Sâu,
Code Trên Dự Án Thực Tế Tại DN, Ưu Đãi 45%

TRANG WEB

CHỈ ĐƯỜNG

Laptop Asus S530FN-
BQ283T (Grey)-...

21.990.000đ
- Bộ VXL: Core i7 8565U
1.8Ghz-8Mb - Cạc đồ
họa: Nvidia MX150 2GB
GDDR5 - Bộ nhớ:....
phucanh.vn

Học Lập Trình Web PHP Từ A-Z - Viện Công
Nghệ Và Đào Tạo DEV

VIỆN CÔNG NGHỆ VÀ ĐÀO TẠO DEVMASTER

Học Lập Trình Web PHP Từ Cơ Bản Đến Chuyên Sâu,
Code Trên Dự Án Thực Tế Tại DN, Ưu Đãi 45%

TRANG WEB

CHỈ ĐƯỜNG

