5. Handling Exceptions in JSP

Contents

[1. Introduction 1](#_Toc26700)

[2. Demo: Handling Exceptions in JSP 2](#_Toc8588)

[3. Summary 3](#_Toc28671)

# 1. Introduction

=>slides: Pg. 1

 In the last module we have understood JSP built-in objects and in this module we shall understand how to handle exception in JSP and then we shall extend our virtual training company application to handle the exceptions. Well, what is an exception and why do we need to know about exception handling in JSP?

=>slides: Pg. 2

Exceptions are just a way to indicate that there is an error, where an exception handling is an error handling mechanism that is built-in into the language.

=>slides: Pg. 3

We may encounter with different types of exceptions while writing the JSP code. Checked exceptions, unchecked exceptions, error. Checked exceptions is an exception that is typically a user error or a problem that cannot be forcing by the programmer. For example, if a file is to be open, but the file cannot be found, an exception occurs. These exceptions cannot simply be ignored at the time of compilation. Runtime exceptions, a runtime exception is an exception that occurs that probably would have been avoided by the programmer, as opposed to checked exceptions, runtime exceptions are ignored at the time of compilation. Errors are typically ignored in your code because you can rarely do anything about an error. For example, if a stack overflow occurs an error will raise. They are also ignored at the time of compilation. These exceptions may occur anytime in our web application, so handling exception is a safer side for the web developer because it is quite obvious that we don't want to show an error stack trace to any random user surfing our web application.

=>slides: Pg. 4

Exception handling in JSP can be achieved using three ways, using simple try-catch block, using isErrorPage and ErrorPage attribute of page directive, using error-page tag in deployment descriptor. =>slides: Pg. 5

Now first let us understand how do we use try-catch block. Let us say we have the following JSP code which will divide the number by 0. When we execute the application, we know that it will raise an exception, so to handle this exception the easiest way is we can surround the lines of code which may raise an exception using the try-catch block. So I can type in try and then open the flower brackets and then we can close the brackets and we can just add the catch block by typing catch of Exception variable and then the statements to handle the exception, for example, out. println or Error +e. getMessage. So when we execute the application we can observe the error message specified using the out built-in object. But when we develop a JSP application, it is advisable not to use the traditional error handling within the JSP page inside a scriptlet. Instead it is advisable to use the isErrorPage and ErrorPage attributes of the page directive. Now let us understand how to use the isErrorPage and ErrorPage attribute of page directive.

=>slides: Pg. 6

Let us assume that we wanted to have a JSP page as an ErrorPage, for example, with the name error. jsp. Then once we have designed the page with the error description to be shown, then we can just add a page directive with the attribute isErrorPage=true where isErrorPage is used to officially designate the JSP page as an ErrorPage and within the JSP page, which might rise an exception, we need to use the page directive with the attribute ErrorPage= the name of the ErrorPage, for example, errorPage=error. jsp. Where the ErrorPage attribute in the page directive informs the web container that if an exception occurs in the current page forward the request to the specified ErrorPage. This is the preferred method to handle JSP errors, but when there are some common exceptions--, such as file not found or if you wanted to handle the exceptions at the application level, then we can handle using the ErrorPage tag within the deployment descriptor file. We can even configure different error pages for different exception types or HTTP error codes.

=>slides: Pg. 7

For example, in order to declare an error page for all the types of exception, we can just define error-page exception-type Java. lang. Throwable, location the URL of the error page.

=>slides: Pg. 8

In order to declare an error page for more detailed exception, we can just define error-page exception-type Java. lang. ArithmeticException, location the URL of the error page.

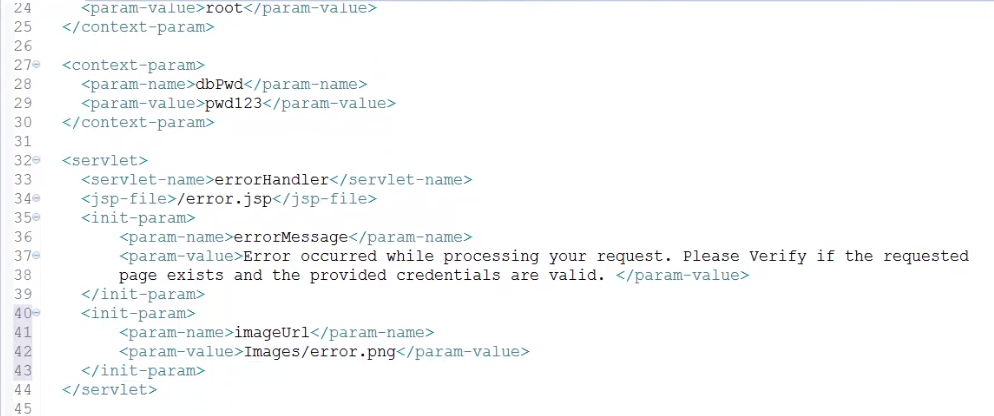
=>slides: Pg. 9

We can also declare an error page based on HTTP error status codes. To do, we can define error-page error-code 404, location URL of the error-page. If you'd like to request a page which is not present we can observe the error description from the error-page what we created. Whenever we handle the exceptions at the web deployment descriptor and also using the page directive, the priority will be given to the page directive exception handling only. In the next section let us understand how to extend our virtual training company application to handle the exceptions practically

# Demo: Handling Exceptions in JSP

=>slides: Pg. 10

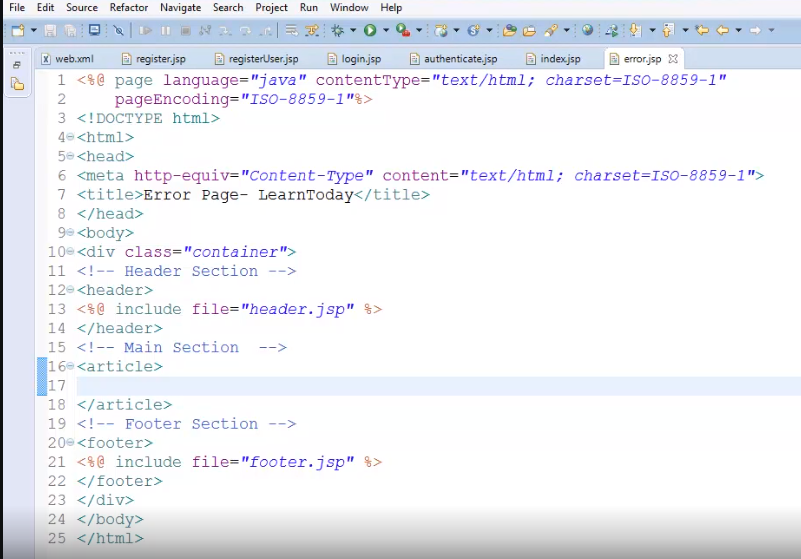
In the last module we have observed that since there was no courses. jsp page in our application, it has thrown an exception and we got the default error page from the server and there will be many scenarios where we might get an exception, so it is advisable to handle the exceptions globally.



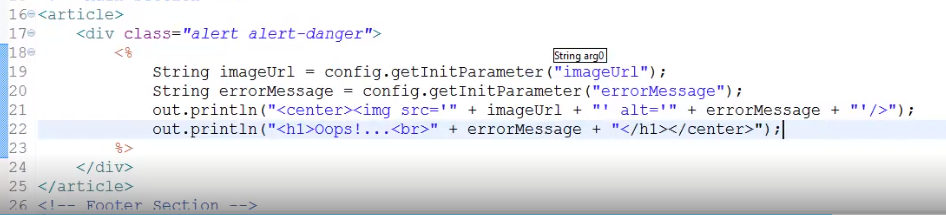
To do, let me open the web. xml file to configure the error page globally. So let me type in servlet, then we need to provide the servlet name, so let me type in servlet-name errorHandler and then we need to specify the JSP file, so let me type in jsp-file/error. jsp. I would like to provide a custom error page for all the exceptions that are raised within our application. So let me pass some parameters for the JSP page. To do, let me type in init-param, then we need to provide the param name. For example, let me type in errorMessage and then we wanted to provide the parameter value. To do let me type in param-value and let me paste a custom error message to be displayed for the client. Once again, let me add one more parameter to display the error image. I have taken the liberty to copy an image for the error with the name error. png to the images folder. Now let me type in init-param, param-name imageUrl and param-value Images/error. png.



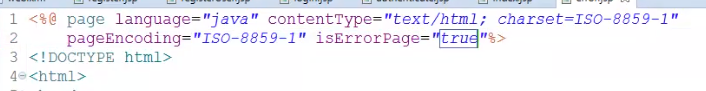
Now let us add the servlet mapping entry in web. xml file. To do, let me type in servlet-mapping, servlet-name errorHandler, url-pattern/error. jsp. We are already aware that we can configure the error page based on the error code also. To understand let me type in error-page, then we need to provide the error-code, for example, 404, and then we need to specify the location of the error-page using location/error. jsp. Now let me save the web. xml file.



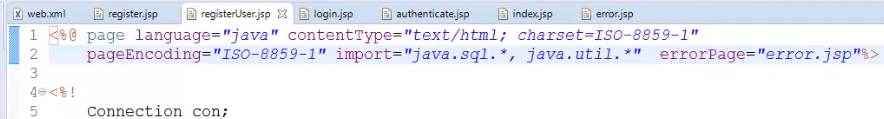
Now let us add the JSP error page with the name error. jsp. I have taken the liberty to add the file and updated the title for the page as Error Page- LearnToday. And also, I have included the header and footer JSP files to provide consistent experience for the user. Now within the article section let us read the values from the web. xml file and display that content,



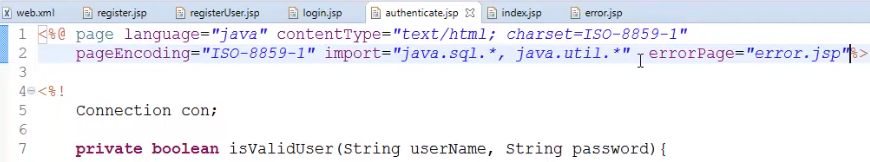
so let me type in div class=alert alert-danger, which displays a block in light red color. Let me add a scriptlet then let us collect the value of the image URL and the error message. To do let me type in String imageUrl = config. getInitParameter of imageUrl. String errorMessage = config. getInitParameter of errorMessage. Now in order to display these values, we don't have to take a separate expression. We can use the out built-in object. So let me type in out. println of center impingement src = imageUrl + alt= errorMessage and let me add one more out statement to display the error message. So let me type in out. println of, let me add a header h1 Oops!, and then to display the error message let me type in +errorMessage. Since this page is an error page, we need to update our page directive with the isErrorPage attribute.



So let me type in isErrorPage=true at the page directive. Once we have created the custom error page, then we need to update the JSP pages which has to use this custom error page. To do let me open the JSP pages, which we have already created and we need to update the page directive using errorPage=error. jsp.



So let me open the registerUser. jsp file and let me type in errorPage=error. jsp.



And let me take the liberty to updating other JSP pages also. I have updated the other JSP pages also to use errorPage=error. jsp. Now let us execute our application. Let me click on the login link and let me provide the username as Sekhar and the password as Srinivasan, and let me click on the Login. Now we can observe the custom error page, what we have created is displayed.

# Summary

=>slides: Pg. 11

In this module we have understood how to handle the exceptions in JSP page. In the next module we shall understand how to work with the standard JSP actions and we shall extend our virtual training company application by using the standard actions provided by the JSP to search for a specific course and we shall display that course with the support of JavaBeans and standard actions. Let us meet in the next module to understand and extend our application.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*