



검색

카페정보

나의활동

매니저 자바샘

since 2021.05.22.

카페소개

씨앗3단계

7

초대하기

즐거찾는 멤버

0명

게시판 구독수

0회

우리카페앱 수

0회

주제

컴퓨터/통신 > 프로그래밍 언어

지역

서울특별시 서초구

카페 글쓰기

카페 채팅

즐거찾는 게시판

전체글보기

72

공지사항

수업내용

수업동강

자료실

최근 댓글 · 답글

최근 댓글 · 답글이 없습니다.

카페탈퇴하기

궁금한게 있을 땐
카페 스마트봇

수업내용 >

에러(error) 처리

자바샘 카페마니저

1:1 채팅

2021.06.05. 09:31

조회 7

댓글 0 URL 복사

첨부파일 모아보기 1

1. 에러 처리 페이지

1) 에러처리 페이지 지정

: 에러를 전문적으로 처리할 페이지에 아래와 같은 page 지시자(directive)를 작성합니다.

```
<%@ page isErrorPage = "true" %>
```

2) 에러처리 페이지 호출

```
<%@ page errorPage = "에러 발생시 이동할 JSP" %>
```

3) web.xml 에서의 광역적(전역적) 에러 페이지 설정

```
<error-page>
  <error-code>에러 코드</error-code>
  <location>에러 페이지</location>
</error-page>

<error-page>
  <exception-type>예외처리 클래스</exception-type>
  <location>에러 페이지</location>
</error-page>
```

ex)

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">

  ... (중략) ...

  <error-page>
    <exception-type>java.lang.Throwable</exception-type>
    <location>/common/error.isp</location>
```

```

</error-page>

<error-page>
  <error-code>404</error-code>
  <location>/common/error.jsp</location>
</error-page>

<error-page>
  <error-code>500</error-code>
  <location>/common/error.jsp</location>
</error-page>
</web-app>

```

2. 에러 처리 우선순위(priority)

- 1) page 지시자(directive) errorPage 속성으로 지정한 에러 페이지
- 2) web.xml의 <exception-type>에서 지정한 예외처리 클래스 발생시
- 3) web.xml의 <error-code>에서 지정한 HTTP 상태 코드(status code) 상황 발생시
- 4) WAS(Tomcat)가 제공하는 자동 에러 처리 검출에 감지될 경우

3. 예외처리 예시

1) web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
id="WebApp_ID" version="4.0">
  <display-name>error_page</display-name>

  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>

  <error-page>
    <error-code>404</error-code>
    <location>/error/404.jsp</location>
  </error-page>

  <error-page>
    <error-code>500</error-code>
    <location>/error/500.jsp</location>
  </error-page>

  <error-page>
    <exception-type>java.lang.NullPointerException</exception-type>
    <location>/error/NullPointerException.jsp</location>
  </error-page>

</web-app>

```

2) JSP : 예외처리 페이지

```

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="UTF-8">
</head>
<body>

```

```

<b>해당 페이지가 없습니다(404)</b>
</body>
</html>

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="UTF-8">
<title>웹서버 문제</title>
</head>
<body>
<b>웹서버에 문제가 있습니다</b>
</body>
</html>

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="UTF-8">
<title>NullPointerException</title>
</head>
<body>
<b>NullPointerException 오류가 발생했습니다.</b>
</body>
</html>

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page isErrorPage="true" %>
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="UTF-8">
<title>에러 처리 메시지 페이지</title>
</head>
<body>

<hr>
에러 타입: <%= exception.getClass().getName() %> <br>
에러 메시지: <b><%= exception.getMessage() %></b>

</body>
</html>

```

3) Servlet : 테스트시 서블릿을 바로 실행해봅니다.

```

package servlet;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class DemoServlet
 */
@WebServlet("/DemoServlet")
public class DemoServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

```

```

/**
 * @see HttpServlet#HttpServlet()
 */
public DemoServlet() {
    // TODO Auto-generated constructor stub
}

/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

    request.setCharacterEncoding("UTF-8");
    response.setCharacterEncoding("UTF-8");
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();

    // 아래의 두가지 경우에 대해서 각각 예외처리를 확인합니다.

    // 1)
    /*
    if (request.getParameter("name")==null)
        throw new NullPointerException();
    else {
        String name = request.getParameter("name");
        out.println("인자(name) : "+name);
    }
    */

    // 2)
    String name = request.getParameter("name")==null ?
        null : request.getParameter("name");

    name = name.toString();
    out.println("인자(name) : "+name);

} //

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    doGet(request, response);
}
}

```

4) JSP

```

<%@ page contentType="text/html; charset=UTF-8" %>
<%--
    여기서 buffer 크기를 1kb로 했을 경우와 10kb로 했을 때의 차이를 비교해봅니다.
--%>
<%@ page buffer="1kb" %>
<%@ page errorPage="/error/viewError.jsp" %>
<!doctype html>
<html lang="ko">
<head>
<title>버퍼 플러시(buffer flush) 이후 예외 발생</title>
</head>
<body>

```

```

<%
    for (int i=0; i<256; i++) {
        out.println(i);
    }
%>

<!-- divide by zero -->
<%= 1 / 0 %>

</body>
</html>

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="ko">
<head>
<meta charset="UTF-8">
<title>NullPointerException 상황</title>
</head>
<body>
<%
    // Object obj; // 이런 경우는 500 에러 발생
    Object obj = null; // 이런 경우는 NullPointerException 발생
    // out.println(obj); // 이런 경우는 NullPointerException 감지 안됨
    out.println(obj.toString()); // 이런 경우는 NullPointerException 감지됨 !
%>
</body>
</html>

<%@ page errorPage="/error/viewError.jsp" %>
<%@ page contentType="text/html; charset=UTF-8" %>
<!doctype html>
<html lang="ko">
<head>
<title>NullPointerException 상황 : 인자 출력</title>
</head>
<body>
<%
    // 안타깝게도 이럴 경우에는 NullPointerException 에러 감지가 안됩니다 !
    String name = request.getParameter("name");
    out.println("name 의 자료형 : " +
        (name instanceof Object ? "문자열(String)" : "문자열(String)이 아님") + "<br>");
%>
    인자 값(name) : <%= name %>

</body>
</html>

<%--
    아래의 errorPage를 활성화하면 예외처리시
    web.xml에 표기된 아래의 사항 보다 우선적으로 처리됩니다. 번갈아 가면서 실행해봅시다.

    <error-page>
        <exception-type>java.lang.NullPointerException</exception-type>
        <location>/error/NullPointerException</location>
    </error-page>
--%>
<%--@ page errorPage="/error/viewError.jsp" --%>
<%@ page contentType="text/html; charset=UTF-8" %>
<!doctype html>
<html lang="ko">
<head>
<title>NullPointerException 상황 : 인자 출력</title>
</head>

```

```

<body>
<%
    // 안타깝게도 이럴 경우에는 NullPointerException 에러 감지가 안됩니다 !
    String name = request.getParameter("name")==null ?
        null : request.getParameter("name");
    out.println("name 의 자료형 : " +
        (name instanceof Object ? "문자열(String)" : "문자열(String)이 아님") + "<br>");

    // 아래의 경우를 번갈아 가면서 실행해 봅니다.
    // out.println("name : " + name + "<br>"); // NullPointerException 감지 안됨
    out.println("name : " + name.toString() + "<br>"); // NullPointerException 감지됨 -> 예외처리 페이지 이동
%>
    인자 값(name) : <%= name %>

</body>
</html>

<%--@ page errorPage="/error/viewError.jsp" --%>
<%@ page contentType="text/html; charset=UTF-8" %>
<!doctype html>
<html lang="ko">
<head>
<title>NullPointerException 상황 : 인자 출력</title>
</head>
<body>
<%
    // 아래와 같이 일일이 null을 점검하여 throw로 예외를 던지면 그때는 감지되어 예러 처리됩니다.
    if (request.getParameter("name") == null) {
        throw new NullPointerException();
    }

    String name = request.getParameter("name");
    out.println(name);
%>
    인자 값(name) : <%= name %>

</body>
</html>

```

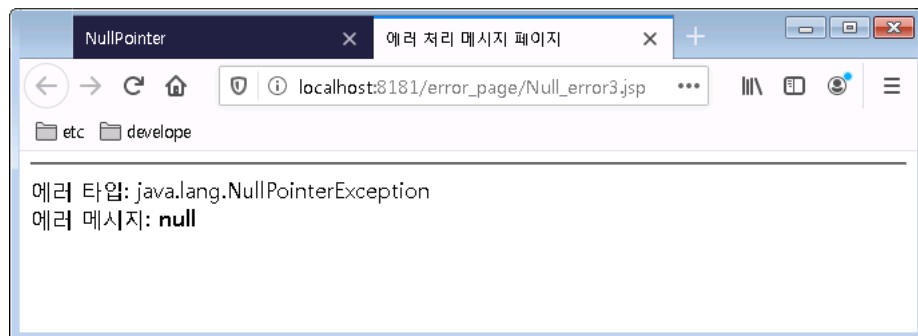
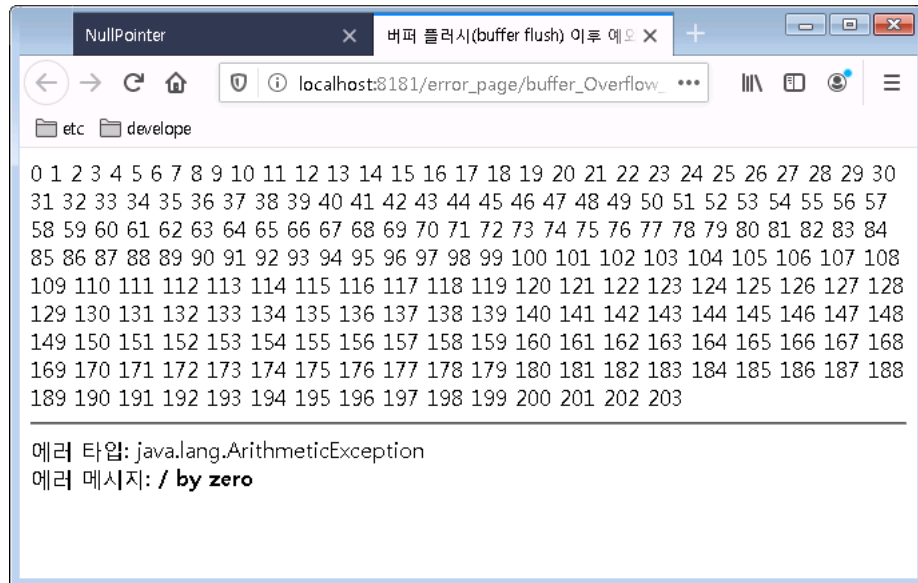
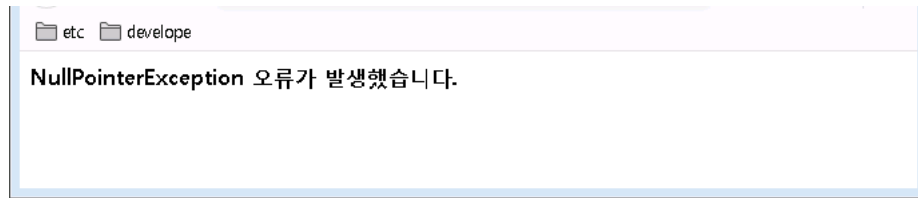
4. 실행 결과

1) 예외처리 없이 에러가 발생시



2) 예외 페이지 처리된 경우들



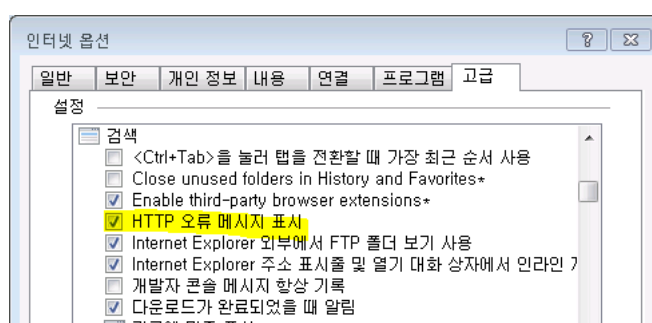


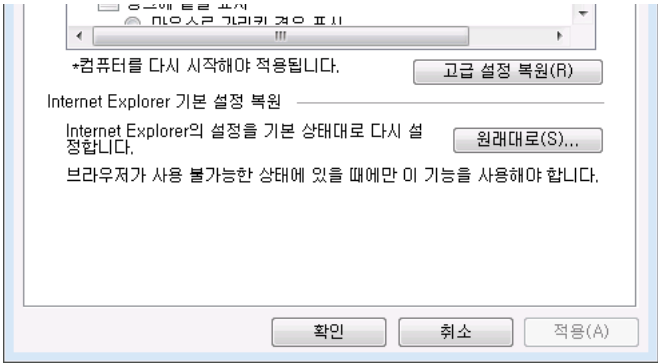
참고) IE에서의 문제점

: <https://javacan.tistory.com/entry/IE%EC%9D%98-HTTP-%EC%98%A4%EB%A5%98-%EB%A9%94%EC%8B%9C%E7%A7%80Friendly-Error-Message>

원문)

Several frequently-seen status codes have "friendly" error messages that Internet Explorer 5.x displays and that effectively mask the actual text message that the server sends. However, these "friendly" error messages are only displayed if the response that is sent to the client is less than or equal to a specified threshold. For example, to see the exact text of an **HTTP 500 response, the content length must be greater than 512 bytes.**





해결법) IE 설정 변경 및 강제로 513 byte 이상의 데이터를 첨가하는 방법이 있습니다.

주의사항) 에러 페이지는 반드시 버퍼(buffer)가 플러시(flush) 되기 전에 처리되어야 합니다.



error_page.war

댓글

댓글알림

김다솔

댓글을 남겨보세요

등록

더조은 강남 : 자바 프로그래밍(JSP) <https://cafe.naver.com/djjsp3>