

# 익셉션 처리



# 1. 에러 페이지 지정

- ❖ jsp페이지가 요청을 처리하는 도중에 예외가 발생하면 톰캣 서버가 에러화면을 출력합니다.

## HTTP Status 500 - java.lang.NullPointerException

**type** Exception report

**message** java.lang.NullPointerException

**description** The server encountered an internal error that prevented it from fulfilling this request.

**exception**

```
org.apache.jasper.JasperException: java.lang.NullPointerException
    org.apache.jasper.servlet.JspServletWrapper.handleJspException(JspServletWrapper.java:549)
    org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:470)
    org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:390)
    org.apache.jasper.servlet.JspServlet.service(JspServlet.java:334)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:728)
```

**root cause**

```
java.lang.NullPointerException
    org.apache.jsp.test_jsp._jspService(test_jsp.java:69)
    org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:728)
    org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:432)
    org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:390)
    org.apache.jasper.servlet.JspServlet.service(JspServlet.java:334)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:728)
```

Markers Properties Servers Data Source Explorer Snippets Console

Tomcat v7.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre7\bin\javaw.exe (2013. 1. 1. 오후 5:43:57)

```
at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter
at org.apache.coyote.http11.AbstractHttp11Processor.process(Abstract
at org.apache.coyote.AbstractProtocol$AbstractConnectionHandler.pro
at org.apache.tomcat.util.net.JIoEndpoint$SocketProcessor.run(JIoEn
at java.util.concurrent.ThreadPoolExecutor.runWorker(Unknown Source
at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Sourc
at java.lang.Thread.run(Unknown Source)
```

# 에러처리의 개요

- ❖ 웹 페이지에 에러가 발생시 에러발생페이지를 그대로 보여주면 사이트의 신뢰도가 떨어짐
  - 따라서, 에러메시지를 표시하지 말고 다른 페이지를 보여주는것=> 에러페이지
- ❖ page디렉티브의 `errorPage` 속성이 컨테이너의 버전에 따라 제대로 작동을 안할 수 있음.
- ❖ JSP규약에서는 에러코드별 처리를 권장
- ❖ HTTP에러코드에서 자주 발생하는 에러코드
  - 404는 주로 사용자가 잘못된 페이지를 요청할 때
  - 500은 프로그램 코딩 오류일 때 발생

# date.jsp

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
```

```
    pageEncoding="EUC-KR"%>
```

```
<%@ page import="java.util.*, java.text.*" %>
```

```
<%@ page errorPage="error.jsp"%>
```

```
<%
```

```
    Date date = new Date();
```

```
    SimpleDateFormat simpleDate = new SimpleDateFormat("yyyy-MM-dd");
```

```
    String strdate = simpleDate.format(date);
```

```
%>
```

보통의 JSP 페이지의 형태입니다.<br>

오늘 날짜는 <%= strdat%> 입니다. <!--고의로 에러를 발생시킨 라인  
으로 strdate변수명을 strdat로 틀리게 입력했다. -->

# AdderInput.html

```
<HTML>
  <HEAD>
    <META http-equiv="Content-Type"
content="text/html; charset=euc-kr">
    <TITLE>덧셈 프로그램 - 입력 화면</TITLE>
  </HEAD>
  <BODY>
    <FORM ACTION=/ch05_web/adder>
      첫 번째 수: <INPUT TYPE=TEXT NAME=NUM1><BR>
      두 번째 수: <INPUT TYPE=TEXT NAME=NUM2><BR>
      <INPUT TYPE=SUBMIT VALUE='더하기'>
    </FORM>
  </BODY>
</HTML>
```

# 웹 컴포넌트에서 발생하는 익셉션 처리

## ❖ 스탠드얼론 프로그램과 웹 컴포넌트에서의 익셉션 처리 방법

- 웹 서버와 무관하게 독립적으로 작동하는 스탠드얼론 프로그램에서는 try문을 이용해서 익셉션을 처리한다.

### Adder.java

```
public class Adder {  
    public static void main(String args[]) {  
        try {  
            int num1 = Integer.parseInt(args[0]);  
            int num2 = Integer.parseInt(args[1]);  
            int result = num1 + num2;  
            System.out.printf( "%d + %d = %d ", num1, num2, result);  
        }  
        catch (NumberFormatException e) {  
            System.out.println( "잘못된 데이터가 입력되었습니다." );  
        }  
    }  
}
```

## ❖ 스탠드얼론 프로그램과 웹 컴포넌트에서의 익셉션 처리 방법

### AdderServlet.java

```
import javax.servlet.http.*; import javax.servlet.*; import java.io.*;
public class AdderServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        response.setContentType("text/html;charset=utf-8");
        PrintWriter out = response.getWriter();
        out.println("<html> <body>");
        try {
            int num1 = Integer.parseInt(request.getParameter("num1"));
            int num2 = Integer.parseInt(request.getParameter("num2"));
            int sum = num1 + num2;
            out.printf("%d + %d = %d", num1, num2, sum);
        } catch (NumberFormatException e) {
            out.println("그게 숫자냐");
        }
        out.println("</body> </html>");
        out.close();
    }
}
```

## ❖ 스탠드얼론 프로그램과 웹 컴포넌트에서의 익셉션 처리 방법

### Adder\_old.jsp

```
<html> <head> <meta http-equiv= "Content-Type" content="text/html;
charset=UTF-8">
<title>Insert title here</title> </head>
<body>
<%
    try{
        int num1 = Integer.parseInt(request.getParameter("num1"));
        int num2 = Integer.parseInt(request.getParameter("num2"));
        out.println(num1 + " + " + num2 + " = " +(num1+num2));
    }catch(NumberFormatException e ) {
%>
<script type= "text/javascript">
    alert("그게 숫자냐");
    history.go(-1);
</script>
<% } %>
</body>
</html>
```



# 에러 페이지 만들어서 호출하기

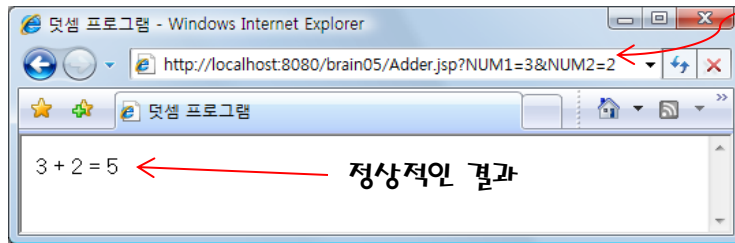
## Adder.jsp

```
<html> <head> <meta http-equiv= "Content-Type" content="text/html;
charset=UTF-8">
<title>Insert title here</title> </head>
<body>
<%
    try{
        int num1 = Integer.parseInt(request.getParameter("num1"));
        int num2 = Integer.parseInt(request.getParameter("num2"));
        out.println(num1 + " + " + num2 + " = " +(num1+num2));
    }catch(NumberFormatException e ) {
        request.setAttribute("err", e.getMessage());
        RequestDispatcher rd =
            request.getRequestDispatcher("dataError.jsp");
        rd.forward(request, response);
    }
%>
</body>
</html>
```

## ❖ JSP 페이지에서 에러 페이지 호출하기

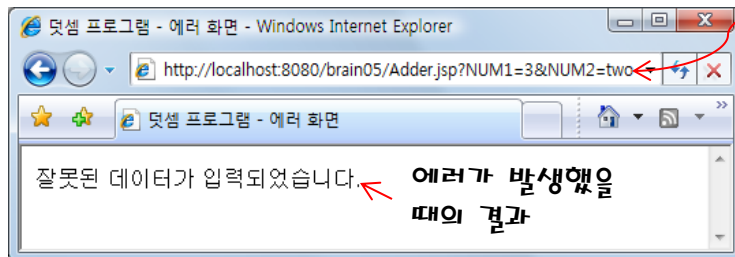
### DataError\_old.jsp

```
<% @page contentType= "text/html; charset=euc-kr" %>
<HTML>
  <HEAD><TITLE>덧셈 프로그램 - 에러 화면</TITLE></HEAD>
  <BODY>
    잘못된 데이터가 입력되었습니다.
  </BODY>
</HTML>
```



(그림 5-4)를 호출하면서 URL 뒤에 데이터를 직접 쓰세요

er.jsp?NUM1=3&NUM2=2



이번에는 입력 데이터 중 하나를 숫자가 아닌 값으로 쓰세요

r.jsp?NUM1=3&NUM2=two

[그림 5-1] 예제 5-4, 5-5의 실행 결과

# 에러 페이지 지정

- ❖ page 디렉티브를 이용해서 에러 페이지를 지정할 수 있습니다.
- ❖ 에러 페이지 지정 형식: <%@ page errorPage = 페이지 경로 %>
- ❖ 에러 페이지 작성 요령
  - ❖ page 디렉티브에서 isErrorPage 속성을 true로 설정
  - ❖ exception 객체 사용 가능
  - ❖ explorer에서는 전체의 응답 상태 코드가 404나 500 같은 에러코드이고 전체 응답 결과 데이터의 길이가 513바이트 보다 작으면 자체적으로 제공하는 오류 메시지 화면을 출력하므로 유의

# test.jsp

```
<%@ page contentType = "text/html; charset=UTF-8" %>
<%@ page errorPage = "/viewErrorMessage.jsp" %>
<html>
<head> <title>파라미터 출력</title> </head>
<body>

name 파라미터 값: <%=
request.getParameter("name").toUpperCase() %>

</body>
</html>
```

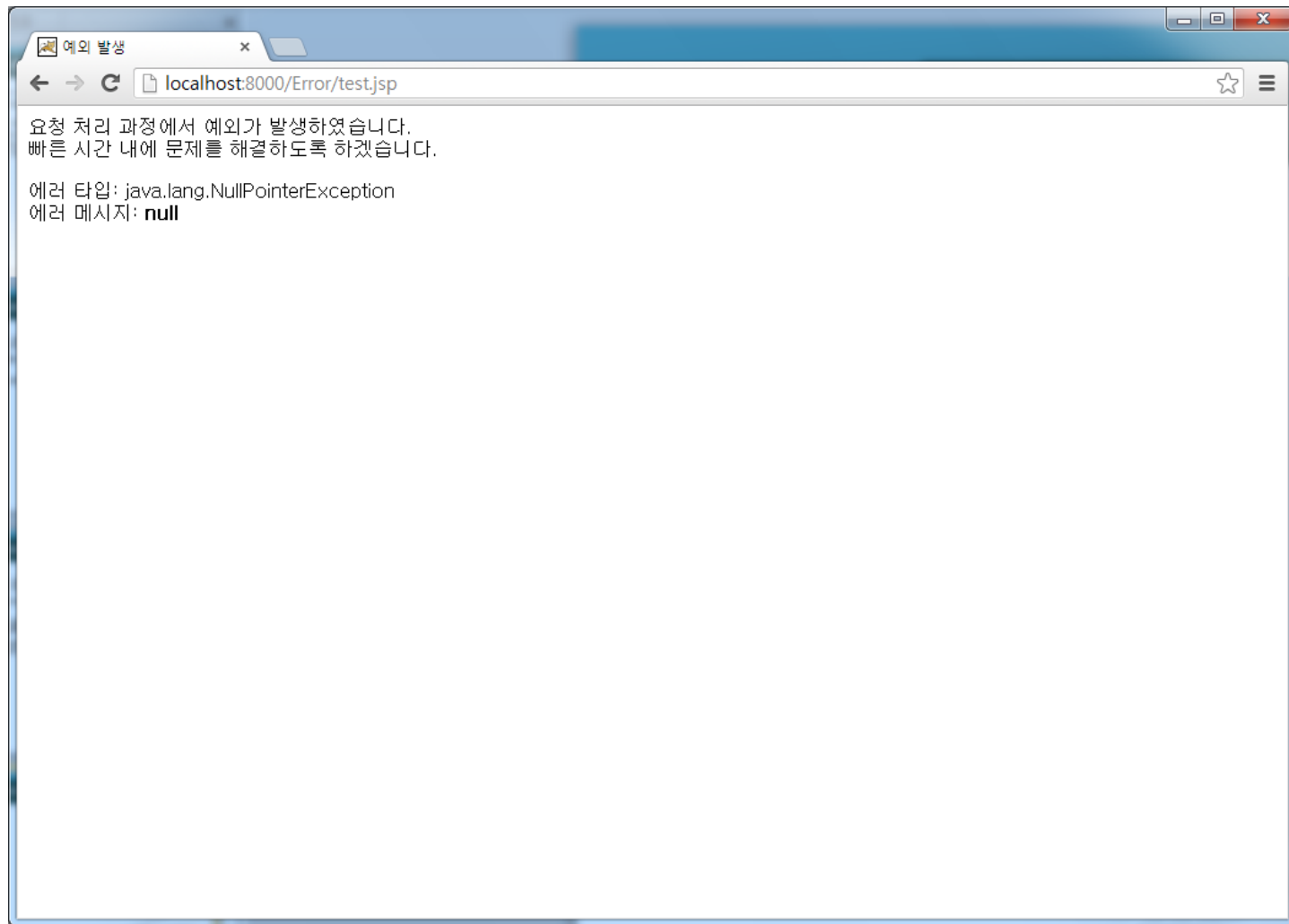
# viewErrorMessage.jsp

```
<%@ page contentType = "text/html; charset=UTF-8" %>
<%@ page isErrorPage = "true" %>
<html>
<head> <title> 예외 발생 </title> </head>
<body>
```

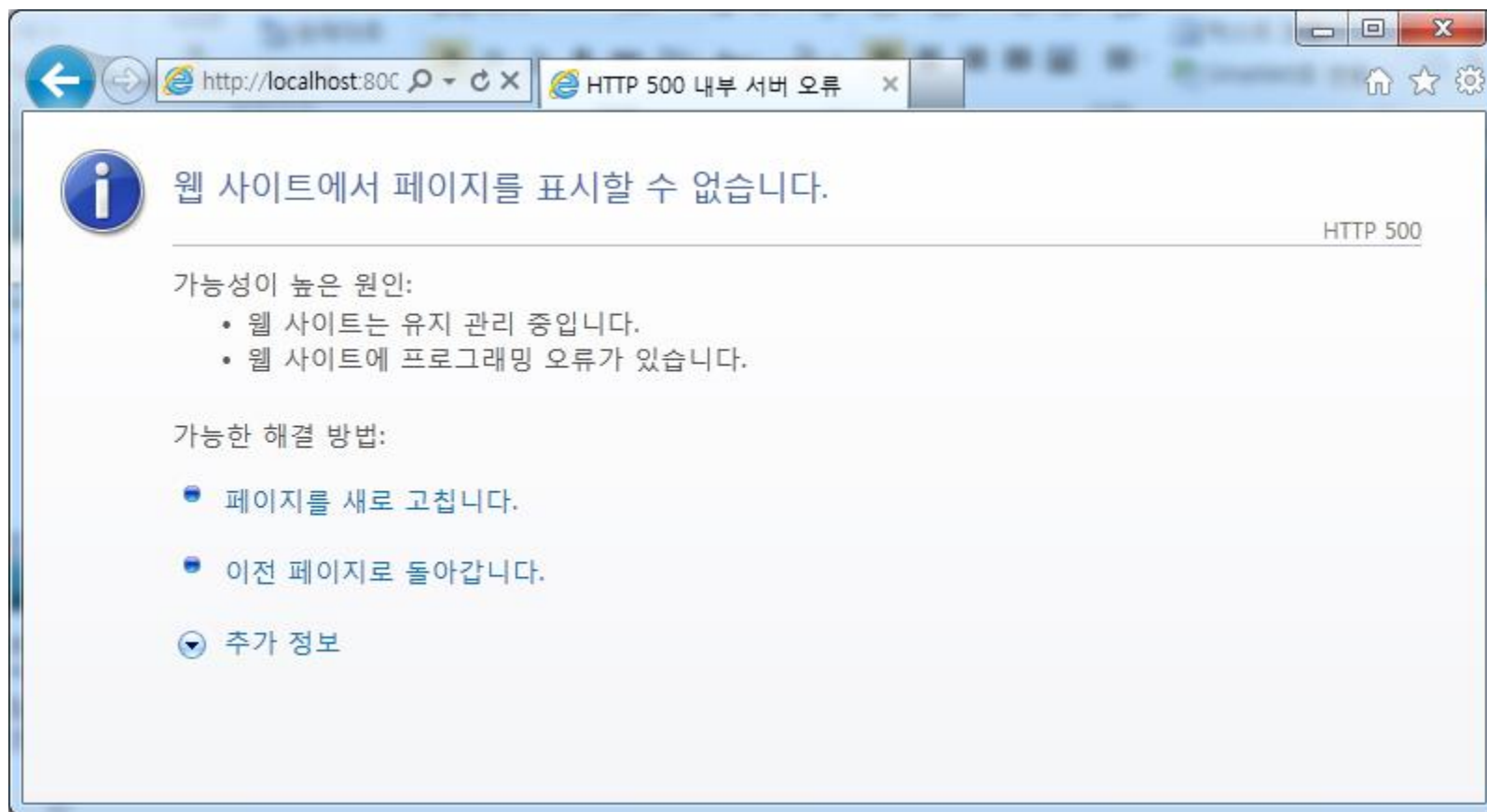
요청 처리 과정에서 예외가 발생하였습니다.<br>빠른 시간 내에 문제를 해결하도록 하겠습니다.

```
<p>
에러 타입: <%= exception.getClass().getName() %> <br>
에러 메시지: <b> <%= exception.getMessage() %> </b>
</body>
</html>
```

# 결과(크롬)



# 결과(explorer)



# error/viewErrorMessage.jsp 수정

```
<%@ page contentType = "text/html; charset=UTF-8" %>
<%@ page isErrorPage = "true" %>
<html>
<head> <title>예외 발생</title> </head>
<body>
요청 처리 과정에서 예외가 발생하였습니다.<br>
빠른 시간 내에 문제를 해결하도록 하겠습니다.
<p>
에러 타입: <%= exception.getClass().getName() %> <br>
에러 메시지: <b><%= exception.getMessage() %></b>
</body>
</html>
<%--
<!--
만약 에러 페이지의 길이가 513 바이트보다 작다면
인터넷 익스플로러는 이 페이지가 출력하는 에러 페이지를 출력하지 않고
자체적으로 제공하는 'HTTP 오류 메시지' 화면을 출력할 것입니다.
만약 에러 페이지의 길이가 513 바이트보다 작는데
에러 페이지의 내용이 인터넷 익스플로러에서도 올바르게 출력되길 원한다면
응답 결과에 이 주석과 같은 내용을 포함시켜서
에러 페이지의 길이가 513 바이트 이상이 되도록 해 주어야 합니다.
-->
--%>
```



## ❖ JSP 페이지에서 에러 페이지 호출하기

### 두 수를 더하는 JSP 페이지 – page 지시자의 errorPage 애트리뷰트 사용

```
<% @page contentType= "text/html; charset=euc-kr " errorPage=
"DataError.jsp " %>
```

```
<%
```

```
String str1 = request.getParameter( "NUM1 ");
```

```
String str2 = request.getParameter( "NUM2 ");
```

```
int num1 = Integer.parseInt(str1);
```

```
int num2 = Integer.parseInt(str2);
```

```
int result = num1 + num2;
```

```
%>
```

```
<HTML>
```

```
<HEAD><TITLE>덧셈 프로그램</TITLE></HEAD>
```

```
<BODY>
```

```
<%= num1 %> + <%= num2 %> = <%= result %>
```

```
</BODY>
```

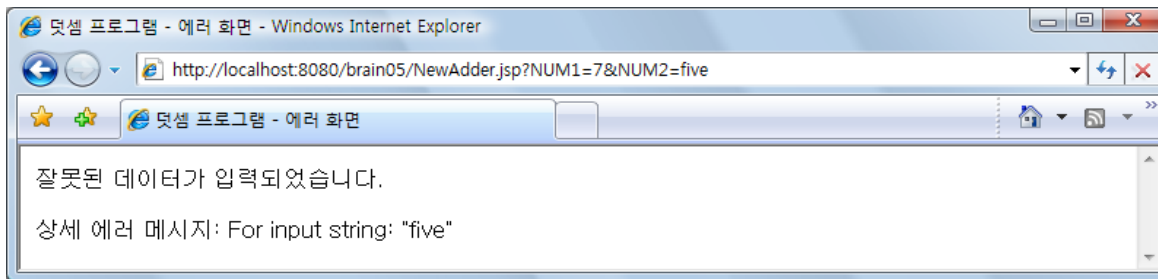
```
</HTML>
```

## ❖ JSP 페이지에서 에러 페이지 호출하기

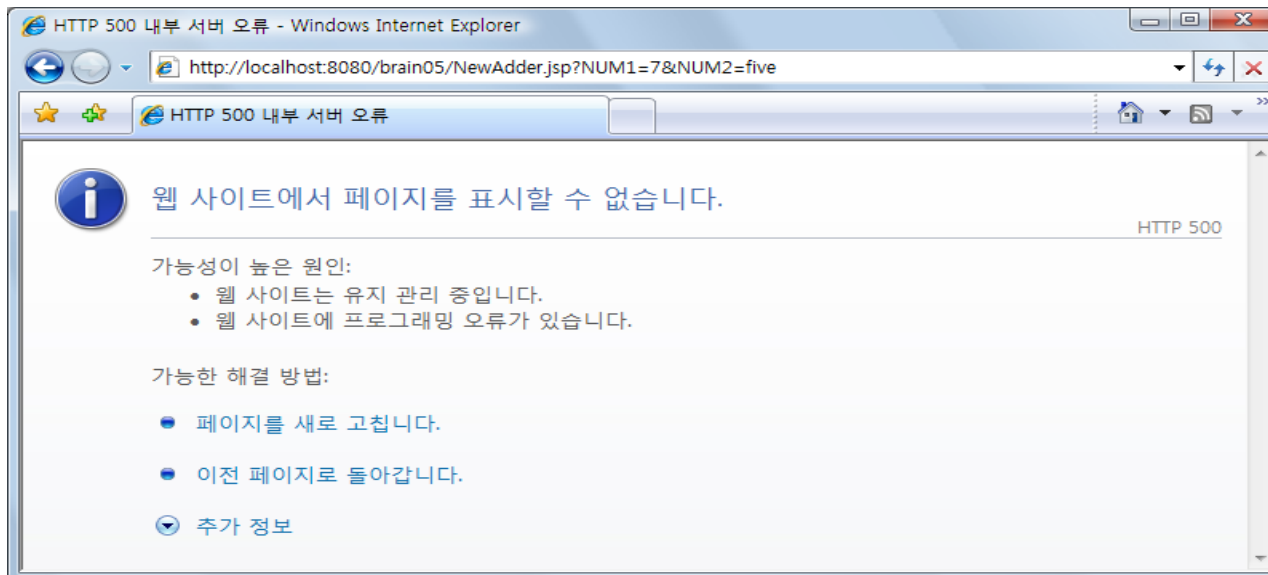
### DataError.jsp

```
<% @page contentType= "text/html; charset=euc-kr" isErrorPage= "true" %>
<HTML>
    <HEAD><TITLE>덧셈 프로그램 - 에러 화면</TITLE></HEAD>
    <BODY>
        잘못된 데이터가 입력되었습니다. <BR><BR>
        상세 에러 메시지: <%= exception.getMessage() %>
    </BODY>
</HTML>
```

## ❖ JSP 페이지에서 에러 페이지 호출하기



[그림 5-3] 예제 5-6, 5-7의 실행 결과(1)



[그림 5-3] 예제 5-6, 5-7의 실행 결과(2)

## ❖ JSP 페이지에서 에러 페이지 호출하기

- [그림 5-4]는 웹 브라우저에 내장되어 있는 에러 표시용 웹 페이지이며, 이런 결과가 나오는 이유는 [예제 5-7]이 생성한 HTML 문서와 함께 웹 브라우저로 전달된 HTTP 상태 코드 때문이다.
- HTTP 상태 코드란 HTTP 응답 메시지의 시작 행에 표시되는 3자리의 숫자인데, 메시지에 포함된 HTML 문서가 정상적인 실행의 결과인지 에러 발생의 결과인지 구분하는 역할을 한다.

## ❖ JSP 페이지에서 에러 페이지 호출하기

### HTTP 응답 메시지

정상적인 실행 결과일 때

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/html;charset=euc-kr
Content-Length: 110
Date: Thu, 18 Sep 2008 19:52:11 GMT

<HTML>
  <HEAD><TITLE>덧셈 프로그램 - 결과 화면</TITLE></HEAD>
  <BODY>
    5 + 3 = 8
  </BODY>
</HTML>
```

### HTTP 응답 메시지

실행 도중 에러가 발생했을 때

```
HTTP/1.1 500 Internal Server Error
Server: Apache-Coyote/1.1
Content-Type: text/html;charset=euc-kr
Content-Length: 133
Date: Thu, 18 Sep 2008 19:58:53 GMT
Connection: close

<HTML>
  <HEAD><TITLE>덧셈 프로그램 - 에러 화면</TITLE></HEAD>
  <BODY>
    잘못된 데이터가 입력되었습니다.
  </BODY>
</HTML>
```

[그림 5-5] HTTP 응답 메시지의 상태 코드

# 에러 페이지 만들어서 호출하기

## ❖ JSP 페이지에서 에러 페이지 호출하기

- 웹 브라우저는 상태 코드 값이 500이면 HTTP 응답 메시지에 포함된 HTML 문서의 내용을 무시하고 웹 브라우저 자체에 내장된 에러 메시지를 출력한다.
- 인위적으로 HTTP 상태 코드의 값을 200으로 바꿔주기 위해서는 response 내장 변수에 대해 setStatus라는 메서드를 호출하면서 200이라는 파라미터값을 넘겨주면 된다.

```
response.setStatus(200);
```

HTTP 상태 코드

### DataError.jsp

```
<% @page contentType= "text/html; charset=euc-kr" isErrorPage= "true " %>
<% response.setStatus(200); %>
<HTML>
  <HEAD><TITLE>뎃셈 프로그램 - 에러 발생</TITLE></HEAD>
  <BODY>
    잘못된 데이터가 입력되었습니다. <BR><BR>
    상세 에러 메시지: <%= exception.getMessage() %>
  </BODY>
</HTML>
```

## ❖ 서블릿 클래스에서 에러 페이지 호출하기

### NewAdderServlet.java

```
import javax.servlet.http.*; import javax.servlet.*; import java.io.*;
public class NewAdderServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        try {
            String str1 = request.getParameter( "NUM1 ");
            String str2 = request.getParameter( "NUM2 ");
            int num1 = Integer.parseInt(str1);          int num2 = Integer.parseInt(str2);
            int result = num1 + num2;
            response.setContentType( "text/html;charset=euc-kr ");
            PrintWriter out = response.getWriter();
            out.println( "<HTML> ");
            out.println( "<HEAD><TITLE>덧셈 프로그램</TITLE></HEAD> ");
            out.println( "<BODY> ");
            out.printf( "%d + %d = %d ", num1, num2, result);
            out.println( "</BODY> ");
            out.println( "</HTML> ");
        }
        catch (NumberFormatException e) {
            RequestDispatcher dispatcher = request.getRequestDispatcher( "data-error ");
            dispatcher.forward(request, response);
        }
    }
}
```

## ❖ 서블릿 클래스에서 에러 페이지 호출하기

### DataErrorServlet.java

```
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;
public class DataErrorServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        response.setContentType( "text/html;charset=euc-kr ");
        PrintWriter out = response.getWriter();
        out.println( "<HTML> ");
        out.println( "<HEAD><TITLE>뎃셈 프로그램 - 에러
화면</TITLE></HEAD> ");
        out.println( "<BODY> ");
        out.println( "잘못된 데이터가 입력되었습니다. " );
        out.println( "</BODY> ");
        out.println( "</HTML> ");
        return;
    }
}
```



# 에러코드별 처리

## ❖ HTTP에러코드에서 자주 발생하는 에러코드

- 200 : request 가 성공적으로 완료되었음
- 204: request가 실행되었으나, 클라이언트에게 보낼 데이터가 없음
- 200: 요구된 request가 여러 위치에 존재하는 자원을 필요하므로, response는 위에 대한 정보를 보낸다. 클라이언트는 가장 적당한 위치를 선택하여야 한다.
- 302: request가 요구한 데이터를 발견 하였으나, 실제로 다른 URL에 존재함
- 307 : 임시로 페이지가 리다이렉트
- 400: request의 문법이 잘못되었음
- 401 : 접근이 허용되지 않음
- 403: request는 금지된 자원을 요청하였음
- 404: 요청한 URL을 찾을 수 없음
- 405 : 요청된 메서드는 허용되지 않음
- 500: 서버에 내부적인 오류가 있어 더 이상 진행 할 수 없음
- 501: 요청된 request는 합법적이거나 서버는 요구된 method를 실행할 수 없음
- 503: 서버가 busy한 상태라 서비스를 할 수 없음

# 응답 상태 코드 별 에러 페이지

## ❖ web.xml 파일에 작성

<error-page>

<error-code>에러코드</error-code>

<location>에러페이지의 경로</location>

</error-page>

## ❖ 주요 응답 상태 코드

❖ 200 : 요청이 정상적으로 처리

❖ 307 : 임시로 페이지가 리다이렉트

❖ 400 : 클라이언트의 요청이 잘못된 구문

❖ 401 : 접근이 허용되지 않음

❖ 404 : 자원이 존재하지 않음

❖ 405 : 요청된 메서드는 허용되지 않음

❖ 500 : 서버 내부 에러(JSP에서 예외 발생)

❖ 503 : 서버가 일시적으로 서비스 할 수 없음(과부하나 임시 보수 중)

# 실습

⏪ ⏩ 🛑 💰 http://localhost:8000/Error/test1.jsp

**요청한 페이지는 존재하지 않습니다:**

주소를 올바르게 입력했는 지 확인해보시기 바랍니다.

# web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  version="3.0">
  <error-page>
    <error-code>404</error-code>
    <location>/error/error404.jsp</location>
  </error-page>

</web-app>
```

# error/error404.jsp

```
<%@ page contentType = "text/html; charset=UTF-8" %>
<html>
<head><title>404 에러 발생</title></head>
<body>
<strong>요청한 페이지는 존재하지 않습니다:</strong>
<br><br>
주소를 올바르게 입력했는 지 확인해보시기 바랍니다.
</body>
</html>
<!--
```

만약 에러 페이지의 길이가 513 바이트보다 작다면,  
인터넷 익스플로러는 이 페이지가 출력하는 에러 페이지를 출력하지 않고  
자체적으로 제공하는 'HTTP 오류 메시지' 화면을 출력할 것이다.

만약 에러 페이지의 길이가 513 바이트보다 작으면  
에러 페이지의 내용이 인터넷 익스플로러에서도 올바르게 출력되길 원한다면,  
응답 결과에 이 주석과 같은 내용을 포함시켜서  
에러 페이지의 길이가 513 바이트 이상이 되도록 해 주어야 한다.  
참고로 이 주석은 456바이트이다.  
-->

# 예외 타입 별 에러 페이지

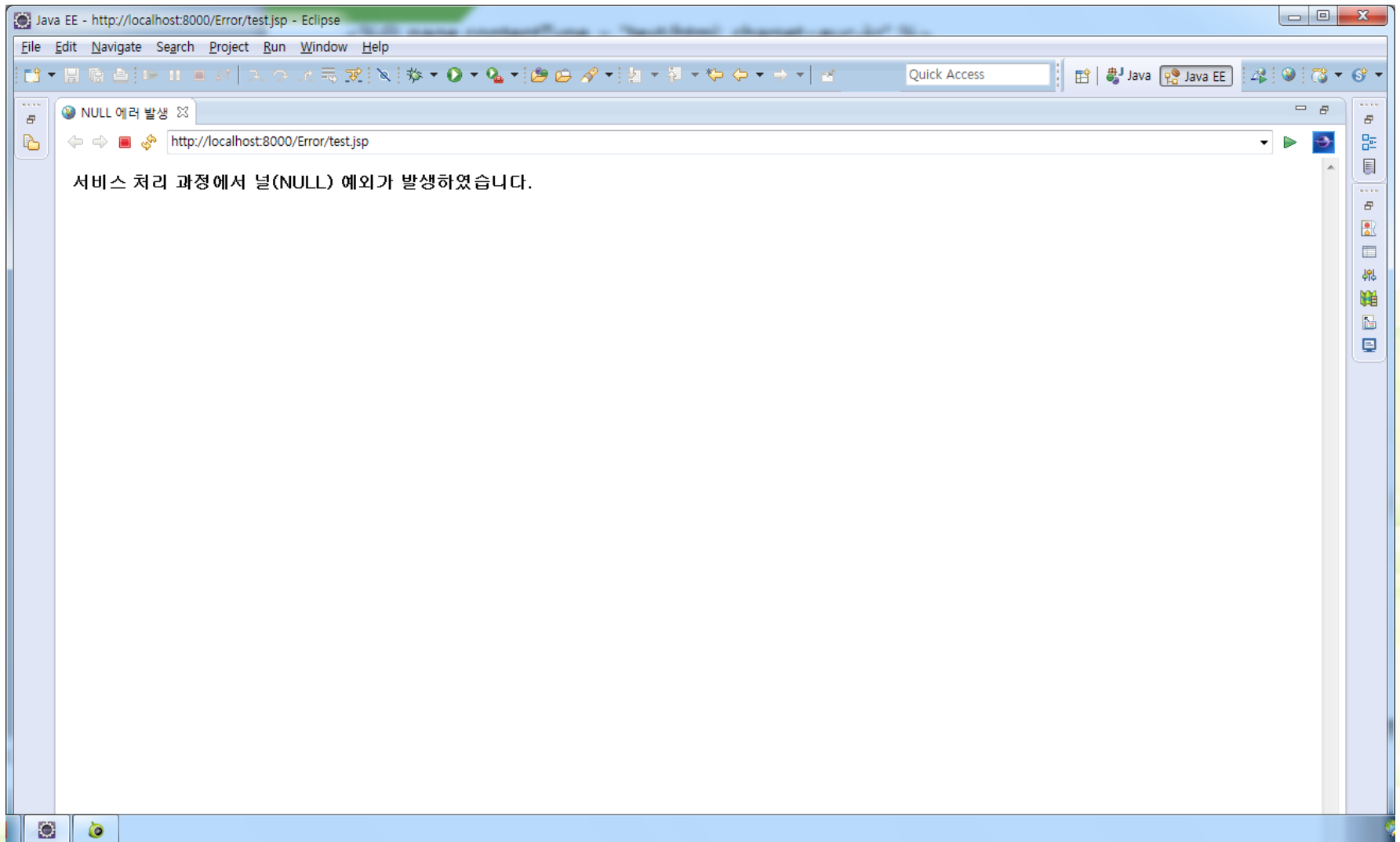
❖ web.xml 파일에 작성

<error-page>

<exception-type>예외 이름</exception-type>

<location>에러 페이지 경로</location>

</error-page>



# web.xml 수정

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"  
  version="3.0">
```

```
  <error-page>
```

```
    <error-code>404</error-code>
```

```
    <location>/error/error404.jsp</location>
```

```
  </error-page>
```

```
  <error-page>
```

```
    <exception-type>java.lang.NullPointerException</exception-type>
```

```
    <location>/error/errorNullPointerException.jsp</location>
```

```
  </error-page>
```

```
</web-app>
```



# error/errorNullPointerException.jsp

```
<%@ page contentType = "text/html; charset=UTF-8" %>
```

```
<%
```

```
response.setStatus(HttpServletResponse.SC_OK);
```

```
%>
```

```
<html>
```

```
<head> <title>NULL 에러 발생</title> </head>
```

```
<body>
```

```
<strong>서비스 처리 과정에서 널(NULL) 예외가 발생하였습니다.</strong>
```

```
<%--
```

```
<!--
```

만약 에러 페이지의 길이가 513 바이트보다 작다면

인터넷 익스플로러는 이 페이지가 출력하는 에러 페이지를 출력하지 않고  
자체적으로 제공하는 'HTTP 오류 메시지' 화면을 출력할 것입니다.

만약 에러 페이지의 길이가 513 바이트보다 작으면

에러 페이지의 내용이 인터넷 익스플로러에서도 올바르게 출력되길 원한다면

응답 결과에 이 주석과 같은 내용을 포함시켜서

에러 페이지의 길이가 513 바이트 이상이 되도록 해 주어야 합니다.

```
-->
```

```
--%>
```

```
</body>
```

```
</html>
```

# test.jsp 파일 수정

```
<%@ page contentType = "text/html; charset=UTF-8" %>
```

```
<html>
```

```
<head> <title>파라미터 출력</title> </head>
```

```
<body>
```

```
name 파라미터 값: <%= request.getParameter("name").toUpperCase() %>
```

```
</body>
```

```
</html>
```

# 에러 페이지의 우선 순위와 지정

## ❖ 우선 순위

- ❖ Page 디렉티브의 `errorPage` 속성에서 지정한 페이지
- ❖ JSP 페이지에서 발생한 예외 타입이 `web.xml`에서 지정된 경우
- ❖ JSP 페이지에서 발생한 에러 코드가 `web.xml`에서 지정된 경우
- ❖ 웹 컨테이너의 에러 페이지

## ❖ 지정 방법

- ❖ 별도의 에러 페이지가 있어야 하는 경우 `page` 디렉티브 속성을 이용
- ❖ 범용적인 에러코드(404, 500)는 `web.xml`에서 처리
- ❖ 별도로 처리해야 하는 심각한 예외 타입에 대해서는 `web.xml` 파일을 이용해서 별도의 에러 페이지를 지정