SERVLETS - EXCEPTION HANDLING

When a servlet throws an exception, the web container searches the configurations in **web.xml** that use the exception-type element for a match with the thrown exception type.

You would have to use the **error-page** element in web.xml to specify the invocation of servlets in response to certain **exceptions** or HTTP **status codes**.

web.xml Configuration:

Consider, you have an *ErrorHandler* servlet which would be called whenever there is any defined exception or error. Following would be the entry created in web.xml.

```
<!-- servlet definition -->
<servlet>
        <servlet-name>ErrorHandler</servlet-name>
        <servlet-class>ErrorHandler</servlet-class>
</servlet>
<!-- servlet mappings -->
<servlet-mapping>
        <servlet-name>ErrorHandler</servlet-name>
        <url-pattern>/ErrorHandler</url-pattern>
</servlet-mapping>

<!-- error-code related error pages -->
<error-page>
    <error-code>404</error-code>
    <location>/ErrorHandler</location>
</error-page>
<error-page>
    <error-code>403</error-code>
    <location>/ErrorHandler</location>
</error-page>

<!-- exception-type related error pages -->
<error-page>
    <exception-type>
          javax.servlet.ServletException
    </exception-type >
    <location>/ErrorHandler</location>
</error-page>

<error-page>
    <exception-type>java.io.IOException</exception-type >
    <location>/ErrorHandler</location>
</error-page>
```

If you want to have a generic Error Handler for all the exceptions then you should define following error-page instead of defining separate error-page elements for every exception:

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

Following are the points to be noted about above web.xml for Exception Handling:

- The servlet ErrorHandler is defined in usual way as any other servlet and configured in web.xml.
- If there is any error with status code either 404 *NotFound* or 403 *Forbidden*, then ErrorHandler servlet would be called.

- If the web application throws either *ServletException* or *IOException*, then the web container invokes the /ErrorHandler servlet.
- You can define different Error Handlers to handle different type of errors or exceptions. Above example is very much generic and hope it serve the purpose to explain you the basic concept.

Request Attributes - Errors/Exceptions:

Following is the list of request attributes that an error-handling servlet can access to analyse the nature of error/exception.

S.N.	Attribute & Description
1	**javax.servlet.error.status_code** This attribute give status code which can be stored and analysed after storing in a java.lang.Integer data type.
2	**javax.servlet.error.exception_type** This attribute gives information about exception type which can be stored and analysed after storing in a java.lang.Class data type.
3	**javax.servlet.error.message** This attribute gives information exact error message which can be stored and analysed after storing in a java.lang.String data type.
4	**javax.servlet.error.request_uri** This attribute gives information about URL calling the servlet and it can be stored and analysed after storing in a java.lang.String data type.
5	**javax.servlet.error.exception** This attribute gives information the exception raised which can be stored and analysed after storing in a java.lang.Throwable data type.
6	**javax.servlet.error.servlet_name** This attribute gives servlet name which can be stored and analysed after storing in a java.lang.String data type.

Error Handler Servlet Example:

Following is the Servlet Example that would be used as Error Handler in case of any error or exception occurs with your any of the servlet defined.

This example would give you basic understanding of Exception Handling in Servlet, but you can write more sophisticated filter applications using the same concept:

```
// Import required java libraries
import java.io.*;
```

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;

// Extend HttpServlet class
public class ErrorHandler extends HttpServlet {

  // Method to handle GET method request.
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
            throws ServletException, IOException
  {
      // Analyze the servlet exception
      Throwable throwable = (Throwable)
      request.getAttribute("javax.servlet.error.exception");
      Integer statusCode = (Integer)
      request.getAttribute("javax.servlet.error.status_code");
      String servletName = (String)
      request.getAttribute("javax.servlet.error.servlet_name");
      if (servletName == null){
         servletName = "Unknown";
      }
      String requestUri = (String)
      request.getAttribute("javax.servlet.error.request_uri");
      if (requestUri == null){
         requestUri = "Unknown";
      }

      // Set response content type
      response.setContentType("text/html");

      PrintWriter out = response.getWriter();
   String title = "Error/Exception Information";
      String docType =
      "<!doctype html public \"-//w3c//dtd html 4.0 " +
      "transitional//en\">\n";
      out.println(docType +
        "<html>\n" +
        "<head><title>" + title + "</title></head>\n" +
        "<body bgcolor=\"#f0f0f0\">\n");

      if (throwable == null && statusCode == null){
         out.println("<h2>Error information is missing</h2>");
         out.println("Please return to the <a href=\"" +
           response.encodeURL("http://localhost:8080/") +
           "\">Home Page</a>.");
      }else if (statusCode != null){
         out.println("The status code : " + statusCode);
      }else{
         out.println("<h2>Error information</h2>");
         out.println("Servlet Name : " + servletName +
                             "</br></br>");
         out.println("Exception Type : " +
                             throwable.getClass( ).getName( ) +
                             "</br></br>");
         out.println("The request URI: " + requestUri +
                             "<br><br>");
         out.println("The exception message: " +
                                  throwable.getMessage( ));
      }
      out.println("</body>");
      out.println("</html>");
  }
  // Method to handle POST method request.
  public void doPost(HttpServletRequest request,
                     HttpServletResponse response)
      throws ServletException, IOException {
     doGet(request, response);
  }
```

```
}
```

Compile **ErrorHandler.java** in usual way and put your class file in <Tomcat-installation-directory>/webapps/ROOT/WEB-INF/classes.

Let us add the following configuration in web.xml to handle exceptions:

```
<servlet>
        <servlet-name>ErrorHandler</servlet-name>
        <servlet-class>ErrorHandler</servlet-class>
</servlet>
<!-- servlet mappings -->
<servlet-mapping>
        <servlet-name>ErrorHandler</servlet-name>
        <url-pattern>/ErrorHandler</url-pattern>
</servlet-mapping>
<error-page>
    <error-code>404</error-code>
    <location>/ErrorHandler</location>
</error-page>
<error-page>
    <exception-type>java.lang.Throwable</exception-type >
    <location>/ErrorHandler</location>
</error-page>
```

Now try to use a servlet which raise any exception or type a wrong URL, this would trigger Web Container to call **ErrorHandler** servlet and display an appropriate message as programmed. For example, if you type a wrong URL then it would display the following result:

```
The status code : 404
```

Above code may not work with some web browsers. So try with Mozilla and Safari and it should work