Com gestionar les excepcions en la fase de renderitzat

A qui va dirigit

Aquest how-to va dirigit a tots els desenvolupadors que treballin amb qualsevol de les versions de Canigó.

Versió de Canigó

Els passos descrits en aquest document són d'aplicació a les versions 1..x i 2.x de Canigó.

Context

En el tractament d'una petició (*request*), en la majoria dels casos, ens trobem dues fases principals: l'execució de l'acció (*action*) i el renderitzat de la vista.

En la primera de les dues fases, Canigó realitza el tractament i gestió de les excepcions correctament, redirigint a una pàgina d'error -definida per l'usuari- en el cas que es produeixi una excepció.

D'altra banda, trobem la fase de renderitzat de la vista. Quan s'arriba a aquesta fase significa que s'ha executat l'acció sense problemes i ja s'ha començat a generar una resposta (*response*) amb la informació associada. Si es produeix una excepció en aquest punt, amb la configuració actual, ja 'no estem a temps' de redireccionar-nos a una vista, perquè l'*action* s'ha executat sense problemes i el *response* conté un resultat positiu.

Per tant, presentem un mètode per a què es puguin redirigir aquestes excepcions de segona fase a una vista mitjançant Spring. Estem parlant d'excepcions produïdes per un error de sintaxi en un tag, de propietats incorrectes, etc. que són errors que es produeixen durant el desenvolupament d'una aplicació.

Cal destacar que, bàsicament, oferim una tècnica per a millorar el desenvolupament i una ràpida localització dels errors que es donen en aquesta etapa. És molt difícil que aquest problema que plantegem sorgeixi a l'entorn de producció.

La solució proposada és **provisional** i es pot implementar sense haver d'actualitzar el servei web de Canigó. En properes versions ja s'inclourà dins aquest servei la solució definitiva, en el que s'inclourà un *handler* que gestionarà aquest tipus d'excepcions.

Objectiu

Aconseguir capturar les excepcions produïdes en la fase de renderitzat de la pàgina es mostrin en una vista concreta definida per l'usuari.

Com gestionar les excepcions en la fase de renderitzat

Passos a seguir

1. **Definir els beans de resolució d'excepcions i la vista on es mostraran les excepcions:**

Fitxer de configuració	**prototip-servlet.xml**
Ubicació proposada	<PROJECT_ROOT>/src/main/resources/spring

```
<bean id="viewResolver"
      class="org.springframework.web.servlet.view.UrlBasedViewResolver">
     <property name="prefix" value="/WEB-INF/jsp/"/>
     <property name="suffix" value=".jsp"/>
     <property name="viewClass"
               value="org.springframework.web.servlet.view.JstlView"/>
</bean>

<bean id="exceptionResolver"
      class="org.springframework.web.servlet.handler.SimpleMappingExceptionResolver">
     <property name="exceptionMappings">
        <props>
          <prop key="net.gencat.ctti.canigo.services.exceptions.SystemException">
            error
          </prop>
        </props>
     </property>
</bean>
```

En el bean "viewResolver" es defineixen les propietats de les vistes que es mostraran quan es produeixi una excepció. En l'exemple s'està definint que la vista es troba ubicada al directori "/WEB-INF/jsp" i aquesta tindrà l'extensió ".jsp".

Aquesta configuració permet definir una vista per a cada tipus d'excepció, en la que es mostraran els detalls de la mateixa. El bean "exceptionResolver" conté la propietat "exceptionMappings" on es definiran els mapejos a les vistes en funció del tipus d'excepció.

Per exemple, si es volen tractar de manera particular les excepcions del tipus *TagsServiceException* (pròpia de Canigó) i les excepcions *IOException* (excepció genèrica), s'hauria de definir de la següent manera:

```
<prop key="net.gencat.ctti.canigo.services.exceptions.TagsServiceException">
   errorTag
</prop>
<prop key="java.io.IOException">
   errorIO
</prop>
```

On "errorTag" i "errorIO" seran vistes que hauran d'estar ubicades en el directori que s'hagi definit en el bean anterior, en aquest cas, "/WEB-INF/jsp" amb extensió ".jsp".

Com gestionar les excepcions en la fase de renderitzat

2. Crear la vista

Extensió fitxer	*.jsp
Ubicació proposada	<PROJECT_ROOT>/src/webapp/WEB-INF/jsp

Si es produeix una excepció de renderitzat, a la *request* tindrem un atribut anomenat "exception", el qual conté la tota la informació de l'excepció. La manera de recuperar aquesta informació a la vista és la següent:

```
<%
request.setAttribute(Globals.ERROR_KEY,ActionMessagesHelper.
     buildSystemExceptionMessages((SystemException)request.
        getAttribute("exception")));
%>
```

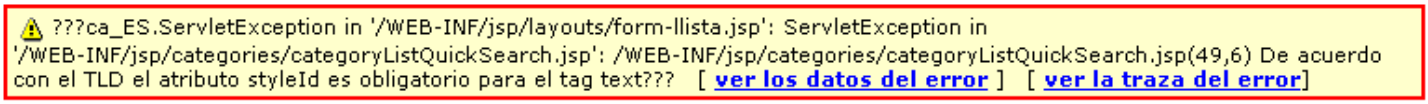
Tot seguit mostrem un exemple, però pot ser perfectament reemplaçada o modificada amb el disseny, estils propis de cada aplicació, exceptuant la manera de recuperar el contingut de l'excepció.

```
<%@ include file="/WEB-INF/jsp/includes/fwkTagLibs.jsp" %>
<%@page import="org.apache.struts.Globals"%>
<%@page import="net.gencat.ctti.canigo.services.exceptions.SystemException"%>
<%@page import="net.gencat.ctti.canigo.services.web.struts.ActionMessagesHelper"%>
<center>

<%request.setAttribute(Globals.ERROR_KEY,ActionMessagesHelper.
     buildSystemExceptionMessages((SystemException)request.getAttribute("exception"))); %>

 <html:messages id="msg" property="org.apache.struts.action.ERROR">
  <div class="error"
       style="margin-left: 10px; margin-right: 10px; margin-bottom: 3px; margin-top: 3px">
      <img src="<c:url value="/images/iconWarning.gif"/>" class="icon" alt="Warning" />
      <c:out value="${msg}" escapeXml="false"/>   
     [ <a href="javascript:showEL('errorDetails')">
          <b><fmt:message key="jsp.includes.see_error_details"/></b>
       </a> ]   
     [ <a href="javascript:showEL('stackTrace')">
          <b><fmt:message key="jsp.includes.see_stack_trace"/></b>
       </a>]<br>
      <html:messages id="msg2" property="net.gencat.ctti.canigo.services.exceptions.Level.WARNING">
         <div id="errorDetails" style="display: none">
           <br>
           <font color="#ff0000"><c:out value="${msg2}" escapeXml="false"/></font>
         </div>
      </html:messages>
      <html:messages id="msg3" property="net.gencat.ctti.canigo.services.exceptions.Level.ERROR">
         <div id="errorDetails" style="display: none"><font color="#00cc00">
           <<c:out value="${msg3}" escapeXml="false"/></font>
        </div>
```

Com gestionar les excepcions en la fase de renderitzat

```
        </html:messages>
        <html:messages id="msg4" property="net.gencat.ctti.canigo.services.exceptions.STACKTRACE">
           <div id="stackTrace" style="display: none">
              <br>
              <pre><c:out value="${msg4}" escapeXml="false"/></pre>
           </div>
        </html:messages>
        </div> 
</html:messages>
</center>
```

A l'implementar aquests canvis obtindrem un resultat tal com aquest:



Com es pot veure en la imatge, s'acaba mostrant amb detall la informació de l'excepció provocada.

NOTA: a partir de la versió 2.3.1 de Canigó, per a recuperar l'excepció a la jsp s'ha de canviar la sentència marcada en negreta i subratllada en l'exemple anterior per la sentència següent:

```
<% request.setAttribute(Globals.ERROR_KEY,
       (ActionMessages)request.getAttribute("exception")); %>
```

3. Altres excepcions

Tot i aquest mecanisme per capturar les excepcions de segona fase, hi ha excepcions que no mostren el motiu 'real' de l'error. En aquest cas, es pot introduir a la jsp que s'està renderitzant (no a la jsp d'error) una sentència `try/catch` per a què mostri la traça de l'error, de la manera següent:

```
<%@ include file="/WEB-INF/jsp/includes/fwkTagLibs.jsp" %>

<% try { %>

<fwk:form action="categories.do" styleId="actionForm" reqCode="search">
   <table width="100%">
      ...
   </table>
</fwk:form>

<% } catch (Exception e) {
       e.printStackTrace();
   }
%>
```

Posant la sentència `try` que englobi tot el codi de la jsp, aconseguirem que es mostri la traça d'error quan es produeixi una excepció, en la primera línia de la qual se'ns mostrarà a quina classe s'ha produït l'excepció.