

Gestió d'excepcions amb JSF 2

A qui va dirigit

Arquitectes i desenvolupadors d'aplicacions web basades en el framework Canigó 3.

Versió de Canigó

A partir de la versió 1.0.0 de Canigó 3 i que estigui utilitzant la versió 1.1.0 o superior del mòdul canigo.web.jsf.

Introducció

La versió 1.1.0 del mòdul canigo.web.jsf realitza un upgrade de la versió de JSF de 1.2 a 2.1 Aquest upgrade de versió introdueix millores i noves funcionalitats. D'entre aquestes millores cal destacar la possibilitat de definir un gestor d'excepcions i integrar-lo de forma fàcil en el cicle de vida de JSF.

Instal·lació

Cal haver creat una projecte Canigó 3 amb la versió 1.1.0 o superior del mòdul de canigo.web.jsf . El pom.xml del projecte per tant tindrà inclosa la següent dependència:

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd"
xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
...
<properties>
...
  <canigo.web.jsf.version>[1.1.0,1.2.0)</canigo.web.jsf.version>
...
</properties>
...
<dependencies>
  <dependency>
    <groupId>cat.gencat.ctti</groupId>
    <artifactId>canigo.web.jsf</artifactId>
    <version>${canigo.web.jsf.version}</version>
  </dependency>
</dependencies>
...
</project>
```

Gestió d'excepcions amb JSF 2

Qualsevol de les versions dins d'aquest rang especificat per el mòdul canigo.web.jsf importa JSF 2.1, que tal i com s'ha comentat en la introducció d'aquest HowTo permet definir un gestor d'excepcions integrat en el cicle de vida de JSF. A continuació es mostra un exemple de configuració:

faces-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<faces-config version="2.0" xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xi="http://www.w3.org/2001/XInclude"
  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-facesconfig_2_0.xsd">
  ...
  <factory>
    <exception-handler-factory>
      cat.gencat.formacio.exceptions.CustomExceptionHandlerFactory
    </exception-handler-factory>
  </factory>
  ...
</faces-config>
```

Aquest és el codi tant del Factory com del Handler:

CustomExceptionHandlerFactory.java

```
package cat.gencat.formacio.exceptions;

import javax.faces.context.ExceptionHandler;
import javax.faces.context.ExceptionHandlerFactory;

public class CustomExceptionHandlerFactory extends ExceptionHandlerFactory {

    private ExceptionHandlerFactory parent;

    // this injection handles jsf
    public CustomExceptionHandlerFactory(ExceptionHandlerFactory parent) {
        this.parent = parent;
    }

    @Override
    public ExceptionHandler getExceptionHandler() {

        ExceptionHandler handler = new CustomExceptionHandler(
            parent.getExceptionHandler());

        return handler;
    }
}
```

CustomExceptionHandler.java

```
package cat.gencat.formacio.exceptions;

import java.util.Iterator;
import java.util.logging.Level;
import java.util.logging.Logger;

import javax.faces.FacesException;
import javax.faces.application.FacesMessage;
```

Gestió d'excepcions amb JSF 2

```
import javax.faces.context.ExceptionHandler;
import javax.faces.context.ExceptionHandlerWrapper;
import javax.faces.context.FacesContext;
import javax.faces.event.ExceptionQueuedEvent;
import javax.faces.event.ExceptionQueuedEventContext;
import javax.servlet.ServletContext;

import org.springframework.web.context.support.WebApplicationContextUtils;

import cat.gencat.ctti.canigo.arch.core.exceptions.WrappedCheckedException;
import cat.gencat.ctti.canigo.arch.core.i18n.I18nResourceBundleMessageSource;

public class CustomExceptionHandler extends ExceptionHandlerWrapper {

    private I18nResourceBundleMessageSource i18n;

    private static final Logger log = Logger
        .getLogger(CustomExceptionHandler.class.getCanonicalName());

    private ExceptionHandler wrapped;

    CustomExceptionHandler(ExceptionHandler exception) {
        this.wrapped = exception;
        this.i18n = WebApplicationContextUtils
            .getRequiredWebApplicationContext(
                (ServletContext)
FacesContext.getCurrentInstance()

                .getExternalContext().getContext())
            .getBean(I18nResourceBundleMessageSource.class);
    }

    @Override
    public ExceptionHandler getWrapped() {
        return wrapped;
    }

    @Override
    public void handle() throws FacesException {

        final Iterator<ExceptionQueuedEvent> i =
getUnhandledExceptionQueuedEvents()
            .iterator();
        while (i.hasNext()) {
            ExceptionQueuedEvent event = i.next();
            ExceptionQueuedEventContext context =
(ExceptionQueuedEventContext) event
                .getSource();

            // get the exception from context
            Throwable t = context.getException();
            log.log(Level.SEVERE, "S'ha produït un error", t);

            WrappedCheckedException ex = getWrappedCheckedException(t);
            if (ex != null) {
                String errorMsg =
i18n.getMessage(ex.getExceptionDetails()
                    .getErrorCode());
                FacesContext.getCurrentInstance().addMessage(
                    "errorsZone",
```

Gestió d'excepcions amb JSF 2

```
        new
FacesMessage (FacesMessage.SEVERITY_ERROR, errorMsg,
                errorMsg));
    } else {
        FacesContext.getCurrentInstance().addMessage (
            "errorsZone",
            new
FacesMessage (FacesMessage.SEVERITY_ERROR, t
                .getMessage(),
t.getMessage()));
    }
}

private WrappedCheckedException getWrappedCheckedException(Throwable th)
{
    if (th instanceof WrappedCheckedException) {
        return (WrappedCheckedException) th;
    } else if (th == null || th.getCause() == null) {
        return null;
    } else {
        return getWrappedCheckedException(th.getCause());
    }
}
}
```

Aquest gestor tracta les excepcions produïdes dins el cicle de vida de JSF afegint informació al context de JSF per la seva visualització al usuari. Com es pot veure, en cas de tractar-se d'una excepció de tipus `WrappedCheckedException`, internacionalitza el missatge. I en cas contrari prepara el missatge rebut a l'excepció.

A continuació es pot veure un exemple de mètode d'un Managed Bean que encapsula les excepcions de capes inferiors en `WrappedCheckedException`'s, afegint un codi i18n que el handler d'excepcions internacionalitzarà per ser mostrat al usuari. En cas que el mètode de negoci hagi estat executat satisfactòriament afegeix al context de JSF un missatge d'informació:

CrudUsuariBean.java

```
@Component ("crudUsuariBean")
public class CrudUsuariBean {

    private static final Log log = LogFactory.getLog(CrudUsuariBean.class);

    @Autowired
    private I18nResourceBundleMessageSource i18n;

    @Autowired
    private TbUsuariService service;

    private TbUsuaris usuari = new TbUsuaris();

    ...

    public void altaUsuari() {
        if(log.isDebugEnabled()) {
            log.debug("Alta d'usuari");
        }
    }
}
```

Gestió d'excepcions amb JSF 2

```

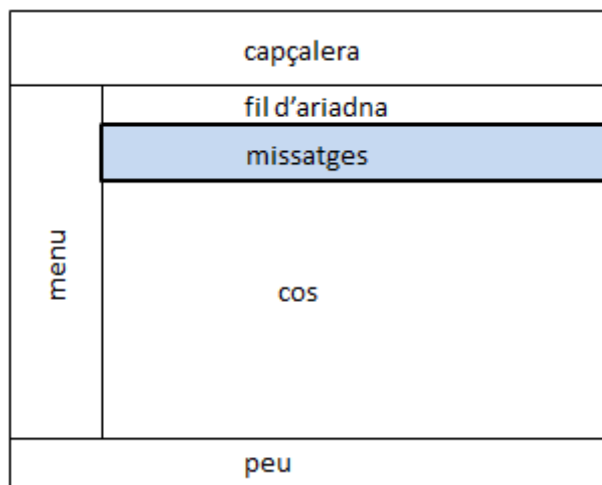
        this.usuari.setIdUsuari(null);
        try {
            service.altaUsuari(this.usuari);
        } catch (Exception e) {
            ExceptionDetails ed = new
ExceptionDetails("error.altaUsuari");
            throw new WrappedCheckedException(e, ed);
        }

        FacesContext.getCurrentInstance().addMessage(
            "infoZone",
            new FacesMessage(FacesMessage.SEVERITY_INFO,
                i18n.getMessage("info.altaUsuari"),
                null));
    }

    ...
}

```

Per la visualització de missatges al usuari, tant d'errors com d'informació o warnings, s'ha reservat una zona dins el template:



Aquest és el codi del template per el layout de les pàgines. En la figura anterior s'han destacat les parts principals.

template.jsf

```

<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:rich="http://richfaces.org/rich"
    xmlns:h="http://java.sun.com/jsf/html">
    <h:head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />

```

Gestió d'excepcions amb JSF 2

```
<ui:insert name="title">
    <title><ui:insert
name="title">#{msg.welcomeMessage}</ui:insert></title>
    <link rel="stylesheet"
href="#{facesContext.externalContext.requestContextPath}/css/import.css"
type="text/css" />
    <script type="text/javascript"
src="#{facesContext.externalContext.requestContextPath}/js/dgac/uiEvents.class
.js"></script>
    <script type="text/javascript"
src="#{facesContext.externalContext.requestContextPath}/js/dgac/innerNavigatio
n.class.js"></script>
    <script type="text/javascript"
src="#{facesContext.externalContext.requestContextPath}/js/dgac/globals.js"></
script>
    <script type="text/javascript"
src="#{facesContext.externalContext.requestContextPath}/js/dgac/guia-
aplicacions.js"></script>
    </ui:insert>
</h:head>
<h:body>
    <div id="wrapper">
        <div id="header">
            <div id="languages">
                <ui:insert name="navigation" id="languages">
                    <ui:include
src="/views/includes/idiomes.jsf" />
                </ui:insert>
            </div>
            <div id="corporate">
                <ui:insert name="logo">
                    <ui:include
src="/views/includes/logo.jsf" />
                </ui:insert>
            </div>
            <div id="loginPoint">
                <ui:insert name="loginInfo">
                    <ui:include
src="/views/includes/logininfo.jsf" />
                </ui:insert>
            </div>
            <div id="navTools">
                <ui:insert name="mapaWeb">
                    <ui:include
src="/views/includes/mapaweb.jsf" />
                </ui:insert>
            </div>
        </div>
        <div id="headerSection">
            <ui:insert name="header">
                <ui:include src="/views/includes/header.jsf"
/>
            </ui:insert>
        </div>
        <div id="navMain">
            <ui:insert name="navegacioPrincipal">
                <ui:include
src="/views/includes/navegacioPrincipal.jsf" />
            </ui:insert>
        </div>
    </div>
</hr />
<div id="contentWrapper" class="withNav">
```

Gestió d'excepcions amb JSF 2

```

        <div id="nav">
            <ui:insert name="menu">
                <ui:include
src="/views/includes/menu.jsf" />
            </ui:insert>
        </div>
        <div id="navSkills"></div>
        <div id="breadCrumb">
            <ui:insert name="ariadna">
                <ui:include
src="/views/includes/ariadna.jsf" />
            </ui:insert>
        </div>
        <div id="nav2">
            <ui:insert name="pestanyes">
                <ui:include
src="/views/includes/pestanyes.jsf" />
            </ui:insert>
        </div>
        <div id="contentMain">
            <span class="dgac">
                <ui:insert name="missatges">
                    <ui:include
src="/views/includes/missatges.jsf" />
                </ui:insert>
                <ui:insert name="body"/>
            </span>
        </div>
    </div>
    <div id="footer">
        <div id="siteInfo">
            <ui:insert name="footer">
                <ui:include
src="/views/includes/footer.jsf" />
            </ui:insert>
        </div>
    </div>
</h:body>
</html>

```

Com es pot veure la zona per mostrar missatges al usuari s'ha afegit a sobre del div per el contingut (body). El codi de la pàgina JSF per els missatges és el següent:

missatges.jsf

```

<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:ui="http://java.sun.com/jsf/facelets"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:a4j="http://richfaces.org/a4j"
xmlns:rich="http://richfaces.org/rich"
xmlns:c="http://java.sun.com/jstl/core">

    <h:outputText id="errorsZone" />
    <rich:message for="errorsZone" style="color: red" />

    <h:outputText id="warningsZone" />

```

Gestió d'excepcions amb JSF 2

```
<rich:message for="warningsZone" style="color: orange" />  
  
<h:outputText id="infoZone" />  
<rich:message for="infoZone" style="color: green" />  
  
</html>
```

Aquesta és una versió molt bàsica, es podrien afegir estils i la possibilitat de visualitzar la traça per els errors.

A continuació es mostra una captura d'exemple de visualització d'un missatge d'error amb la gestió d'excepcions exposada en aquest HowTo:

The screenshot shows the CSCanigó web application interface. At the top left is the logo 'CSCanigó'. On the right, there are language links: 'Català', 'Castellà', and 'Anglès'. Below the header is a navigation menu with 'Plantilla Canigó' and a breadcrumb trail: 'Mainteniments > Tests > Administració'. The 'Administració' menu item is selected. A red error message is displayed: 'S'ha produït un error donant d'alta l'usuari'. Below the message is a form titled 'Alta d'usuari' with the following fields: 'Nom' (input with value 'Nom1'), 'Cognoms' (input with value 'Cognom1'), 'Telèfon' (input with value '933333333'), 'eMail' (input with value 'nom1.cognom1@ger'), and 'Data de naixement' (calendar input with value '27/11/2013'). At the bottom of the form is an 'Alta' button. The footer contains the text: 'Avis legal | Sobre el web | © 1995-2010 Generalitat de Catalunya |'.

Amb aquesta gestió d'errors no es redirigeix a una plana d'error genèrica, sense perdre per tant el flux dins l'aplicació ni la informació de la plana web que l'ha generat.