

3.2 Gestion des exceptions



Relu

Relecture RB faite le 17/02/2011

Sommaire :

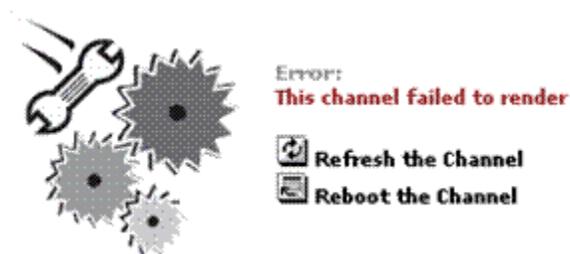
- Généralités
- Configuration
- Implémentations disponibles
- Exemple
- Qui reçoit les rapports d'exception ?
- Vue utilisée pour les rapports d'exceptions
- Redémarrage de l'application.
- Utilisation de plusieurs vues d'exceptions
- Ne pas envoyer de courrier électronique pour certaines exceptions

Généralités

Même dans une application mûre, il n'est de cas pour lesquels une exception ne puisse se produire. *esup-commons* offre une gestion des exceptions qui évite au développeur de recevoir le message traditionnel de *Tomcat* :

```
HTTP Status 500 -  
-----  
type Exception report  
message  
description The server encountered an internal error () that prevented it from fulfilling this request.  
exception  
org.hibernate.TransactionException: JDBC rollback failed  
org.hibernate.transaction.JDBCTransaction.rollback(JDBCTransaction.java:170)  
org.esupportail.commons.dao.HibernateThreadData.closeSession(HibernateThreadData.java:81)  
org.esupportail.commons.dao.HibernateUtils.close(HibernateUtils.java:118)  
org.esupportail.commons.web.servlet.FacesServlet.service(FacesServlet.java:322)  
javax.servlet.http.HttpServlet.service(HttpServlet.java:802)  
org.apache.myfaces.webapp.filter.ExtensionsFilter.doFilter(ExtensionsFilter.java:97)  
org.apache.myfaces.webapp.filter.ExtensionsFilter.doFilter(ExtensionsFilter.java:144)  
  
note The full stack trace of the root cause is available in the Apache Tomcat/5.5.17 logs.  
-----  
Apache Tomcat/5.5.17
```

Ou encore celui de *uPortal* :



esup-commons permet de remonter les exceptions de manière propre, par exemple :

Exception report

Restart the application

The following exception was thrown:

```
Application esup-repository
Version 0.11.0
Server 148.60.10.37 (caillou.ifsic.univ-rennes1.fr)
Date jeu. 25-01-2007 12:17:43
User unknown
Portal unknown
Client 148.60.10.131 (paubry.ifsic.univ-rennes1.fr)
Query string unknown
User agent Mozilla/5.0 (Windows; U; Windows NT 5.1; en-GB; rv:1.8.0.9) Gecko/20061206 Firefox/1.5.0.9
```

Exception details

```
Name NoSuchMethodException
Message org.esupportail.repository.web.controllers.AboutController.throwException()
Short stack javax.faces.FacesException: Error calling action method of component with id _idjsp25:_idjsp26
trace caused by: javax.faces.el.EvaluationException: Exception while invoking expression #{aboutController
caused by: java.lang.NoSuchMethodException: org.esupportail.repository.web.controllers.AboutContro

javax.faces.FacesException: Error calling action method of component with id _idjsp25:_idjsp26
org.apache.myfaces.application.ActionListenerImpl.processAction(ActionListenerImpl.java:69)
javax.faces.component.UICommand.broadcast(UICommand.java:106)
javax.faces.component.UIViewRoot._broadcastForPhase(UIViewRoot.java:94)
javax.faces.component.UIViewRoot.processDecodes(UIViewRoot.java:136)
org.apache.myfaces.lifecycle.LifecycleImpl.applyRequestValues(LifecycleImpl.java:218)
```

Cette gestion des exceptions est une véritable aide au développeur, qui peut ainsi connaître tout le contexte d'exécution au moment où s'est produite l'erreur :

Request parameters

```
_idjsp25:_idjsp26 = [Test exception handling]
_idjsp25:_idcl = []
_idjsp25_SUBMIT = [1]
jsf_sequence = [2]
```

Hibernate properties

```
hibernate.bytecode.use_reflection_optimizer = [false]
hibernate.version = [3.2 cr2]
```

System properties

```
awt.toolkit = [sun.awt.windows.WToolkit]
catalina.base = [C:\devel\esup-repository\apache-tomcat-5.5.17]
catalina.home = [C:\devel\esup-repository\apache-tomcat-5.5.17]
catalina.useNaming = [true]
```

Request headers

```
accept-charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
accept-encoding: gzip,deflate
accept-language: en-gb,en;q=0.5
accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,tes
connection: keep-alive
```

Cookies

```
JSESSIONID = [84A501ACD249D482287B82F926D3622C]
repository-user =
[pascal.aubry@univ-rennes1.fr:faed3d904bd017b0101a7a795bfaaeff]
```

Session attributes

```
aboutController = [AboutController#2721032]
administratorsController = [AdministratorsController#31231651[em
userToDelete=null, paginator=FixedQueryHibernatePaginator#1838
currentPage=0]]]
deepLinkingRedirector =
[org.esupportail.repository.web.deepLinking.DeepLinkingRedirectorI
```

Configuration

Le gestionnaire d'exception se configure à l'aide du fichier `/properties/exceptionHandling/exceptionHandling.xml`, dans lequel on déclare la *bean exceptionServiceFactory*, à partir duquel sont produits les services de gestion des exceptions.

Implémentations disponibles

esup-commons offre plusieurs gestionnaires d'exceptions :

- **SafeExceptionHandlerImpl** permet seulement de logger les exceptions. Cette implémentation n'est pas recommandée, elle est en revanche utilisée par les autres implémentations lorsqu'une exception se produit lors du traitement d'une autre exception.
- **SimpleExceptionHandlerImpl** permet de logger les exceptions mais présente aussi une vue à l'utilisateur avec le contenu de l'erreur. Cette implémentation peut être utilisée en production.
- **EmailExceptionHandlerImpl** étend **SimpleExceptionHandlerImpl** en permettant l'envoi de l'erreur par courrier électronique à une adresse donnée.
- **CachingEmailExceptionHandlerImpl** étend **EmailExceptionHandlerImpl** en ajoutant des fonctionnalités de cache afin de ne pas envoyer plusieurs fois la même exception et ainsi prévenir des effets de *spam* (quand par exemple la base de données est indisponible, il n'est pas nécessaire d'envoyer plusieurs fois la même exception).

Exemple

On utilisera par exemple :

```
<bean id="exceptionServiceFactory"
      class="org.esupportail.commons.services.exceptionHandling.CachingEmailExceptionHandlerImpl"
      parent="abstractApplicationAwareBean">
  <property name="smtpService" ref="smtpService" />
  <property name="recipientEmail" value="{exceptionHandling.email}" />
  <property name="exceptionViews" >
    <map>
      <entry key="java.lang.Exception" value="go_exception" />
    </map>
  </property>
  <property name="logLevel" value="{exceptionHandling.logLevel}" />
  <property name="cacheManager" ref="cacheManager" />
</bean>
```

* La propriété **smtpService** est le service d'envoi des messages électroniques.

- La propriété **recipientEmail** est l'adresse mail où seront envoyés les messages d'erreur.
- La propriété **cacheManager** est le gestionnaire de cache.
- La propriété **logLevel** indique le niveau de log des rapports d'exception (**ERROR** par défaut).

Qui reçoit les rapports d'exception ?

La dernière implémentation (**CachingEmailExceptionHandlerImpl**) n'envoie les courriers électroniques qu'à l'adresse spécifiée dans la configuration *Spring*. Lorsque cette classe est étendue dans *esup-blank* pour, en plus d'une adresse fixée par configuration, envoyer les exceptions à une adresse fixée dans le code (celles des développeurs). Cela permet de remonter automatiquement les exceptions aux développeurs, qui peuvent ainsi suivre à distance les problèmes sur les applications déployées. Même lorsque la classe **CachingEmailExceptionHandlerImpl** est étendue, les exploitants de l'application disposent de la propriété **doNotSendExceptionReportsToDevelopers** pour activer ou désactiver cette fonctionnalité.

Vue utilisée pour les rapports d'exceptions

SimpleExceptionHandlerImpl (ainsi que les classes qui en héritent) présente une vue à l'utilisateur avec le contenu de l'erreur. Cette vue peut être configurée en utilisant la propriété **exceptionView** du *bean* **exceptionServiceFactory**. Si cette propriété n'est pas définie, alors la vue **/stylesheets/exception.jsp** de *esup-commons* est utilisée par défaut. La vue proposée par défaut présente des détails qui ne sont certainement pas utiles à l'utilisateur. Il convient avant de mettre en production une application de personnaliser la vue **/stylesheets/exception.jsp** (en la copiant dans le projet de l'application) ou bien d'en créer une autre (par exemple **/stylesheets/customizedException.jsp**) et d'indiquer au *bean* **exceptionServiceFactory** de l'utiliser (à l'aide de la propriété **exceptionView**).

Il est possible de copier la vue **/stylesheets/exception.jsp** de *esup-commons* dans votre projet pour l'adapter et configurer votre gestionnaire d'exception pour pointer vers cette nouvelle vue.

Exercice : Changer la vue des exceptions [Afficher l'énoncé](#)

Redémarrage de l'application.

La vue **/stylesheets/exception.jsp** présente à l'utilisateur un bouton permettant de redémarrer l'application. Ce bouton appelle la méthode **restart()** du contrôleur **exceptionController** déclaré dans le fichier de configuration **/properties/web/controllers.xml**.

Exemple de déclaration :

```
<bean id="exceptionController"
      class="org.esupportail.commons.web.controllers.ExceptionController"
      scope="session">
</bean>
```

La méthode **restart()** de **exceptionController** appelle la méthode **reset()** de tous les *beans* qui implémentent l'interface **Resettable**. En conséquence, tout *bean* (contrôleur, ...) qui doit être réinitialisé lors du redémarrage de l'application doit implémenter l'interface **Resettable**.

Exercice : Réinitialiser un contrôleur après une exception [Afficher l'énoncé](#)

Utilisation de plusieurs vues d'exceptions

Il est possible d'utiliser plusieurs vues d'exceptions en fonction de l'exception levée par l'application :

```
<property name="exceptionViews" >
  <map>
    <entry
      key="org.esupportail.commons.exceptions.WebFlowException"
      value="/stylesheets/webFlowException.jsp" />
    <entry
      key="org.esupportail.commons.exceptions.ConfigException"
      value="/stylesheets/configException.jsp" />
    <entry
      key="java.lang.Exception"
      value="/stylesheets/exception.jsp" />
  </map>
</property>
```

Ne pas envoyer de courrier électronique pour certaines exceptions

Il est possible de ne pas envoyer de courrier électronique pour certaines exceptions, en ajoutant la propriété suivante :

```
<property name="noEmailExceptions" >
  <list>
    <value>
      org.esupportail.commons.exceptions.WebFlowException
    </value>
  </list>
</property>
```