# JSF par la pratique : Erreur à eviter

par Faissal Boutaounte (http://faissal-boutaounte.developpez.com/) (Blog)

Date de publication : 04 April 2010

Dernière mise à jour :

Cet article a pour but de clarifier des points qui entrainent souvent les débutants JSF à commettre des erreurs en utilisant les différents types d'actions disponibles avec JSF.

# I - Des actions qui ne s'exécutent pas

Les actions sont un concept très fort dans JSF il permet d'associer un bouton ou un lien dans une page html a une action (une méthode) dans une classe JAVA (un managed bean). Ceci dit qu'il ya derrière un système très fort qui permet de gérer tous cela, l'ignorance de ce système (architecture de JSF) entrainera dans plusieurs cas des erreurs dans l'utilisation de ces outils. Partons d'un exemple très simple un bouton qui exécute une action, tous ce dont on a besoin pour ça c'est :

```
   package edu.faissal.jsf; public class SimpleBean
    { private String nom; public String getNom() { return nom; } public void setNom(String
nom) { this.nom
    = nom; } public String action(){ this.nom = return null; } }
```

```
  <%@ page language="java" pageEncoding="ISO-8859-1"%> <%@ taglib uri="http://java.sun.com/jsf/html"
       prefix="h"%> <%@ taglib uri="http://java.sun.com/jsf/core" prefix="f"%> <!DOCTYPE HTML
       PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"> <html> <head> </head> <body>
       <f:view>
Hello, <br> <h:form> <h:outputText value="#{nomDusimpleBean.nom}"></h:outputText><br/>
       <h:commandLink value="cliquez sur!" action="#{nomDusimpleBean.action}"></h:commandLink>
       </h:form> </f:view> </body></html>
```

```
  <managed-bean>
       <managed-bean-name>nomDusimpleBean</managed-bean-name>
       <managed-bean-class>
        edu.faissal.jsf.SimpleBean
       </managed-bean-class>
       <managed-bean-scope>request</managed-bean-scope>
  </managed-bean>
```

## I-A - Mauvaise utilisation de l'attribut rendered

Si on modifie le lien qui déclenche notre action, en ajoutant une condition pour l'attribut rendrer , qui l'affichera seulement si un paramètre nomer test est passer dans la requête notre code sera donc comme ça :

```
   <h:commandLink rendered="#{param['test'] == 'montrer'}"
        value="Vous pouvez clicker sur moi pour voir le changement !"
   action="#{nomDusimpleBean.action}"></h:commandLink>
```

En suite on appel la page en ajoutant un paramètre test=monter dans la requête, on clique sur le lien et rien ne se passe ! Oui l'action ne s'est pas exécutée ! Le diagnostique a faire lorsque une action ne s'exécute pas est de voir tout d'abord s'il n'y a pas des erreurs de validation et puisque notre exemple ne contient pas des règles de validation alors c'est pas ça notre problème.

Pour comprendre le problème il faut alors commencer par le cycle de vie JSF. Après le clique le navigateur enverra une requête http est envoyé au serveur et justement c'est la servlet FacesServlet qui s'occupe de répondre a cette requête. FacesServlet est ensuite responsable d'enchainer les six phase du cycle de vie de JSF :

1 Restore View
2 Apply Request Values
3 Process Validations
4 Update Models Values
5 Invoke Application
6 Render Response

Pour lancer une action JSF commence en premier lieu et suite à la restauration de la vue par vérifier si le bouton a été source de la soumission du formulaire. Pour faire ce la on parcourt l'arbre des composantes en appelant la méthode UIComponent.processDecodes(javax.faces.context.FacesContext context) :

```
public void processDecodes(FacesContext context)
    {
        if (context == null) throw new NullPointerException("context");
        if (!isRendered()) return;
        for (Iterator<UIComponent> it = getFacetsAndChildren(); it.hasNext();)
        {
            it.next().processDecodes(context);
        }
        try
        {
            decode(context);
        }
        catch (RuntimeException e)
        {
            context.renderResponse();
            throw e;
        }
    }
```

Cette méthode vérifie la propriété rendered (la ligne 2 ) si elle est égale à false elle ne continue pas sinon elle appelle processDecode des composantes filles ensuite decode(FacesContext facesContext, UIComponent uiComponent) qui elle appel la méthode queueEvent(javax.faces.event.FacesEvent event); pour mettre l'événement dans la queue afin qu'il puisse être exécuté plus tard dans la phase 5.

Normalement notre action sera exécutée lors de la 5ieme phase, mais après le click sur le lien, remarquez que la requête ne contient plus notre paramètre test, ce qui fait que notre lien va avoir l'attribut rendered égale à false donc il notre bouton ne va pas mettre l'événement qui lui est associé dans la queue des événement ce qui fait qu'il notre méthode ne sera pas exécutée.

## II - Mauvaise utilisation de l'attribut rendered

## III - Conclusion

La même erreur pourra se reproduire au cas ou vous utilisez un dataTable dont la liste est remplie selon un paramètre de la requête et dans ce cas une fois vous cliquez sur le lien JSF essayera de remplir la table à nouveau et y mettre les boutons mais puisque le paramètre n'existe plus dans la nouvelle requête alors la liste ne sera pas remplie et le bouton cliqué n'existera pas et ne pourra pas lancer l'action qui lui est associée. Il faudrai essayer au maximum de comprendre le " comment ça marche " de la plus part des chose qu'on utilise dans notre programmation pour eviter des erreurs simple mais qui peuvent être difficiles a résoudre.