

Une (petite) introduction à JSF

Note : Ce sujet de travaux pratiques est une adaptation d'un petit tutorial très connu¹.

1 Introduction

Le framework JSF (*Java Server Faces*) a pour ambition de faciliter l'écriture d'applications WEB en offrant d'une part, une architecture MVC et d'autre part, une approche composant qui rappelle les interfaces graphiques natives. L'objectif est de remplacer l'approche « *je traite la requête HTTP et je renvoie une page HTML* » par « *j'envoie des composants (boutons, champs de texte, etc..) et je réagis aux actions de l'utilisateur* ».

JSF est composé de trois parties :

- une servlet générique qui va traiter les dialogues,
- une librairie de balises destinée à mettre en place les composants graphiques,
- une librairie de balises destinée à gérer ces composants et le dialogue avec le serveur.

JSF existe en deux versions (1.1 et 1.2). Dans un souci de simplicité nous allons utiliser la version 1.1. La version 1.2 est difficile à utiliser avec Tomcat [**Nouveauté** : la version 2.0 vient de sortir.]

Quelques liens :

Site officiel de SUN <http://java.sun.com/javaee/javaserverfaces/>

API JSF 1.1 http://download.oracle.com/docs/cd/E17802_01/j2ee/j2ee/javaserverfaces/1.1_01/docs/ap

myFaces : l'implantation libre proposée par Apache <http://myfaces.apache.org/>

Des **cours** sur JSF :

- Un cours complet en français² (et pas seulement sur JSF),
- 176 très bons transparents en français sur JSF³,
- Une liste de cours sur le sujet⁴,
- Un site associé à un livre⁵.

2 Mise en place

Suivez les étapes dans l'ordre pour obtenir une petite application JSF :

1. Créez une application WEB dynamique vide en utilisant le plugin *WTP* de *Eclipse JEE*.
2. Téléchargez la **version 1.1.x** de *myfaces-core*⁶ (disponible aussi ici⁷) et placez les fichiers JAR dans le répertoire `WebContent/WEB-INF/lib`.
3. Téléchargez le fichier JAR de *commons-codec* (disponible ici⁸) et placez le dans le répertoire `WebContent/WEB-INF/lib`.
4. Téléchargez les archives de la *JSTL* (disponible aussi ici⁹) et placez les fichiers JAR dans le répertoire `WebContent/WEB-INF/lib`.

¹<http://www.exadel.com/tutorial/jsf/jsftutorial-kickstart.html>

²<http://www.jmdoudoux.fr/java/dej/chap-jsf.htm#jsf>

³<http://mbaron.developpez.com/javaee/jsf/>

⁴<http://www.jsftutorials.net/>

⁵<http://www.horstmann.com/corejsf/>

⁶<http://myfaces.apache.org/download.html>

⁷ref:ress-jsf

⁸ref:ress-jsf

⁹ref:ress-jsf

5. Créez le fichier WEB-INF/web.xml listé ci-dessous :

```
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
  version="2.4">
  <display-name>JSF Application</display-name>
  <description>Exemple JSF</description>

  <context-param>
    <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
    <param-value>server</param-value>
  </context-param>

  <context-param>
    <param-name>javax.faces.CONFIG_FILES</param-name>
    <param-value>/WEB-INF/faces-config.xml</param-value>
  </context-param>

  <!-- Faces Servlet -->
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup> 1 </load-on-startup>
  </servlet>

  <!-- Faces Servlet Mapping -->
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>*.jsf</url-pattern>
  </servlet-mapping>

</web-app>
```

6. Créez le fichier WEB-INF/faces-config.xml listé ci-dessous :

```
<?xml version="1.0"?>
<!DOCTYPE faces-config PUBLIC
  "-//Sun Microsystems, Inc.//DTD JavaServer Faces Config 1.1//EN"
  "http://java.sun.com/dtd/web-facesconfig_1_1.dtd">

<faces-config>

</faces-config>
```

7. Créez la page JSP index.jsp listée ci-dessous et testez son bon fonctionnement.

```
<html>
<body>
  <h1>Hello</h1>
</body>
</html>
```

3 Les premières pages JSF

Nous allons créer deux pages : la première (/pages/inputname.jsp) demande le nom de l'utilisateur et la seconde (/pages/greeting.jsp) lui souhaite la bienvenue.

3.1 Navigation

Nous allons indiquer cette séquence dans le fichier de configuration des JSF (`faces-config.xml`) :

```
...
<navigation-rule>
  <from-view-id>/pages/inputname.jsp</from-view-id>
  <navigation-case>
    <from-outcome>greeting</from-outcome>
    <to-view-id>/pages/greeting.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
...
```

3.2 Utilisation d'un javaBean

Pour permettre le mouvement de données entre les deux pages, nous allons prévoir un JavaBean `monpkg.Person` qui possède une propriété `name` de type `java.lang.String`.

Une fois ce JavaBean programmé, nous allons le déclarer dans le fichier de configuration pour indiquer à JSF qu'il a la charge de créer, de remplir et de rendre disponible les instances de ce JavaBean. Nous dirons par la suite que ce JavaBean est géré (*managed*) par JSF.

```
...
<managed-bean>
  <description>Une personne gérée par JSF</description>
  <managed-bean-name>person</managed-bean-name>
  <managed-bean-class>monpkg.Person</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
</managed-bean>
...
```

Vous pouvez lire cette présentation détaillée¹⁰ du fichier `faces-config.xml`.

3.3 Le formulaire

Nous pouvons maintenant créer la page JSF (dont le nom est `/pages/inputname.jsp`) qui va poser la question à l'utilisateur :

```
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>

<html>
  <head><title>titre de la page</title></head>
  <body>
    <f:view>
      <h1><h:outputText value="Un exemple pour JSF"/></h1>
      <h:form id="helloForm">
        <h:outputText value="Votre nom :"/>
        <h:inputText value="#{person.name}" />
        <h:commandButton action="greeting" value="Dites bonjour !" />
      </h:form>
    </f:view>
  </body>
</html>
```

¹⁰<http://www.horstmann.com/corejsf/faces-config.html>

Testez cette page en utilisant cette URL :

```
http://localhost:8080/votre-application/pages/inputname.jsf
```

En ajoutant l'extension `.jsf` vous donnez la main à la servlet qui va gérer l'enchaînement des étapes pour répondre à votre requête. Regardez le code source HTML de la page renvoyée. Le framework JSP a préparé les données cachées qui vont lui permettre de traiter la soumission du formulaire.

La notation `#{person.name}` est propre à JSF. Elle permet d'utiliser directement les propriétés d'un JavaBean géré par JSF. Cette notation est proche mais différente des EL de la JSTL. La version 1.2 du framework JSP propose un rapprochement entre ces deux langages d'expression.

Le fichier configuration de JSF permet également d'initialiser certaines propriétés des javaBeans. Essayez le code ci-dessous :

```
...
<managed-bean>
  <description>Une personne gérée par JSF</description>
  <managed-bean-name>person</managed-bean-name>
  <managed-bean-class>monpkg.Person</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
  <managed-property>
    <property-name>name</property-name>
    <property-class>java.lang.String</property-class>
    <value>JavaJoe</value>
  </managed-property>
</managed-bean>
...
```

3.4 La page de bienvenue

Nous pouvons maintenant donner la page de bienvenue :

```
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>

<html>
  <head><title>page de bienvenue</title></head>
  <body>
    <f:view>
      <h3>
        Bienvenue,
        <h:outputText value="#{person.name}" />
      </h3>
    </f:view>
  </body>
</html>
```

Testez le bon fonctionnement de la navigation entre ces pages.

3.5 Les widgets JSF

En vous servant d'une part de la javaDoc des bibliothèques de balises¹¹, et d'autre part de cette documentation¹², ajoutez à votre formulaire un menu permettant de choisir un statut entre enseignants et étudiants. Pour ce faire, vous devez

¹¹ref.doc-myfaces-tlddoc-1.1.5

¹²<http://www.exadel.com/tutorial/jsf/jsftags-guide.html>

1. programmer un nouveau bean (par exemple `monpkg.Statuts`) doté d'une propriété (par exemple `statuts` de type `java.util.Map`. ce bean va coder la liste des statuts possibles ;
2. déclarer ce bean dans le fichier de configuration de JSF (remarquez la portée application de l'instance) :

```
<!-- les statuts d'une personne -->
<managed-bean>
  <managed-bean-name>statuts</managed-bean-name>
  <managed-bean-class>monpkg.Statuts</managed-bean-class>
  <managed-bean-scope>application</managed-bean-scope>
  <managed-property>
    <property-name>statuts</property-name>
    <property-class>java.util.Map</property-class>
    <map-entries>
      <map-entry>
        <key>Professeur</key>
        <value>prof</value>
      </map-entry>
      <map-entry>
        <key>Étudiant</key>
        <value>etud</value>
      </map-entry>
    </map-entries>
  </managed-property>
</managed-bean>
```

3. utiliser le widget `<h:selectOneMenu>` pour mettre en place le menu.

Une fois cette opération terminée, améliorez l'aspect de votre formulaire en mettant en place une grille (`<h:panelGrid>`). Il suffit de déclarer le nombre de colonnes et les Widgets se mettent en place automatiquement dans chaque cellule.

Traitement des liens. Pour illustrer le traitement des liens dans les formulaires, nous allons ajouter à notre exemple un lien « Quitter » associé à l'action de même nom.

```
<h:commandLink action="quitter">
  <h:outputText value="Quitter"/>
</h:commandLink>
```

Dupliquez la page `/pages/greeting.jsp` en `/pages/byebye.jsp` et modifiez les fichiers de configuration de JSF pour indiquer cette nouvelle navigation. Vous remarquerez qu'un clic sur ce lien provoque la soumission du formulaire. En d'autres termes, les modifications effectuées dans les champs de texte ne sont pas perdues lors d'un clic. Regardez le code source HTML pour avoir une idée de la réalisation de cette fonction.

4 Les fichiers de messages

Nous pouvons facilement (comme dans la plupart des frameworks) délocaliser les messages afin de simplifier le code et de faciliter l'internationalisation du logiciel. Pour ce faire, nous allons créer le package `monpkg.bundle` et placer à l'intérieur le fichier `messages.properties` :

```
yourName=Donnez votre nom :
yourStatut=Votre situation :
sayHello=Dites bonjour !
```

Pour utiliser ces messages dans vos pages JSF, il faut déclarer ces messages sous la forme d'un `javaBean` et les utiliser :

```
...  
<f:loadBundle basename="monpkg.bundle.messages" var="msg"/>  
...  
<h:outputText value="#{msg.yourName}"/>  
<h:inputText value="#{person.name}" />
```

Vous trouverez plus d'information sur l'internationalisation dans ce guide¹³.

5 Validation des données

Nous allons maintenant exploiter les possibilités de validation offertes par JSF. Commencez par utiliser l'attribut `required` dans vos champs de texte et vérifiez son bon fonctionnement.

```
<h:inputText id="name" value="#{person.name}" required="true" />
```

Pour que les messages d'erreur liés à la validation soient affichés, nous allons ajouter dans notre formulaire la clause `h:messages`. Vérifiez le fonctionnement de la validation et l'émission des messages d'erreur.

```
<h:messages style="color:red"/>
```

Nous pouvons poser des contraintes plus intéressantes avec la clause `f:validateLength`

```
<h:inputText id="name" value="#{person.name}" required="true">  
  <f:validateLength minimum="3" maximum="10"/>  
</h:inputText>
```

Continuons en ajoutant à notre bean `Person` une nouvelle propriété `age` qui est du type `java.lang.Integer`. Ajoutez à votre formulaire le champ âge. Testez le comportement de ce nouveau champ. Vous pouvez également utiliser la clause `f:validateLength` pour limiter la taille de l'âge.

D'une manière plus générale, vous pouvez aussi ajouter des champs numériques et contrôler la saisie par la clause `f:convertNumber` et `f:validateLongRange`.

6 Navigation avancée

Les possibilités de navigation de JSF sont bien plus complètes que ce simple exemple. Lisez cette documentation¹⁴ pour une présentation rapide des possibilités.

7 Introduire un contrôleur

Pour étudier cette dernière (mais importante) possibilité, je vous propose de suivre un tutorial sur la mise en place d'un petit jeu avec JSF¹⁵.

¹³<http://www.laliluna.de/javaserver-faces-message-resource-bundle-tutorial.html>

¹⁴<http://forum.exadel.com/viewtopic.php?t=579>

¹⁵<http://www.exadel.com/tutorial/jsf/jsftutorial-guessnumber.html>