

## TP 5

### Développer des composants de la couche de présentation IHM avec Java et JSF2

L'objectif de ce TP est de comprendre la déclaration des *ManagedBean* afin de gérer les données des vues JSF en utilisant le langage JSF Expression Language (JSF EL).

#### 1. Les ManagedBean :

##### 1.1. Préambule :

Un **ManagedBean** (bean managé) est une classe Java annotée. Il exécute la logique métier (ou la délègue aux EJB, par exemple), gère la navigation entre les pages ainsi que l'état des données d'une page.

Une application JSF typique contient un ou plusieurs beans managés. Plusieurs pages peuvent partager le même bean managé.

Les données d'une page JSF sont stockées dans les attributs du bean managé qui, dans ce cas, est également appelé « **backing bean** ».

Un **backing bean** définit les données auxquelles est lié un composant de l'interface utilisateur (la cible d'un formulaire, par exemple).

Le langage EL (*Expression Language*) permet de faire le lien entre un composant vue JSF et le bean managé associé.

Dans une application JSF, les traitements liés directement à l'interface utilisateur sont écrits dans les *backing beans*. Ces derniers font appel à des EJB ou des classes Java ordinaires pour effectuer les traitements qui ne sont pas liés directement à l'interface utilisateur (exemple : la couche métier).

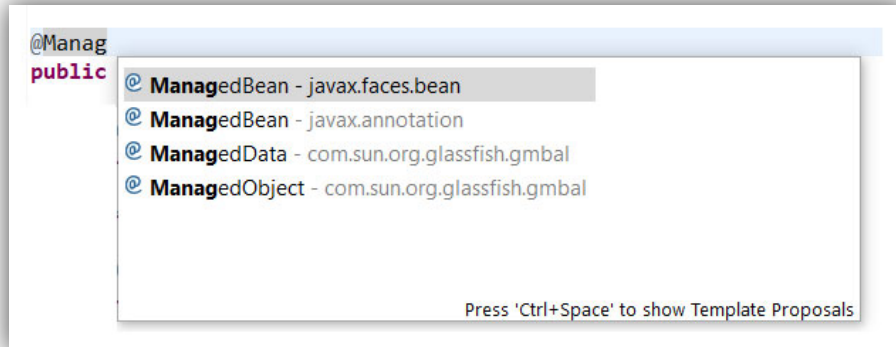
##### 1.2. Création d'un Bean managé « UserBean » :

Effectuez les opérations suivantes :

- Créez un package « **fr.doranco.tpjsf.beans** »
- Créez dans ce package une classe java **UserBean** avec les paramètres suivants : *nom, prenom, dateNaissance, genre, email, niveauDeService*.
- Ajoutez un constructeur sans arguments
- Ajoutez les *Getters* et *Setters* de tous les paramètres.

**Remarque :** A cette étape du codage, il s'agit d'une simple classe java que vous avez créée, et non d'un « *Managed Bean* » JSF2.

Ajoutez maintenant l'annotation **@ManagedBean** au-dessus de la déclaration de la classe : (utilisez l'auto-complétion pour afficher les commandes automatiquement)

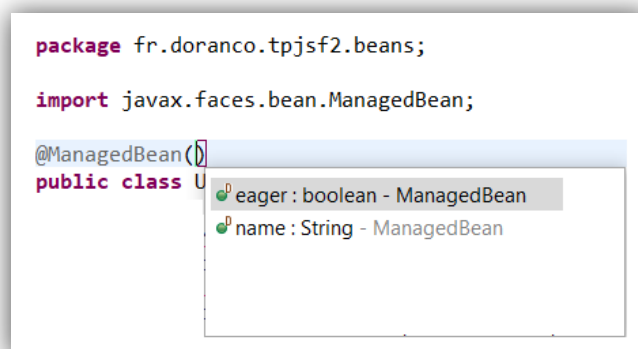


```
package fr.doranco.tpjsf2.beans;

import javax.faces.bean.ManagedBean;

@ManagedBean
public class UserBean {
```

**Info :** La présence de cette annotation à l'exécution enregistre systématiquement cette classe comme étant un bean géré JSF2. Cette classe sera scannée au démarrage de l'application, avant la première requête. En utilisation la documentation JSF2, expliquez les 2 paramètres possibles de cette annotation à savoir « **name** » et « **eager** ».



Donnez au bean le nom "userbean" comme indiqué ci-dessous :

```
package fr.doranco.tpjsf2.beans;

import javax.faces.bean.ManagedBean;

@ManagedBean(name="userbean")
public class UserBean {
```

**Questions :**

- Quel serait le nom « par défaut » du bean si nous n'avions pas donné le nom `userbean` ?
- Expliquez l'annotation `SessionScoped`. Quel est le scope par défaut ?

Imposez maintenant un scope **Session** comme indiqué ci-dessous.

```
package fr.doranco.tpjsf2.beans;

import javax.faces.bean.ManagedBean;
import javax.faces.bean.SessionScoped;

@ManagedBean(name="userbean")
@SessionScoped
public class UserBean {
```

**Question :** Le constructeur sans argument est-il nécessaire pour le bean managé ? Que se passe-t-il s'il n'y en a pas ?

**2. Utilisation de JSF Expression Language (JSF EL) :****2.1. Préambule :**

**JSF EL**, propose un mécanisme de communication entre les composants de la couche Vue et la logique applicative représentée par les *Managed Beans*.

JSF utilise le langage EL pour les fonctionnalités suivantes :

La réalisation des évaluations immédiates et différées des expressions.

- L'écriture des données vers un *Managed Bean*.
- Le déclenchement des méthodes d'un *Managed Bean*.
- La réalisation des opérations arithmétiques.

Afin d'accéder en écriture comme en lecture aux propriétés du *Managed Bean* **UserBean.java** définit par l'annotation `@ManagedBean(name="userbean")`, on utilise le nom du bean « `userbean` » suivi du nom de la propriété ou du nom de la méthode du *Managed bean* (`#userbean.xxx` : où xxx est le nom d'un paramètre managé ou une méthode).

**2.2. Mise en situation pratique du JSF EL :**

Créez un **main3.xhtml** depuis le **main2.xhtml** existant et modifiez la ligne suivante :

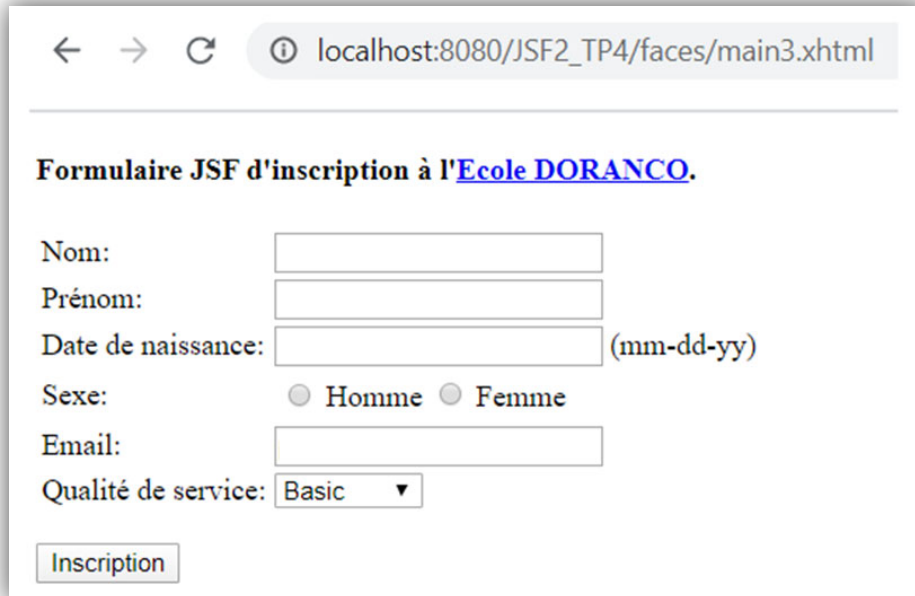
```
<h:form>
  <h2>Formulaire d'enregistrement Pour un stagiaire DORANCO</h2>
  <h:commandLink action="register2">
    <h:outputText value="Cliquez ici pour vous enregistrez..."></h:outputText>
  </h:commandLink>
</h:form>
```

Créez ensuite le fichier **register2.xhtml** depuis le fichier **register.xhtml** existant dans WebContent puis modifiez-le comme suit :

```
<f:view>
<head>
  <title>Un formulaire d'inscription JSF</title>
</head>
<body>
  <h:form>
    <h4>
      Formulaire JSF d'inscription à l'<a
        href="https://www.doranco.fr/candidature/parcours/41?">Ecole DORANCO</a>.
    </h4>
    <table>
      <tr>
        <td>Nom:</td>
        <td><h:inputText value="#{userbean.nom}" required="true" id="nom"></h:inputText>
          <h:message for="nom" /></td>
      </tr>
      <tr>
        <td>Prénom:</td>
        <td><h:inputText value="#{userbean.prenom}" required="true" id="prenom"></h:inputText>
          <h:message for="prenom" /></td>
      </tr>
      <tr>
        <td>Date de naissance:</td>
        <td><h:inputText value="#{userbean.datedenaissance}" id="datenaissance" required="true">
          </h:inputText> (mm-dd-yy) <h:message for="datenaissance" /></td>
      </tr>
      <tr>
        <td>Sexe:</td>
        <td><h:selectOneRadio value="#{userbean.genre}" required="true" id="genre">
          <f:selectItem itemLabel="Homme" itemValue="homme" />
          <f:selectItem itemLabel="Femme" itemValue="femme" />
        </h:selectOneRadio> <h:message for="genre" /></td>
      </tr>
      <tr>
        <td>Email:</td>
        <td><h:inputText value="#{userbean.email}" required="true" id="email"></h:inputText>
          <h:message for="email" /></td>
      </tr>
      <tr>
        <td>Qualité de service:</td>
        <td><h:selectOneMenu value="#{userbean.niveaudeservice}">
          <f:selectItem itemLabel="Basic" itemValue="basic" />
          <f:selectItem itemLabel="Medium" itemValue="medium" />
          <f:selectItem itemLabel="Premium" itemValue="premium" />
        </h:selectOneMenu></td>
      </tr>
    </table>
    <p>
      <h:commandButton value="Inscription" action="#" />
    </p>
  </h:form>
</body>
</f:view>
</html>
```

En utilisant le langage d'expression JSF, comme dans le fichier **register2.xhtml**, identifiez les expressions qui permettent d'accéder en écriture comme en lecture aux propriétés du bean **UserBean**.

L'exécution du projet sur le serveur Tomcat donne le rendu suivant :



← → ↻ ⓘ localhost:8080/JSF2\_TP4/faces/main3.xhtml

**Formulaire JSF d'inscription à l'Ecole DORANCO.**

Nom:

Prénom:

Date de naissance:  (mm-dd-yy)

Sexe: ☐ Homme ☐ Femme

Email:

Qualité de service: Basic ▼

**JSF EL** nous permet d'accéder en lecture comme en écriture aux propriétés du bean depuis la vue, en appelant le getter ou setter de chaque propriété.

#### Propriétés par défaut avec l'annotation **@ManagedProperty** :

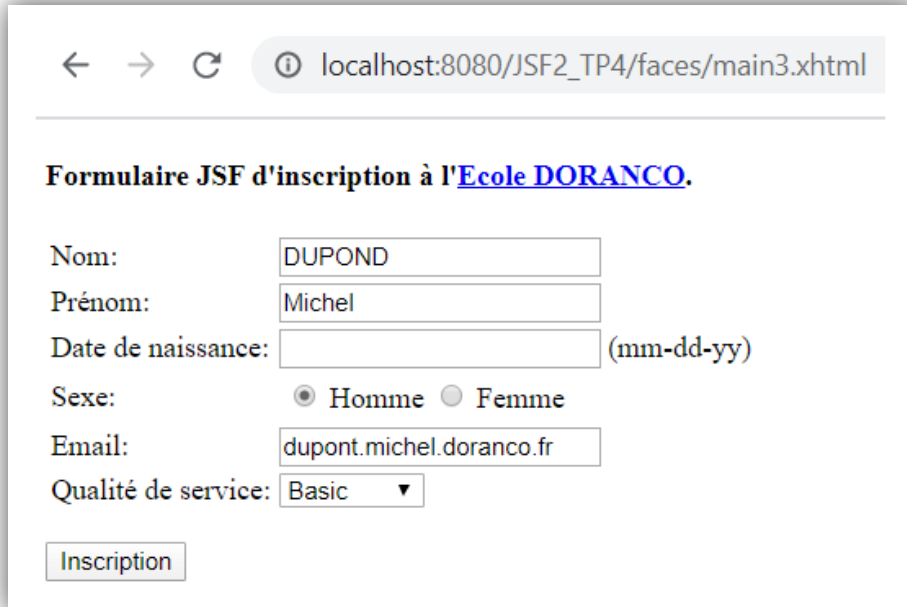
Initialisez les propriétés du Managed bean UserBean.java telle que :

- nom = "DUPONT"
- prenom = "MICHEL"
- genre = "homme"
- email = « dupont.michel@doranco.fr

Pour cela, utilisez pour cela l'annotation **@ManagedProperty** fournie par JSF2 et qui permet de manager et initialiser les propriétés de **UserBean**.

```
@ManagedProperty(value="DUPOND")  
private String nom;
```

L'exécution du projet donne le rendu suivant :



### 2.3. Intégration avec facelet :

Créez un fichier **main4.xhtml** depuis le fichier **main3.xhtml** et modifiez dans ce nouveau fichier la ligne suivante :

```
<h:commandLink action="register3">
  <h:outputText value="Cliquez ici pour vous enregistrez..."></h:outputText>
</h:commandLink>
```

Créez un fichier **register3.xhtml** depuis le fichier **register2.xhtml** et modifiez ce nouveau fichier afin d'obtenir un écran **facelet** permettant d'avoir le résultat suivant :

← → ↻ ⓘ localhost:8080/JSF2\_TP4/faces/main3.xhtml 🔍 ☆

  
Espace Multimédia  
www.doranco.fr

**Formulaire JSF d'inscription à l'[Ecole DORANCO](#).**

Prénom:

Nom:

Sexe: ☒ Homme ☐ Femme

Date de naissance:  (mm-dd-yy)

Email:

Qualité de service:  ▼

## Ajouter votre footer ou garder celui par défaut

Voici à quoi ressemble notre nouveau fichier **register3.xhtml** :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!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:f="http://java.sun.com/jsf/core"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets">

<head>
  <!-- <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" /> -->
  <title>Un formulaire d'inscription JSF</title>
</head>

<body>
  <!-- DECLARATION DU TEMPLATE DONT LA PAGE EST CLIENTE -->
  <ui:composition template="/resources/template/layout.xhtml">

    <!-- Redéfinition de la zone 'content' du layout -->
    <ui:define name="content">

      <f:view>
        <body>
          <h:form>
            <h4>
              Formulaire JSF d'inscription à l'<a
                href="https://www.doranco.fr/candidature/parcours/41?">Ecole DORANCO</a>.
            </h4>
            <table>
              <tr>
                <td>Nom:</td>
                <td><h:inputText value="#{userbean.nom}" required="true" id="nom"></h:inputText>
                  <h:message for="nom" /></td>
              </tr>
              <tr>
                <td>Prénom:</td>
                <td><h:inputText value="#{userbean.prenom}" required="true" id="prenom"></h:inputText>
                  <h:message for="prenom" /></td>
              </tr>
              ...
            </table>
            <p>
              <h:commandButton value="Inscription" action="#" />
            </p>
          </h:form>
        </body>
      </f:view>
    </ui:define>
  </ui:composition>
</body>
</html>
```

Expliquez le comportement du nouveau code dans le fichier **register3.xhtml** ayant ramené à avoir le nouveau rendu visuel précédent.

### 3. Conclusion

Dans ce TP, nous avons compris et mis en œuvre les annotations JSF 2 pour déclarer un bean managé, pour initialiser les propriétés du bean managé, pour définir la portée du bean managé, et enfin fait communiquer notre IHM avec notre bean managé en utilisant **JSF EL**.



# courage, bon taf!

