

## TP 7

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

L'objectif dans ce TP est de mettre en œuvre les différentes techniques de validation des composants JSF.

#### 1. Préambule :

La technologie JSF propose en standard les **Validateurs** et **Convertisseurs**. JSF fournit également une implémentation de la JSR-303, un moteur de validation centralisé au niveau des beans.

Avec JSF, nous pouvons mettre en œuvre une stratégie de validation "*programmative*" ou "*déclarative*".

JSF fournit 4 stratégies de validation :

- Composants standards de validation (attribut "**validator**" ou imbrication)
- Validation au niveau applicatif
- Composants personnalisés de validation
- Méthodes de validation dans le bean Managé

#### 2. Mise en œuvre des validations / conversions standards :

Créez un fichier register5 à partir du fichier register4.html effectué en TP 6 puis modifiez le nouveau fichier comme ci-après puis analysez le code ajouté.

```
<tr>
  <td>Date de naissance:</td>
  <td><h:inputText value="#{userbean.datedenaissance}"
    id="datenaissance" required="true">
      <f:convertDateTime pattern="dd-MM-yyyy"></f:convertDateTime>
    </h:inputText> (dd-mm-yyyy) <h:message for="datenaissance" /></td>
  <td><h:message for="datedenaissance" style="color:red"></h:message>
</tr>
```

Nous allons mettre maintenant en évidence une technique de validation et une technique de conversion fournis en standard par JSF.

Pour cela, effectuez les modifications suivantes dans **User** :

- Changez le type de la propriété *datedenaissance* en **Date** (*java.util.Date*).
- Mettez à jour le constructeur en modifiant le type de *dataedenaissance* en **Date** (*java.util.Date*).
- Régénérez le getter et le setter de l'attribut *datedenaissance* après avoir supprimé ses anciens getter et setter.

```
...  
...  
public Date datedenaissance;  
...  
...  
public User(String prenom, String nom, String genre, Date datedenaissance, String email, String niveaude service) {  
    this.prenom = prenom;  
    this.nom = nom;  
    this.genre = genre;  
    this.datedenaissance = datedenaissance;  
    this.email = email;  
    this.niveaude service = niveaude service;  
}  
...  
...  
public Date getDatedenaissance() {  
    return datedenaissance;  
}  
  
public void setDatedenaissance(Date datedenaissance) {  
    this.datedenaissance = datedenaissance;  
}
```

Effectuez maintenant les modifications suivantes dans **UserBean** :

- Changez le type de la propriété *datedenaissance* en **Date** (*java.util.Date*).
- Régénérez le getter et le setter de l'attribut *datedenaissance* après avoir supprimé ses anciens getter et setter.

```
public String datedenaissance;  
  
...  
...  
  
public Date getDatedenaissance() {  
    return datedenaissance;  
}  
  
public void setDatedenaissance(Date datedenaissance) {  
    this.datedenaissance = datedenaissance;  
}
```

- Ajoutez la méthode statique détaillée ci-dessous.

Ensuite mettez à jour la variable statique **userList** en modifiant la date utilisée en paramètre dans chaque **new User(...)**.

```
public static final Date stringToDate(String dateStr) {  
    SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");  
    Date date = null;  
    try {  
        date = formatter.parse(dateStr);  
    } catch (ParseException e) {  
        e.printStackTrace();  
    }  
    return date;  
}
```

```
private static final List<User> userList = new ArrayList<User>(Arrays.asList(  
  
    new User("Benoît", "Leclerc", "homme", stringToDate("27/10/1977"), "benoit@doranco.fr", "medium"),  
    new User("Paul", "Andrieux", "homme", stringToDate("12/06/1965"), "paul.augustin@doranco.fr", "medium"),  
    new User("Laura", "Treich", "femme", stringToDate("07/10/1987"), "laura@doranco.fr", "medium"),  
    new User("Nathalie", "Tango", "femme", stringToDate("17/07/1980"), "nathalie@doranco.fr", "premium"))  
);
```

Publiez vos changements puis testez. Vous devriez obtenir ceci :

#### Formulaire JSF d'inscription à l'Ecole DORANCO

Prénom:

Nom:

Sexe: ☒ Homme ☐ Femme

Date de naissance:  (dd/mm/yyyy) j\_idt7:datenaissance : «23-04-90» n'a pas pu être interprété en tant que date. Exemple : 17/04/2020

Email:

Qualité de service:

#### Listes des participants inscrits

Prénom	Nom	Email	Date de Naissance	Niveau de Service	Action	Action
Benoît	Leclerc	benoit@doranco.fr	Thu Oct 27 00:00:00 CET 1977	medium	<a href="#">Supprimer</a>	Supprimer
Paul	Andrieux	paul.augustin@doranco.fr	Sat Jun 12 00:00:00 CET 1965	medium	<a href="#">Supprimer</a>	Supprimer
Laura	Treich	laura@doranco.fr	Wed Oct 07 00:00:00 CET 1987	medium	<a href="#">Supprimer</a>	Supprimer
Nathalie	Tango	nathalie@doranco.fr	Thu Jul 17 00:00:00 CEST 1980	premium	<a href="#">Supprimer</a>	Supprimer

Modifiez **register5.xhtml** en mettant cette fois-ci un validateur sur les nom et prénom :

On utilisera le validateur standard suivant : `<f:validateLength ... />`

```
<tr>
  <td>Prénom:</td>
  <td><h:inputText id="prenom" value="#{userbean.prenom}" required="true">
    <f:validateLength minimum="2" maximum="10" />
  </h:inputText> <h:message for="prenom" style="color:red" /></td>
</tr>
<tr>
  <td>Nom:</td>
  <td><h:inputText id="nom" value="#{userbean.nom}" required="true">
    <f:validateLength minimum="2" maximum="10" />
  </h:inputText>
  <h:message for="nom" style="color:red"/>
</td>
</tr>
```

Publiez vos changements puis testez. Vous devriez obtenir ceci :

Prénom:  j\_idt7:prenom : erreur de validation. La longueur est supérieure à la valeur maximale autorisée, "10".  
Nom:  j\_idt7:nom : erreur de validation. La longueur est supérieure à la valeur maximale autorisée, "10".  
Sexe: ☒ Homme ☐ Femme  
Date de naissance:  (dd/mm/yyyy) j\_idt7:datenaissance : «23-04-90» n'a pas pu être interprété en tant que date. Exemple : 17/04/2020  
Email:   
Qualité de service:

#### Listes des participants inscrits

Prénom	Nom	Email	Date de Naissance	Niveau de Service	Action	Action
Benoît	Leclerc	benoit@doranco.fr	Thu Oct 27 00:00:00 CET 1977	medium	<a href="#">Supprimer</a>	Supprimer
Paul	Andrieux	paul.augustin@doranco.fr	Sat Jun 12 00:00:00 CET 1965	medium	<a href="#">Supprimer</a>	Supprimer
Laura	Treich	laura@doranco.fr	Wed Oct 07 00:00:00 CET 1987	medium	<a href="#">Supprimer</a>	Supprimer
Nathalie	Tango	nathalie@doranco.fr	Thu Jul 17 00:00:00 CEST 1980	premium	<a href="#">Supprimer</a>	Supprimer

```
<tr>
  <td>Prénom:</td>
  <td><h:inputText value="#{userbean.prenom}" required="true"
    id="prenom" /> <h:message for="prenom" style="color:red" /></td>
</tr>
<tr>
  <td>Nom:</td>
  <td><h:inputText id="nom" value="#{userbean.nom}" required="true">
    <f:validateLength minimum="2" maximum="10" />
  </h:inputText>
  <h:message for="nom" style="color:red"/>
</td>
</tr>
```

**Exercice :**

Modifiez **register5.xhtml** et ajoutez-y le champ « Nombre d'années d'expérience » qui permet la saisie d'un nombre entier limité compris entre 1 et 10.

On utilisera le validateur standard suivant : `<f:validateLongRange ... />`

Modifiez ensuite **User** puis **UserBean** pour tenir compte de ce nouveau paramètre nommé `nbAnneesExperience`. ⇒ Déclaration paramètre + getter/setter + modification constructeur + modification **userList**.

Publiez vos changements puis testez.

**3. Conclusion :**

Dans ce TP, nous avons mis en œuvre les validateurs fournis en standard dans Mojarra.

