

# TP3: JSF

## Objectif :

Le but de ce TP est de gérer une bibliothèque de livre (emprunt et retour)

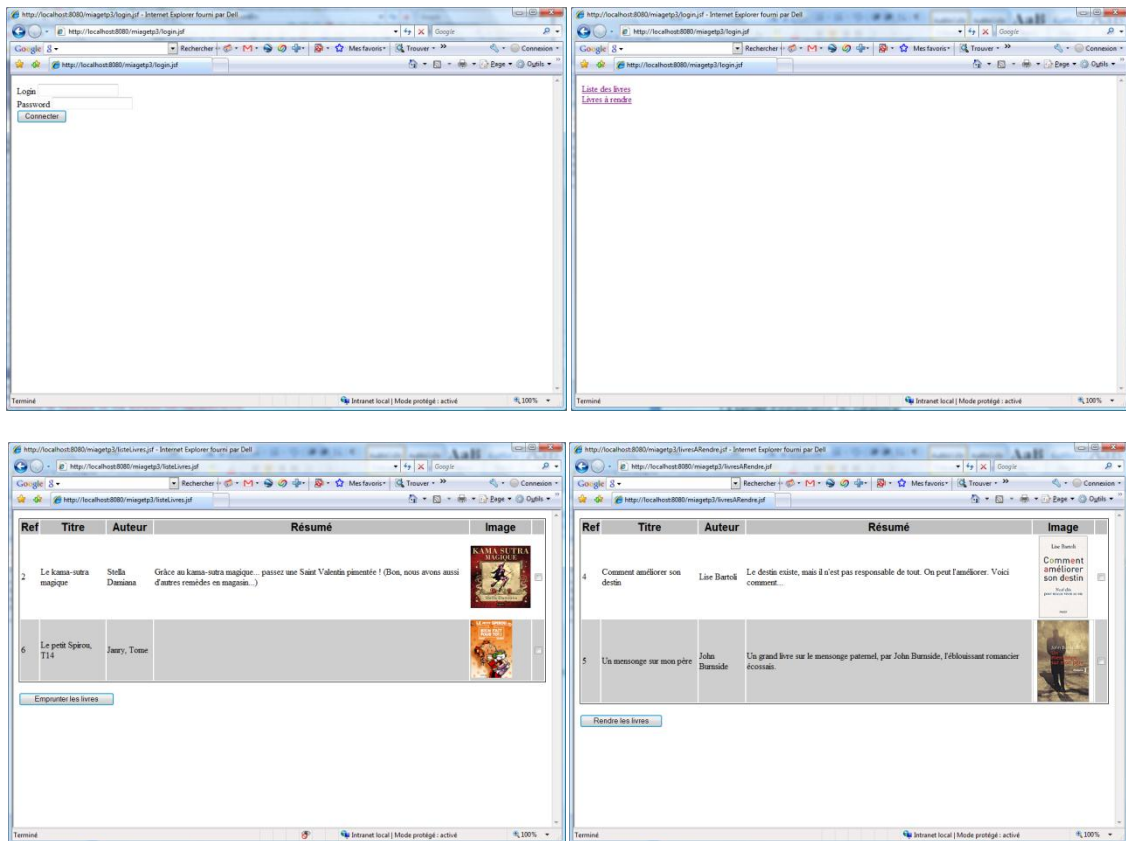
## Technologies :

Java EE 5 : JSF

IDE : Eclipse

Serveur : Tomcat

## Maquette :



## Déroulement du TP

- Créez un nouveau projet « web » avec Eclipse, avec comme configuration « **JavaServer Faces v2.0** » et « **Dynamic web module version 3.0** »
- Ajoutez dans WebContent/WEB-INF/lib les librairies nécessaires pour JSF
- Ajoutez dans un répertoire WebContent/css le fichier styles.css
- Ajouter dans le répertoire WebContent/WEB-INF le fichier web.xml

## Les beans

- Créez dans src un package bean
- Créez dans ce package les objets suivants : Catalogue, Livre, Personne, ListingEmprunteur

```

ListingEmprunteur
  ▢ listePersonne : HashMap<String, Personne>
  ● setListePersonne(HashMap<String, Personne>)
  ● getListePersonne()

```

```

Catalogue
  ▢ listeLivre : ArrayList<Livre>
  ● setListeLivre(ArrayList<Livre>)
  ● getListeLivre()

```

```

Livre
  ▢ id : String
  ▢ titre : String
  ▢ auteur : String
  ▢ resume : String
  ▢ image : String
  ▢ dateSortie : Date
  ▢ dateRetour : Date
  ▢ emprunteur : String
  ● getTitre()
  ● setTitre(String)
  ● getAuteur()
  ● setAuteur(String)
  ● getResume()
  ● setResume(String)
  ● getImage()
  ● setImage(String)
  ● getDateSortie()
  ● setDateSortie(Date)
  ● getDateRetour()
  ● setDateRetour(Date)
  ● getEmprunteur()
  ● setEmprunteur(String)
  ● setId(String)
  ● getId()

```

```

Personne
  ▢ nom : String
  ▢ prenom : String
  ▢ login : String
  ▢ password : String
  ● getNom()
  ● setNom(String)
  ● getPrenom()
  ● setPrenom(String)
  ● getLogin()
  ● setLogin(String)
  ● getPassword()
  ● setPassword(String)

```

## Les classes metier

Deux classes métier vont permettre la création du catalogue et de la liste des emprunteurs. Elles fourniront également les méthodes de récupération des Livres et des Personnes

- Créez dans src un package metier
- Créez dans ce package une nouvelle classe MetierPersonnes

La methode getPersonnes initialise la liste des emprunteurs référencés :

```

public static ListingEmprunteur getPersonnes() {
    if (personnesInstance == null) {
        personnesInstance = new ListingEmprunteur();

        Personne personne1 = new Personne();
        personne1.setLogin("test");
        personne1.setPassword("test");
        personne1.setNom("Test");
        personne1.setPrenom("Test");

        Personne personne2 = new Personne();
        personne2.setLogin("root");
        personne2.setPassword("root");
        personne2.setNom("Root");
        personne2.setPrenom("Root");

        HashMap<String, Personne> listePersonne = new HashMap<String, Personne>();
        listePersonne.put(personne1.getLogin() + personne1.getPassword(),
            personne1);
        listePersonne.put(personne2.getLogin() + personne2.getPassword(),
            personne2);

        personnesInstance.setListePersonne(listePersonne);
    }
    return personnesInstance;
}

```

La classe contient également une methode qui retourne une Personne à partir de son login et password. La signature de cette méthode est la suivante :

```

public static Personne getPersonne(String login, String password)

```

- Créez dans ce package une nouvelle classe MetierCatalogue

La methode getCatalogue initialise la liste des livres référencés :

```

public static Catalogue getCatalogue() {
    if (catalogue == null) {
        catalogue = new Catalogue();

        Livre livre1 = new Livre();
        livre1.setId("1");
    }
}

```

```

        livre1.setTitre("Légendes de Troy, T1");
        livre1.setAuteur("Keramidas, Arleston, Melanyn");
        livre1.setImage("http://static.fnac-static.com/multimedia/Images/FR/NR/3f/9b/25/2464575/1539-6.jpg");

        livre1.setResume("Une nouvelle série permettant à de grands dessinateurs d'explorer le vaste univers de Troy... Nicolas Keramidas ouvre le bal.");

        Livre livre2 = new Livre();
        livre2.setId("2");
        livre2.setTitre("Le kama-sutra magique");
        livre2.setAuteur("Stella Damiana");
        livre2.setImage("http://static.fnac-static.com/multimedia/Images/FR/NR/bb/e0/25/2482363/1539-6.jpg");

        livre2.setResume("Grâce au kama-sutra magique... passez une Saint Valentin pimentée ! (Bon, nous avons aussi d'autres remèdes en magasin...)");

        Livre livre3 = new Livre();
        livre3.setId("3");
        livre3.setTitre("La Conspiration du Temple");
        livre3.setAuteur("Steve Berry");
        livre3.setImage("http://static.fnac-static.com/multimedia/Images/FR/NR/42/2e/30/3157570/1539-1.jpg");

        livre3.setResume("Steve Berry au meilleur de sa forme pour son grand retour. Quand trembler fait du bien...");

        Livre livre4 = new Livre();
        livre4.setId("4");
        livre4.setTitre("Comment améliorer son destin");
        livre4.setAuteur("Lise Bartoli");
        livre4.setImage("http://static.fnac-static.com/multimedia/Images/FR/NR/b0/21/2f/3088816/1539-6.jpg");

        livre4.setResume("Le destin existe, mais il n'est pas responsable de tout. On peut l'améliorer. Voici comment...");

        Livre livre5 = new Livre();
        livre5.setId("5");
        livre5.setTitre("Un mensonge sur mon père");
        livre5.setAuteur("John Burnside");
        livre5.setImage("http://static.fnac-static.com/multimedia/Images/FR/NR/dd/98/2a/2791645/1539-6.jpg");

        livre5.setResume("Un grand livre sur le mensonge paternel, par John Burnside, l'éblouissant romancier écossais.");

        Livre livre6 = new Livre();
        livre6.setId("6");
        livre6.setTitre("Le petit Spirou, T14");
        livre6.setAuteur("Janry, Tome");
        livre6.setImage("http://static.fnac-static.com/multimedia/Images/FR/NR/51/08/26/2492497/1539-6.jpg");

        ArrayList<Livre> listeLivre = new ArrayList<Livre>();
        listeLivre.add(livre1);
        listeLivre.add(livre2);
        listeLivre.add(livre3);
        listeLivre.add(livre4);
        listeLivre.add(livre5);
        listeLivre.add(livre6);

        catalogue.setListeLivre(listeLivre);
    }
    return catalogue;
}

```

La classe contient également deux methodes qui retournent la liste des livres empruntés par une personne et la liste des livres non empruntés. Les signatures de ces méthodes sont les suivantes :

```
public static ArrayList<Livre> getLivreEmprunte(String login)
```

```
public static ArrayList<Livre> getLivreNonEmprunte()
```

## Page de login

Pour accéder à la bibliothèque les personnes doivent s'authentifier.

Vous créez donc un formulaire login.xhtml en utilisant les composants JSF suivants :

- `<core:view>`
- `<html:form>`

- `<html:outputText>`
- `<html:inputText>`
- `<html:inputSecret>`
- `<html:commandButton>`

Pour remplir le formulaire, vous utiliserez un managedBean.

Pour cela, créez un package managedBean qui contiendra une classe Login.java. Cette classe sera un bean qui aura deux attributs « login » et « password », les getters et setters de ces attributs, ainsi qu'une méthode « validate » qui vérifiera si la personne existe dans le listingEmprunteur et redirigera vers une page success ou error.

Ajoutez les annotations suivantes pour faire de votre classe un bean stocké en session

```
@ManagedBean (name="login")
@SessionScoped
```

Créez des fichiers error.xhtml et success. 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:core="http://java.sun.com/jsf/core"
      xmlns:html="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets">
  <body>
    <core:view>
      <html:outputText value="Error !" />
    </core:view>
  </body>
</html>
```

```
<?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:core="http://java.sun.com/jsf/core"
      xmlns:html="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets">
  <body>
    <core:view>
      <html:outputText value="Success !" />
    </core:view>
  </body>
</html>
```

➔ TESTEZ VOTRE APPLICATION

## Liste des livres à emprunter

Modifiez votre fichier succes.xhtml en y ajoutant des liens vers les pages listeLivres.xhtml et livresARendre.xhtml grâce à la balise :

- `<html:outputLink>`

Créez une page listeLivres.xhtml qui contiendra une datatable sur le modele :

```
<html:dataTable value="#{listeLivres.livres}" var="livre" rowClasses="row1,row2" headerClass="heading" styleClass="dataTable">
<html:column>
<core:facet name="header">
<core:verbatim>Ref</core:verbatim>
</core:facet>
<html:outputText value="#{livre.id}"/>
</html:column>

<html:column>
<core:facet name="header">
<core:verbatim>Titre</core:verbatim>
</core:facet>
<html:outputText value="#{livre.titre}"/>
</html:column>
...
```

Afficher les colonnes suivantes : Référence, titre, auteur, résumé et image (`<html:graphicImage>`).

Créez un managedBean ListeLivres qui aura une méthode :

```
/**
 * @return the livres
 */
public ArrayList<Livre> getLivres() {
    return MetierCatalogue.getLivreNonEmprunte();
}
```

## ➔ TESTEZ VOTRE APPLICATION

Ajoutez une colonne à votre page pour ajouter une checkbox sur chaque ligne de la datatable. Ajoutez dans votre bean une HashMap `<Long,Boolean>` qui contiendra pour chaque id de livre la valeur de sa checkbox

Ajoutez également un bouton pour envoyer le formulaire en faisant passer le login :

```
<html:column>
<html:selectBooleanCheckbox value="#{listeLivres.checked[livre.id]}"/>
</html:column>
...

<html:commandButton value="Emprunter les livres" action="#{listeLivres.emprunterLivres}"
    <core:setPropertyActionListener target="#{listeLivres.login}" value="#{login.login}"/>
</html:commandButton>
```

Ajoutez à votre managedBean les attributs suivants :

- `private HashMap <Long,Boolean> checked;`
- `private String login;`

Ajoutez les getters et setters de ces attributs ainsi que la méthode suivante :

```
public String emprunterLivres() {
    ArrayList<Livre> livres = MetierCatalogue.getCatalogue()
        .getListeLivres();
    for (Livre livre : livres) {
        if (checked.get(livre.getId()) != null && checked.get(livre.getId())) {
            livre.setEmprunteur("toto");
        }
    }
    return null;
}
```

## ➔ TESTEZ VOTRE APPLICATION

### Liste des livres à rendre

Créez une page livresARendre sur le même principe que listeLivres.xhtml. Vous modifierez les livres affichés pour n'afficher que les livres empruntés par la personne connectée. Vous modifierez le bouton « Emprunter les livres » par « Rendre les livres ». Créez également un managedBean « LivresEmpruntes » pour le traitement de cette action.

Pour récupérer le login dans le managedBean, vous pourrez utiliser la méthode suivante :

```
public String getLoginValue() {
    FacesContext facesContext = FacesContext.getCurrentInstance();
    Login login = (Login) facesContext.getExternalContext().getSessionMap().get("login");
    return login.getLogin();
}
```

## ➔ TESTEZ VOTRE APPLICATION

### Page d'inscription

- Dans la page success..xhtml ajoutez un lien vers une nouvelle page inscription.xhtml
- Créez la page inscription..xhtml qui contiendra 4 champs de saisie (nom, prénom, login et mot de passe) et un bouton Enregistrer. Elle sera à lier à un nouveau Bean Inscription.java.
- Dans ce bean la méthode d'enregistrement appellera une nouvelle méthode de MetierPersonne.java

```
public static void addPersonne (String login, String password, String nom, String prenom)
{
    Personne newPersonne = new Personne ();
    newPersonne.setLogin( login );
    newPersonne.setNom( nom );
    newPersonne.setPassword( password );
    newPersonne.setPrenom( prenom );
    getPersonnes().getListePersonne().put( login + password, newPersonne );
}
```

## ➔ TESTEZ VOTRE APPLICATION

### Rendered

- Ajoutez dans l'objet Personne un champ droitImage de type Boolean. Ce champ donnera à l'utilisateur concerné le droit de voir les images dans les pages de catalogue
- Ajoutez une checkbox pour saisir ce champ
- Modifier vos listes de livre pour afficher ou non l'image en fonction des droits utilisateurs. Pour cela vous pouvez utiliser l'attribut rendered (voir cours)

## ➔TESTEZ VOTRE APPLICATION

### Ajout de la date de naissance

- Ajoutez un attribut dateDeNaissance de type java.util.Date dans la classe Personne
- Ajoutez un attribut dans le bean Inscription.java de type java.util.Date

- Ajoutez un champ de saisie dans la page inscription.xhtml en utilisant un converter pour le lier au bean

➔ TESTEZ VOTRE APPLICATION

#### Ajout de la civilité

- Ajoutez un attribut civilité de type int dans la classe Personne (1 Mademoiselle ,2 Madame ou 3 Monsieur)
- Ajoutez un attribut dans le bean Inscription.java de type SelectItem
- Ajoutez un champ de saisie dans la page inscription.xhtml de type selectOneRadio pour en faire la saisie

#### Ajout d'un validateur

- Ajoutez un validator sur le champ de saisie login pour empêcher la saisie d'un login déjà existant

➔ TESTEZ VOTRE APPLICATION

#### Ajout d'un bouton annuler

- Ajoutez un bouton annuler à l'écran. Utiliser l'attribut immediate pour ne pas que les converter et validateur soient appelés

➔ TESTEZ VOTRE APPLICATION

#### Ajax

- Ajoutez à votre classe Personne un attribut age, ajoutez dans la page un affichage de cet attribut (en utilisant le tag outputText)
- Modifiez le getter de cet attribut pour qu'au lieu de retourner simplement l'attribut il le calcule à partir de la date de naissance
- Modifiez votre page pour qu'en cas de modification du champ date de naissance l'âge soit recalculé directement

➔ TESTEZ VOTRE APPLICATION

#### RichFaces

- Ajoutez à votre projet les librairies présente dans le package
- Modifiez votre web.xml pour y ajouter les paramètres spécifiques RichFaces (en utilisant celui fourni)
- Vous pouvez maintenant utiliser des composants RichFaces en déclarant les URI suivantes :

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:core="http://java.sun.com/jsf/core"
      xmlns:html="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:a4j="http://richfaces.org/a4j"
      xmlns:rich="http://richfaces.org/rich">
```

Vous trouverez à l'URL suivante les documentations des différents composants :

[https://docs.jboss.org/richfaces/latest\\_4\\_3/X/Component\\_Reference/en-US/html/](https://docs.jboss.org/richfaces/latest_4_3/X/Component_Reference/en-US/html/)

Pour tester les possibilités :

- Rendez vos datatable triables en utilisant les composants <rich:dataTable> et <rich:column>
- Modifiez la saisie de votre date de naissance en utilisant un composant <rich:calendar>