

# TP3: JSF

## Objectif :

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

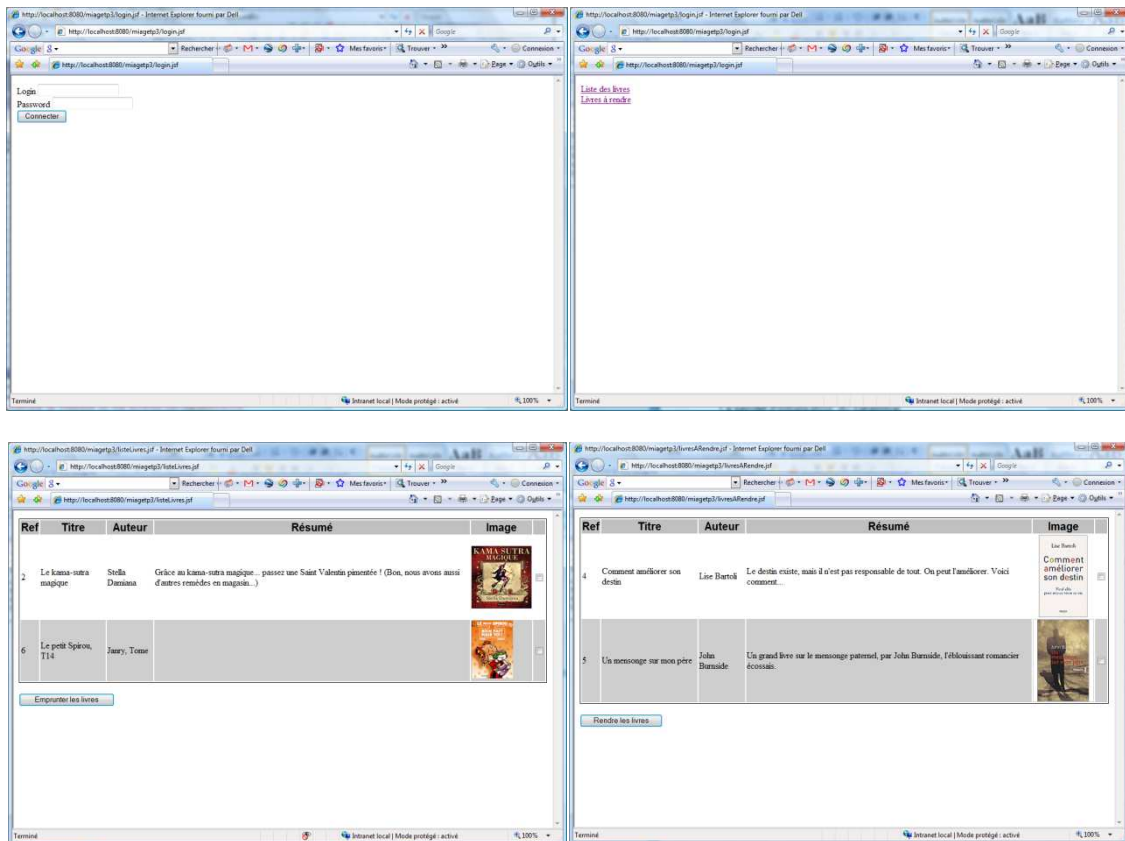
## Technologies :

Java EE 5 : JSF 1.2

IDE : Eclipse

Serveur : Tomcat

## Maquette :

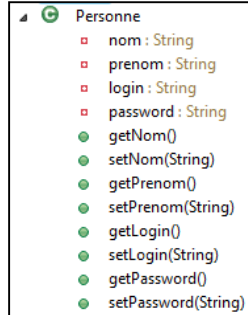
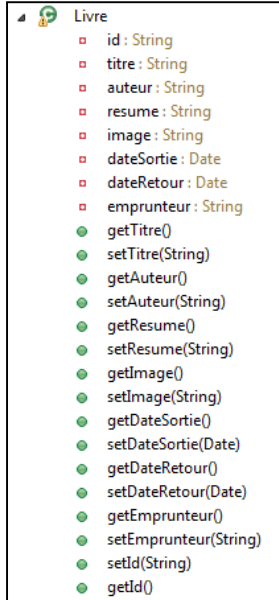
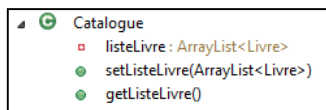
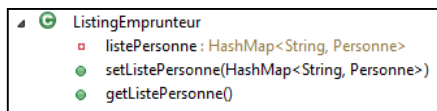


## Déroulement du TP

- Créez un nouveau projet « web » avec Eclipse, avec comme configuration « Dynamic web module 2.5 » et « JavaServer Faces v1.2 »
- Ajoutez les librairies nécessaires pour JSF et JSTL
- Ajoutez dans un répertoire WebContent/css le fichier styles.css

## Les beans

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



## 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://www.fnac.com/Img/V2/homes/ACC/2481489_110x150.png");
        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://www.fnac.com/Img/V2/homes/ACC/2502872_110x150.png");
        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://www.fnac.com/Img/V2/homes/ACC/2516350_110x150.png");
        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://www.fnac.com/Img/V2/homes/ACC/2494629_110x150.png");
        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://www.fnac.com/Img/V2/homes/ACC/2501551_110x150.png");
        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://www.fnac.com/Img/V2/homes/ACC/2514274_76x110.gif");

        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 méthodes 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 créerez donc un formulaire login.jsp en utilisant les composants JSF suivants :

- <core:view>
- <html:form>
- <html:outputText>
- <html:inputText>
- <html:inputSecret>
- <html:commandButton>

La page index.jsp redirigera vers cette page de login avec la ligne suivante :

```
<% response.sendRedirect("login.jsf"); %>
```

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 « valide » qui vérifiera si la personne existe dans le listingEmprunteur et redirigera vers une page success ou error.

N'oubliez pas d'ajouter à votre fichier faces-config.xml votre managed-bean et votre navigation-rule :

```
<managed-bean>
  <managed-bean-name>login</managed-bean-name>
  <managed-bean-class>managedBean.Login</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>

<navigation-rule>
  <from-view-id>/login.jsp</from-view-id>
  <navigation-case>
    <from-outcome>success</from-outcome>
    <to-view-id>/success.jsp</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>error</from-outcome>
    <to-view-id>/error.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
```

Créez des fichiers error.jsp et success.jsp :

```
<%@ page contentType="text/html" %>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="html" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="core" %>
<core:view>

  <html:outputText value="Error !" />

</core:view>
```

```
<%@ page contentType="text/html" %>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="html" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="core" %>
<core:view>

  <html:outputText value="Success !" />

</core:view>
```

➔ TESTEZ VOTRE APPLICATION

## Liste des livres à emprunter

Modifiez votre fichier succes.jsp en y ajoutant des liens vers les pages listeLivres.jsp et livresARendre.jsp grace aux balises :

- `<html:outputLink>`
- `<core:verbatim>`

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

```
<html:dataTable value="#{listeLivres.livres}" binding="#{listeLivres.dataTable}" 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}" binding="#{listeLivres.idLivre}">
  <core:attribute name="idLivre" value="#{livre.id}" />
</html:outputText>
</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<Livres> getLivres() {
    return MetierCatalogue.getLivresNonEmprunte();
}
```

Ajouter votre managedBean à votre fichier faces-config.xml

## ➔ TESTEZ VOTRE APPLICATION

Ajoutez une colonne à votre jsp pour ajouter une checkbox sur chaque ligne de la datatable, ajoutez également un champ caché qui contient le login connecté et un bouton pour envoyer le formulaire :

```
<html:column>
<html:selectBooleanCheckbox binding="#{listeLivres.checkbox}" />
</html:column>

...

<html:outputText value="" binding="#{listeLivres.login}">
  <core:attribute name="login" value="#{login.login}" />
</html:outputText>

...

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

Ajoutez à votre managedBean les attributs suivants :

- `private UISelectBoolean checkbox;`
- `private UIData dataTable;`
- `private HtmlOutputText login;`
- `private HtmlOutputText idLivre;`

Ajoutez les getters et setters de ces attributs ainsi que les méthodes suivantes :

```
public String getIdLivreValue() {
    return (String) idLivre.getAttributes().get("idLivre");
}

public String getLoginValue() {
    return (String) login.getAttributes().get("login");
}

public String emprunterLivres() {
    int size = this.dataTable.getRowCount();
    ArrayList<Livre> livres = MetierCatalogue.getCatalogue()
        .getListeLivre();
    for (int i = 0; i < size; i++) {
        this.dataTable.setRowIndex(i);
        if (this.checkbox.isSelected()) {
            for (Livre livre : livres) {
                if (getIdLivreValue().equals(livre.getId())){
                    livre.setEmprunteur(getLoginValue());
                }
            }
        }
    }
    return null ;
}
```

## ➔ TESTEZ VOTRE APPLICATION

### Liste des livres à rendre

Créez une jsp livresARendre sur le même principe que listeLivres.jsp. 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

### Compléments

Vous pouvez compléter votre application en utilisant les composants suivants :

Tomahawk Datatable ➔ possibilité de trier les résultats

Tomahawk TabbedPane ➔ (onglets) pour afficher d'un coté les livres empruntés et de l'autre les livres disponibles.

Tomahawk Calendar ➔ pour renseigner une date de retour sur les livres empruntés

...