

# NTIC - Examen n°1 : JSTL et JSF

## Objectif :

Le but de ce TP est de créer un site bancaire

## Technologies :

Java EE 5 : JSTL, JSF

IDE : Eclipse

Serveur : Tomcat

## Maquette :

Bienvenue sur le site de troc, nous vous proposons les services suivants :

- consultation de vos objets
- la possibilité d'effectuer des échanges

[Bonne navigation !](#)

**Accueil**

Login

Password

**Login**

[Consulter mes objets](#)  
[Effectuer un échange](#)  
[Retour au login](#)

**Success**

**Consultation du compte - test**

Date d'acquisition	Libellé	Valeur à Neuf
24/07/2012	Guitare	150,99 EUR
13/10/2012	Table	334,54 EUR
25/10/2012	4 chaises	122,21 EUR
04/06/2011	DVD	13,87 EUR
23/07/2005	TV écran plat full HD -1	120,00 EUR

[Retour au menu](#)

**consulterCompte**

**Effectuer un échange :**

Sélectionner un/plusieurs de vos objets à échanger :

Guitare  
Table  
4 chaises  
DVD  
TV écran plat full HD

root  
admin  
test

Sélectionner un client :

Sélectionner un/des objets à échanger :

**EffectuerEchange**

**Accueil de l'administration**

Sélectionner un client :

☐ Consulter le compte ☐ Effectuer un échange

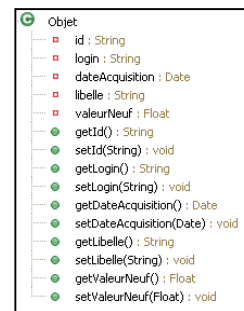
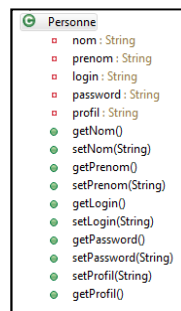
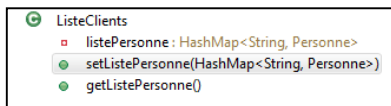
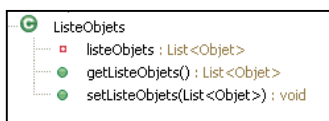
**Administration**

## Déroulement du TP

- Créez un nouveau projet « web » avec Eclipse, avec comme configuration « Dynamic web module 2.5 » et « JavaServer Faces v1.2 »
- **Nommez ce projet de la manière suivante : exam1\_XYYY (avec X = 1<sup>ère</sup> lettre du prénom et YYY = trois 1<sup>ères</sup> lettres du nom).**
- Ajoutez les librairies nécessaires pour JSF
- 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 : Objet, Personne, ListeClients, ListeObjets



### Les classes métier

Deux classes métier vont permettre la gestion des objets et des clients.

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

La methode getPersonnes initialise la liste des clients référencés :

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

        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");

        Personne personne3 = new Personne();
        personne3.setLogin("admin");
        personne3.setPassword("admin");
        personne3.setNom("Admin");
        personne3.setPrenom("Admin");
        personne3.setProfil("ADMIN");

        HashMap<String, Personne> listePersonne = new HashMap<String, Personne>();
        listePersonne.put(personne1.getLogin() + personne1.getPassword(),
            personne1);
        listePersonne.put(personne2.getLogin() + personne2.getPassword(),
            personne2);
        listePersonne.put(personne3.getLogin() + personne3.getPassword(),
            personne3);
        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)
```

Vous êtes libre de créer d'autres méthodes dans cette classe.

- Créez dans ce package une nouvelle classe MetierObjets

La methode getListeObjets initialise la liste des objets référencés :

```
public static ListeObjets getListeObjets() {
    if(listeObjets == null){
        listeObjets = new ListeObjets();
        Objet objet1 = new Objet();
        long maDate = new Date().getTime();
        objet1.setId("1");
        objet1.setLogin("test");
        objet1.setDateAcquisition(new Date(maDate-90000000001));
        objet1.setLibelle("Guitare");
        objet1.setValeurNeuf(150.99f);

        Objet objet2 = new Objet();
        objet2.setId("1");
        objet2.setLogin("test");
        objet2.setDateAcquisition(new Date(maDate-2000000000));
        objet2.setLibelle("Table");
        objet2.setValeurNeuf(334.54f);

        Objet objet3 = new Objet();
        objet3.setId("3");
        objet3.setLogin("test");
        objet3.setDateAcquisition(new Date(maDate-1000000000));
        objet3.setLibelle("4 chaises");
        objet3.setValeurNeuf(122.21f);

        Objet objet4 = new Objet();
        objet4.setId("4");
        objet4.setLogin("test");
        objet4.setDateAcquisition(new Date(maDate-450000000001));
        objet4.setLibelle("DVD");
        objet4.setValeurNeuf(13.87f);

        Objet objet5 = new Objet();
        objet5.setId("5");
        objet5.setLogin("test");
        objet5.setDateAcquisition(new Date(maDate-2300000000001));
        objet5.setLibelle("TV écran plat full HD");
        objet5.setValeurNeuf(-1120f);

        Objet objet6 = new Objet();
        objet6.setId("6");
        objet6.setLogin("root");
        objet6.setDateAcquisition(new Date(maDate-720000000001));
        objet6.setLibelle("Voiture");
        objet6.setValeurNeuf(-20354.54f);

        ArrayList<Objet> objets = new ArrayList<Objet>();
        objets.add(objet1);
        objets.add(objet2);
        objets.add(objet3);
        objets.add(objet4);
        objets.add(objet5);
        objets.add(objet6);

        listeObjets.setListeObjets(objets);
    }
    return listeObjets;
}
```

La classe contient également deux autres méthodes. Les signatures de ces méthodes sont les suivantes :

```
public static List<Objet> getObjets(String pLogin)
```

```
public static void setListeObjets(ListeObjets pListeObjets)
```

## Page d'accueil

La page d'accueil (index.jsp) présentera en quelques mots les services de la banque (utilisez pour cela les tags JSTL et un fichier externe Message\_fr.properties) elle proposera un lien pour accéder à la page de login.

## Page de login

Pour accéder à leur compte les personnes doivent s'authentifier.

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

- `<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 la liste des clients 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.

Créez des fichiers error.jsp et success.jsp.

## Consulter son compte

Modifiez votre fichier success.jsp en y ajoutant des liens vers les pages consulterCompte.jsp et effectuerVirement.jsp grace aux balises :

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

Créez une page consulterCompte.jsp qui contiendra une datatable pour présenter les lignes d'objets.

Afficher les colonnes suivantes : Date d'acquisition, Libellé, Valeur à neuf

Utilisez notamment les balises suivantes pour la mise en forme des dates et montant :

- `<core:convertDateTime>`
- `<core:convertNumber>`

Créez un managedBean Objets qui aura une méthode :

```
public ArrayList<Operation> getObjets() {
    String login = getLoginValue();
    return MetierObjets.getObjets(login);
}
```

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();
}
```

Ajouter votre managedBean à votre fichier faces-config.xml

### Effectuer un échange

Créez une jsp effectuerEchange.jsp qui contiendra la liste des objets, une liste des clients, un bouton pour charger la liste des objets du client, la liste des objets du client et un bouton de validation.

Pour les listes vous pourrez utiliser les balises suivantes :

```
<html:selectOneListbox value="#{ echangeBean.clientSelected}" >
  <core:selectItems value="#{ echangeBean.usersItems}" />
</html:selectOneListbox>
```

Le bouton de chargement des objets du client permettra de remplir la liste des objets du client avec ses objets (utilisez la notion de binding).

Votre méthode de validation du formulaire devra échanger les objets des 2 clients concernés.

### Administration

Pour créer la page d'administration de notre gestionnaire de troc, vous allez permettre la redirection vers la page administration.jsp quand l'utilisateur qui se connectera aura un profil « ADMIN » de renseigné.

La page d'administration proposera la liste des clients (comme sur la page d'échange) ainsi que 2 actions possibles (boutons radio) :

- Consulter le compte de la personne sélectionnée.
- Effectuer un échange à la place de la personne sélectionnée.

Pour créer des boutons radios, vous utiliserez les tags suivants :

- `<html:selectOneRadio>`
- `<core:selectItems>`

Vous modifierez le fonctionnement de la consultation des comptes et des échanges pour que ces pages prennent en compte non plus la personne connectée mais la personne sélectionnée dans le cas d'un utilisateur administrateur.

# BON COURAGE...