**Phase2 : Cours B\_62 Projet Oracle**

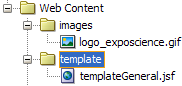
**Utiliser obligatoirement le fichier script.sql :**

**Remise au plus tard le jeudi le 16 mai 20013 , (10% de pénalité par jour de retard)**

\*\*\* Sauvegarder **régulièrement** sous **différentes versions** votre projet, car il est préférable de reprendre une version antérieure fonctionnelle que de s’acharner à débugger la présente (je parle par expérience)

1. Ajouter le répertoire <<image>> remis, sous View Controller/ public\_html
2. Créer une **ADF Page template** sous le dossier **Web Content** ( attention : créer un une **ADF Page templat** et **non** un ADF TaskFlow template) . Aidez-vous du document : Template.doc

**1-\*\***Lors de sa création, créer un dossier **template** où vous situerez votre ADF Page template nommé : **templateGeneral**



2- On y retrouve minimalement les informations suivantes :

Copyright Expo-sciences Bell Tous droits réservés.

Le logo (logo\_exposcience.gif)

**\*\*** Vous devez utiliser le **ViewControllerBundle** pour **tous les libellés, les messages et texte** dans votre application



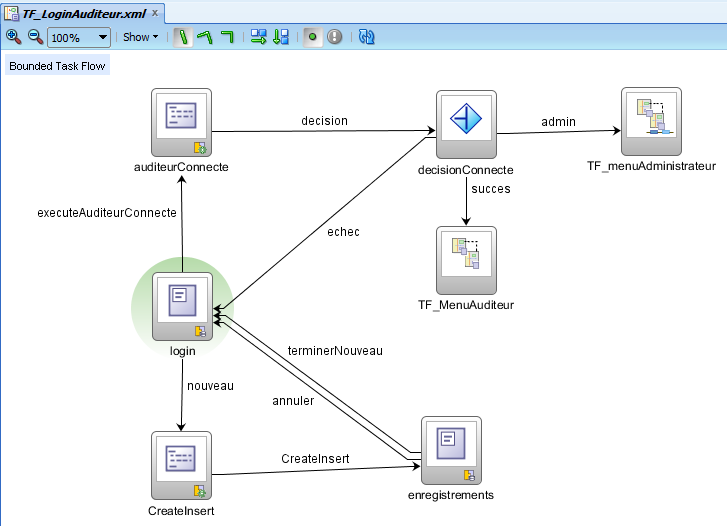
Créer une page **(.jsf)**  dans un **UNBOUD TASK\_FLOW**:

1. Page home.jsf: (**sous public\_html**) **accessible à tous**, cette page ne requiert aucune authentification. Utilisé le **templateGeneral**
2. **Structure à respecter pour votre projet** : **Cette structure vous assurera une meilleure lisibilité dans votre navigateur d’objets.**

**Créer d’abord :**

1. le **BOUND** TASK FLOW ( **directement sous WEB-INF**),nommez le **TF\_loginAuditeur qui utilisera des pages fragmentées (.jsff) et non pas des pages JSF**

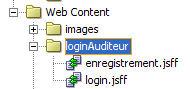




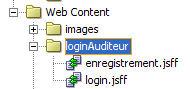
1. Lors de la création de vos pages fragmentées .jsff, créer celles-ci dans un dossier (**sous**  **public\_html**) portant **le même nom que le TaskFlow**

Exemple : pour le TaskFlow : **TF**\_**loginAuditeur**

Les deux pages fragmentées <<enregistrements.jsff>> et <<login.jsff>> sont dans un dossier nommé : **loginAuditeur**



Voici un aperçu des deux pages fragmentées (.jsff) à créer.

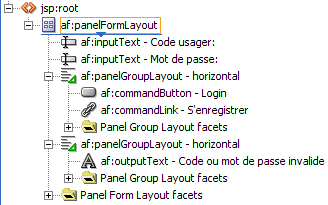


**Page : login.jsff**

**\*\*Important : toujours insérer un composant <<af :panelFormLayout >>à votre .jsff et ce pour toute les pages fragmentées (.jsff) que vous créerez**

****

**Structure de la page login.jsff:**

****

**Fonctionnalités demandées** pour login.jsff

Voir le fichier **<<phase2.swf >>** pour le fonctionnement de cette page

**Pour se loguer** :

Le code usager fait référence au champ codeauditeur de la table p\_auditeur.

Le Mot de passe fait référence au champ motdepasse de la table p\_auditeur.

Si l’usager n’est pas existant dans la table p\_auditeur faites afficher un message.

Le link **<<S’enregistrer>>** navigue ver la page entregistrement.jsff

et permet d’insérer un nouvel auditeur dans la table p\_auditeur

**Démarche suggérée** :

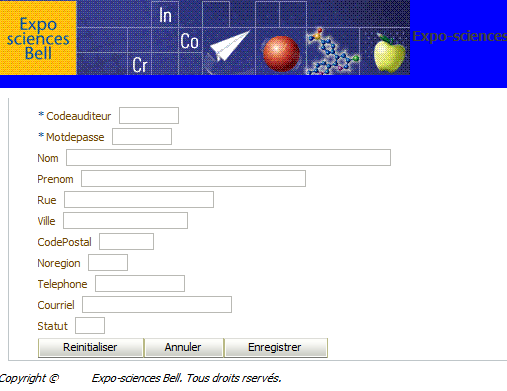
1. Créer une vue PAuditeursView paramétrée avec les champs <<CODEAUDITEUR>> et <<MOTDEPASSE>>. Référer à votre laboratoire Bottin fait dans la 1ère partie du cours.

Dans le fichier PAuditeurViewImpl :

Créer une méthode : public boolean auditeurConnecte(String paramCode, String paramMotDePasse)

Les paramètres passées à cette méthode sont les champs <<code usager>> et <<mot de passe>> de la page Login.jsff.

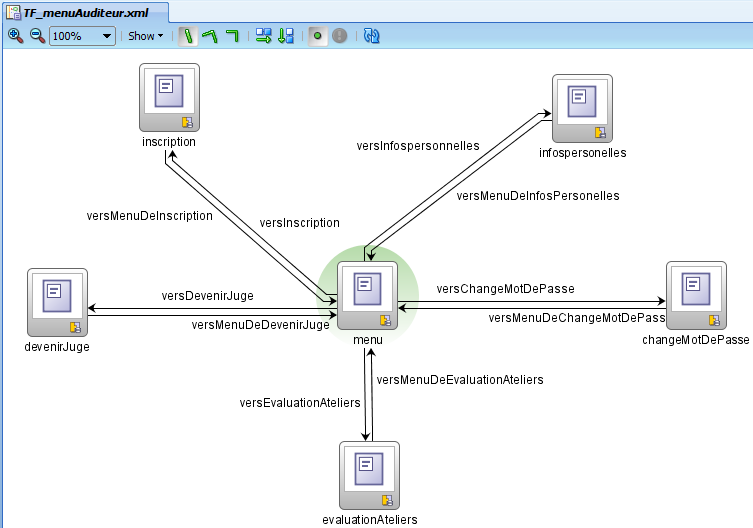
**<<entregistrement.jsff>>**

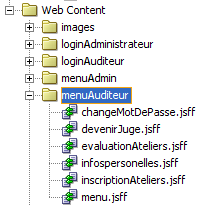


**Fonctionnalités demandées pour <<entregistrement.jsff>>**

* Code auditeur et Mot de passe **obligatoires**
* Formater ou valider: Code postal et téléphone
* Statut doit être implanté comme une liste fixe
* Le champ **no region** devrait être remplacé par le nom de la région soit une liste remplie dynamiquement (LOV) par la table p\_region
* Ajouter le champ CANDIDATJUGE
* Bouton Réinitialiser : vide les champs
* Bouton Annuler : retourne au login
* Bouton enregistrer : valide si le code Auditeur n’est pas déjà existant dans la table p\_auditeur et enregistre à la BD

Créer le **2ième BOUND TaskFlow** : nommez-le : **TF\_menuAuditeur**





Évaluation des ateliers sera développée **plus tard**

Menu.jsff



ChangeMotdePasse.jsff



Traiter les **4 situations suivantes et pour chacune afficher un message approprié à l'usager**:

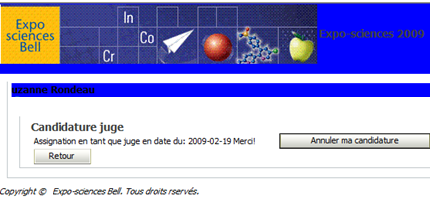
1- Mot de passe déjà existant

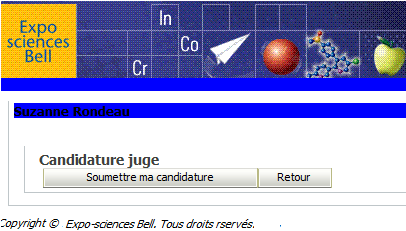
2- Les champs Nouveau et Confirmation Mot de passe sont obligatoires (ne peuvent pas être nuls)

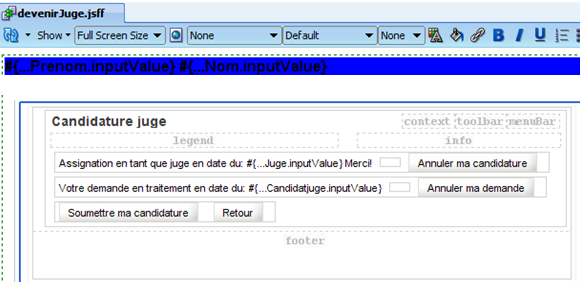
3- Non-correspondance entre les champs Nouveau et Confirmation mot de passe

4- Afficher: Nouveau mot de passe enregistré

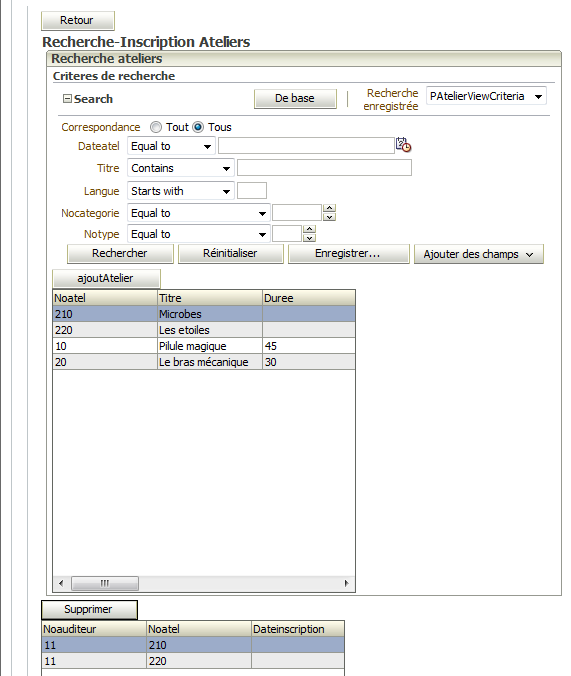
devenirjuge.jsff







**InscriptionAteliers.jsff**



**Bouton <<retour>> :** revient au menu précédent

**Bouton <<ajout Atelier>>** : Ajoute l’atelier sélectionné à la 2ième table

**Bouton <<Supprimer>> :** Supprime l’inscription sélectionnée à la 2ième table

**Remettre : Dans un dossier à votre nom (ou aux 2 noms) :**

* **Votre application**