JavaServer Faces & Toplink : une application simple

Écrit par Chris SCHALK, Oracle Corporation
Janvier 2005

Introduction

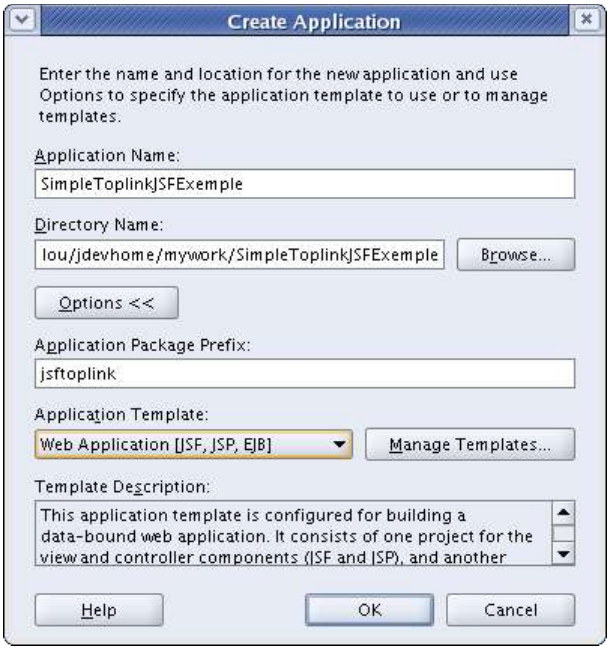
JDeveloper 10.1.3 Preview permet de développer simplement et efficacement des applications JavaServer Faces (JSF) grâce à un nouvel éditeur graphique. Il permet également les développements Toplink, ce qui rend simple la création d'applications JSF dont les composants d'accès aux données sont basés sur Toplink.

L'exemple qui suit présente comment développer une application JSF qui affiche un tableau de données géré par un mid-tiers Toplink

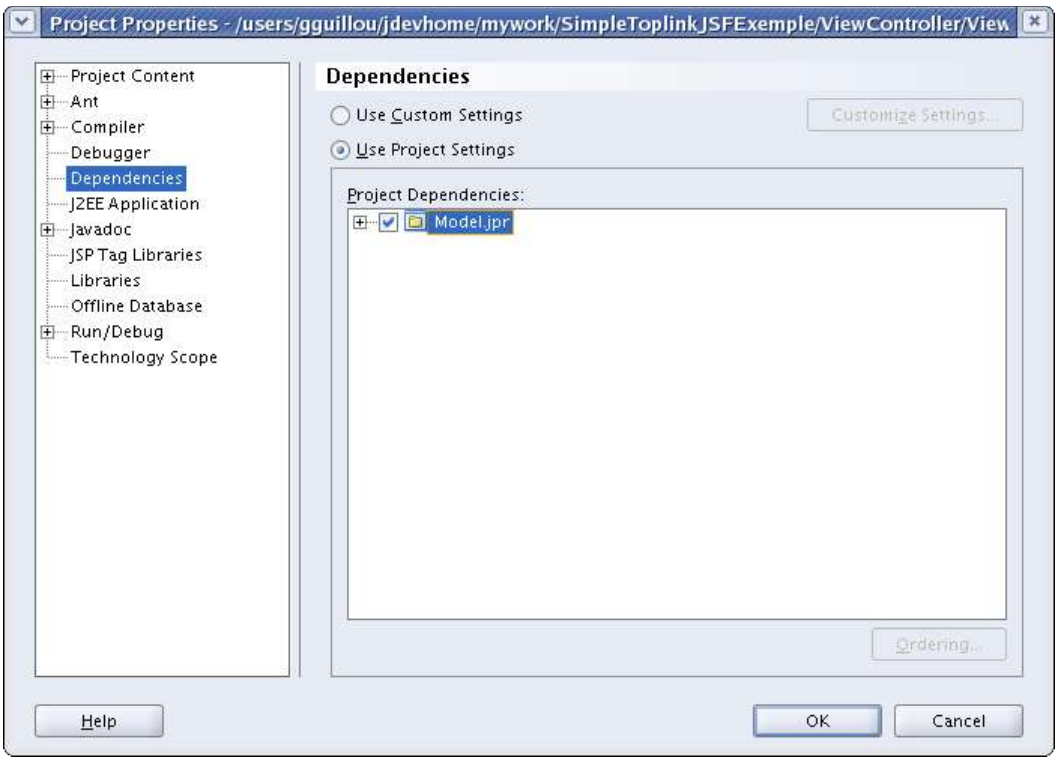
Démarrer

Pour démarrer, téléchargez JDeveloper 10.1.3 Preview sur OTN et installez-le sur votre machine. Vous devez également configurer une connexion à une base de données. Ce tutoriel utilise le schéma exemple SCOTT d'une base de données Oracle mais n'importe quelle connexion conviendra.

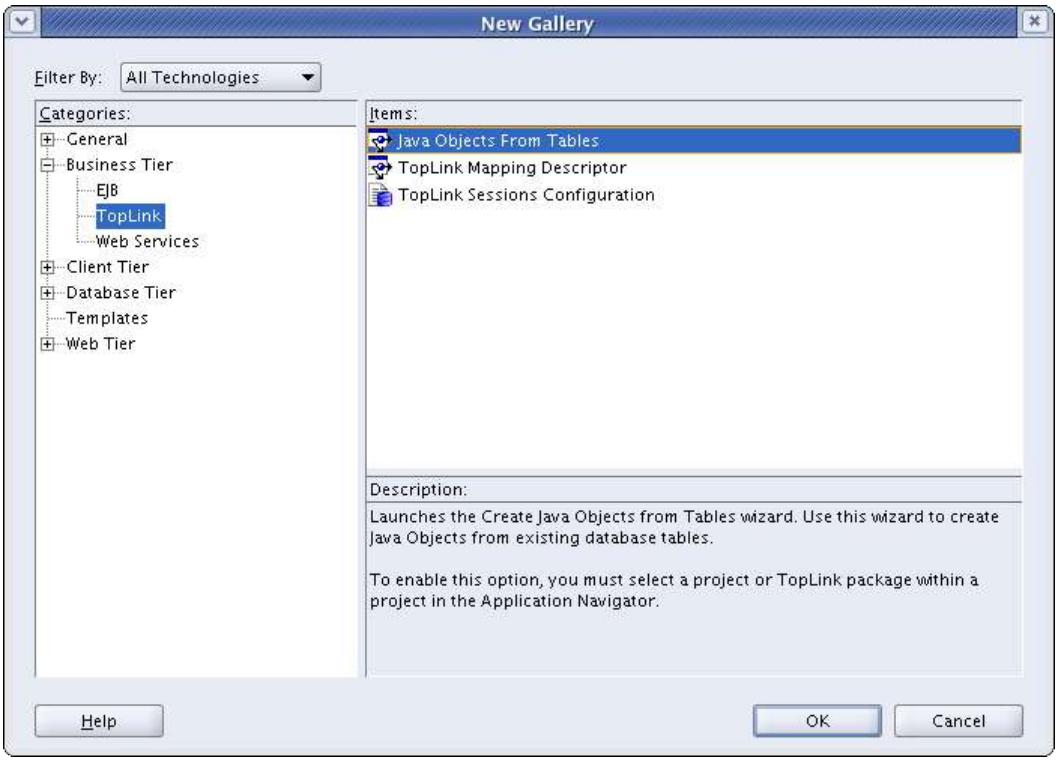
1. Pour commencer, nous allons utiliser l'assistant de création d'applications qui générera le projet « Model » de gestion de la persistance et le projet « ViewController » pour nos composants d'affichage et la logique d'interactions. Sélectionnez le menu « File -> New -> General -> Application »



2. Appuyez sur OK pour générer le workspace
3. Après avoir généré nos 2 projets ViewController et Model, nous devons définir la dépendance du premier par rapport au second. Pour cela, double-cliquons sur le projet ViewController, sélectionnons « Dependencies » et activons la case à cocher devant Model. Sélectionnons le bouton OK pour continuer...



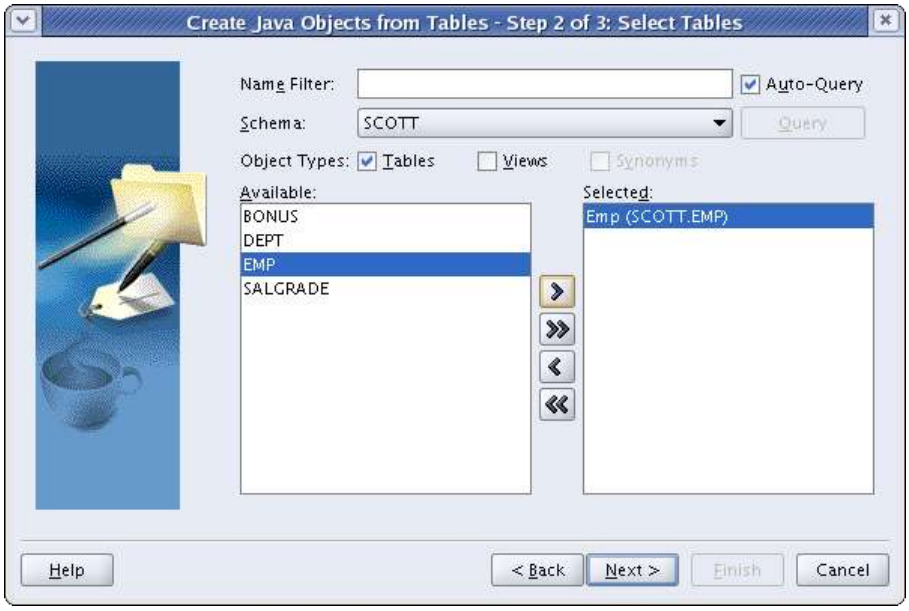
4. Nous pouvons créer le code du projet « Model » avec Toplink. Utilisons un assistant à partir du menu « File -> New -> Business Tiers -> Java Objects from Tables »



5. Une fois l'assistant démarré, nous pouvons cliquer sur « Next » pour faire

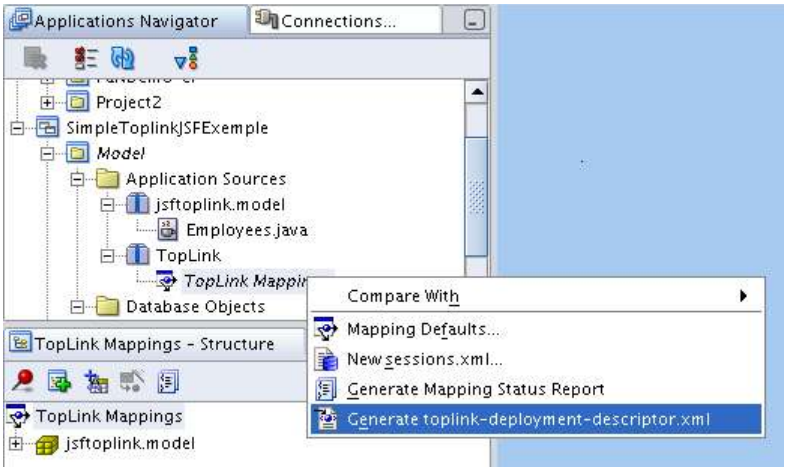
apparaître la fenêtre de sélection de la connexion à la base de données. Choisissons parmi les connexions de bases de données prédéfinies, celle qui convient.

6. Dans l'étape 2/3 de l'assistant, nous sélectionnons la table EMP du schéma SCOTT.

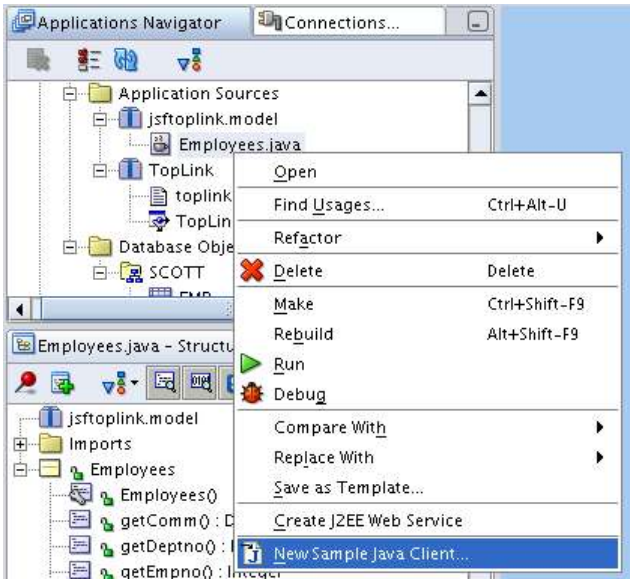


7. Cliquons « Next » puis « Finish » pour générer les classes Java Toplink de la table EMP.

8. Maintenant que nous avons généré notre classe Toplink et un mapping associé, nous devons générer le descripteur de déploiement Toplink. Sélectionnons « Toplink Mappings » puis « Generate toplink-deployment-descriptor.xml » depuis le bouton droit de la souris.



9. Nous pouvons maintenant créer un bean java qui accède à notre projet Toplink pour tester notre exemple et pour utiliser dans notre application JSF à venir. Sélectionnez la classe générée (Employees.java) et à l'aide du bouton droit de la souris « New sample java client »



10. Testons l'application cliente générée en l'exécutant (Menu Run)

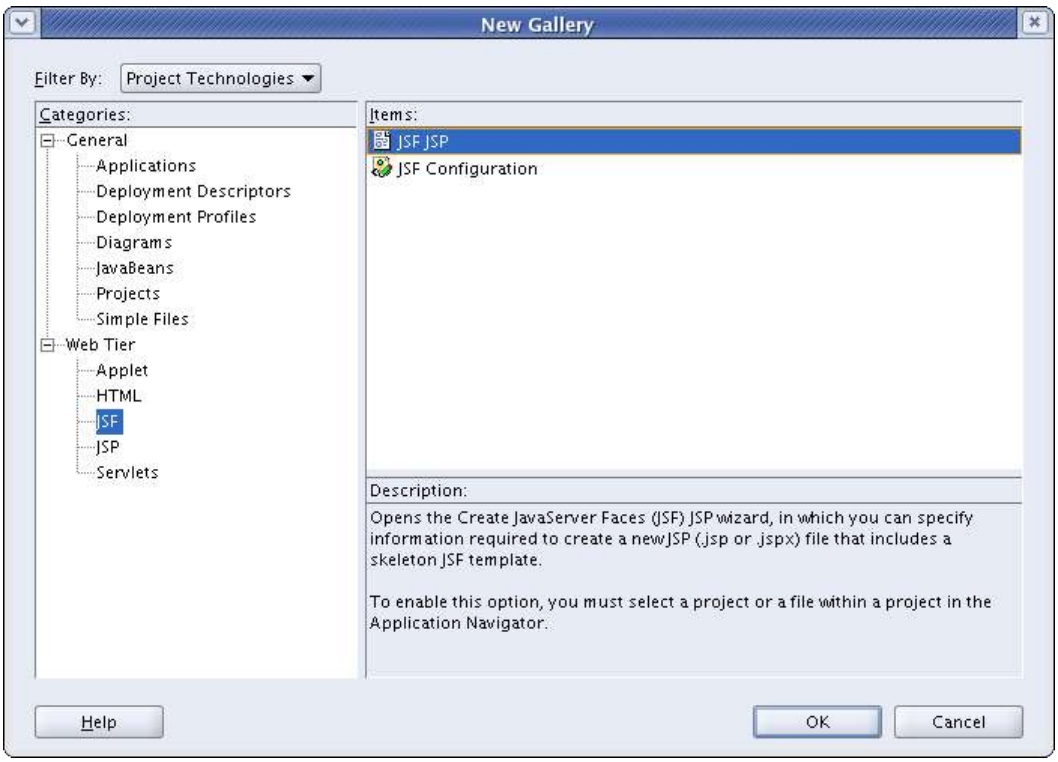
11. Nous allons créer une méthode getEmployees() dans notre classe cliente. Pour cela, il faut :
    - Copier la méthode main() et son contenu et l'utiliser pour générer la méthode getEmployees() en la modifiant. La méthode doit retourner un type java.Util.Vector (nous utiliserons cette méthode dans l'application JavaServer Faces)
    - Au lieu d'afficher le Vector, comme dans la méthode main(), la méthode getEmployees() retournera la liste des employés
    - le code ajouté ressemble à celui qui suit :

```
import java.util.Vector;
 .
 .
public Vector getEmployees()
{
EmployeesClient employeesClient = new EmployeesClient();
oracle.toplink.sessions.Project project = XMLProjectReader.read
            ("path-to-your-toplink-deployment-descriptor.xml");
DatabaseSession session = project.createDatabaseSession();
session.login();
Vector objects = session.readAllObjects
             (jsftoplink.model.Employees.class);
return objects;
}
```

12. Nous pouvons désormais créer la vue de notre application qui consistera en une page JavaServer Faces JSP qui utilise le composant JSF Datatable UI pour présenter les employés.

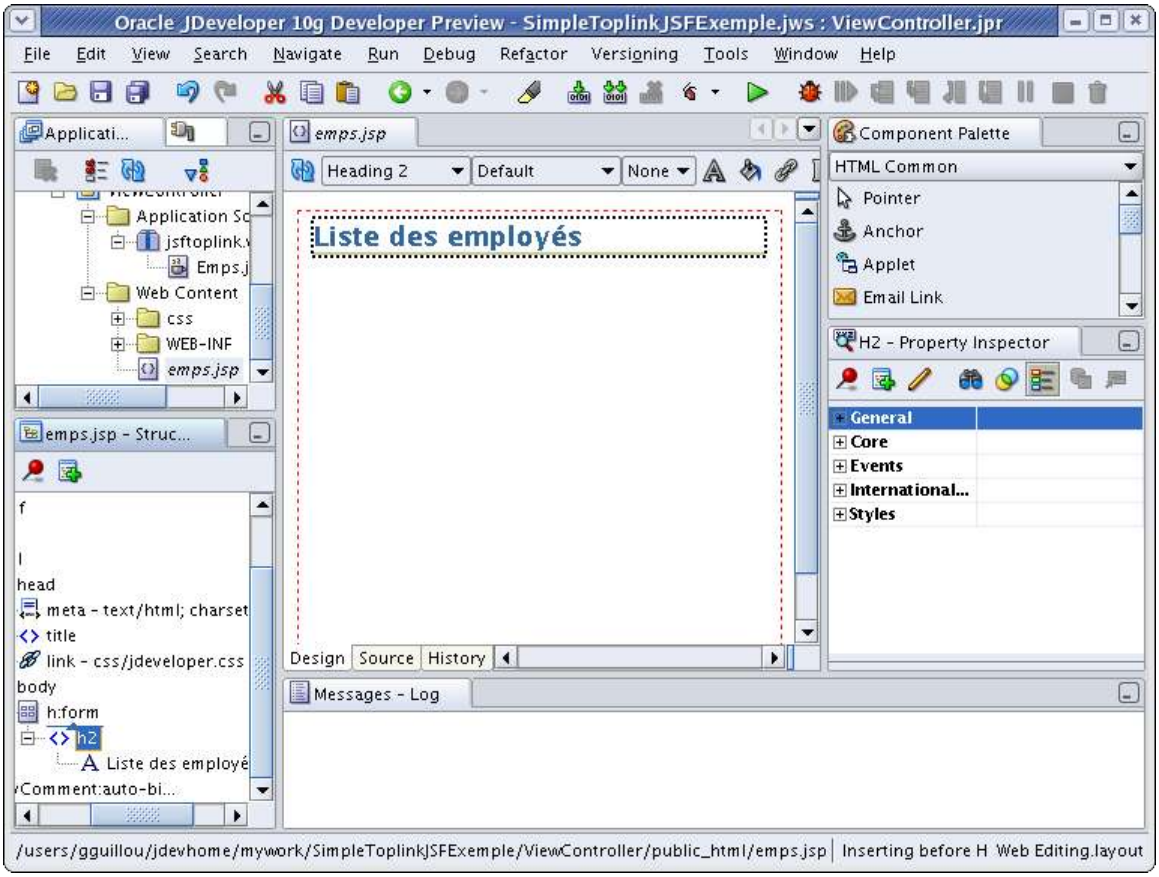
La « View »

1. Dans le projet ViewController, nous allons construire une page JSP qui utilise JavaServer Faces. Pour cela, sélectionnons le menu « File -> New -> Web Tier -> JSF -> JSF JSP »

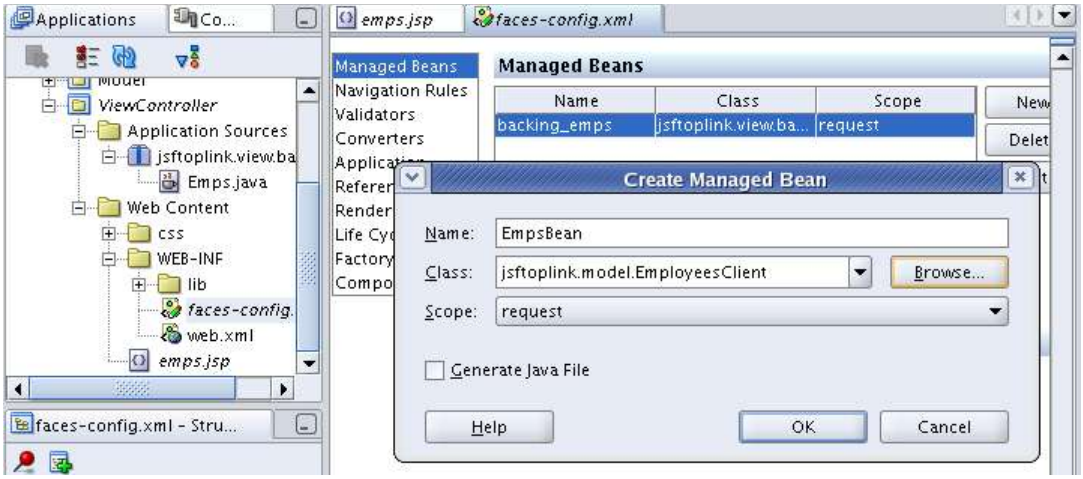


2. Utilisons l'assistant pour nommer la page emps.jsp. Le type de la page est « Page JSP » et l'option de gestion des exceptions pour la page est « Do Not Use an Error Page to Handle Uncaught Exceptions in This File ». Cliquons sur « Finish » pour générer directement la page sans passer par les étapes suivantes.

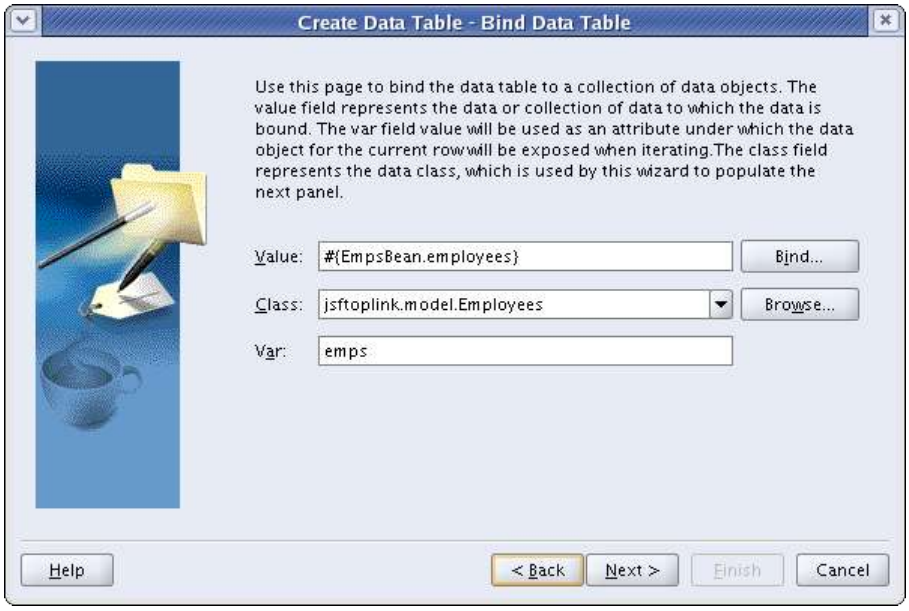
3. Une fois la page générée, nous pouvons simplement ajouter une balise <H2> pour générer une bannière comme ci-dessous. Nous pouvons également glisser/déposer une feuille de style (Cascading Style Sheet) comme celle de JDeveloper depuis la palette des feuilles de styles.



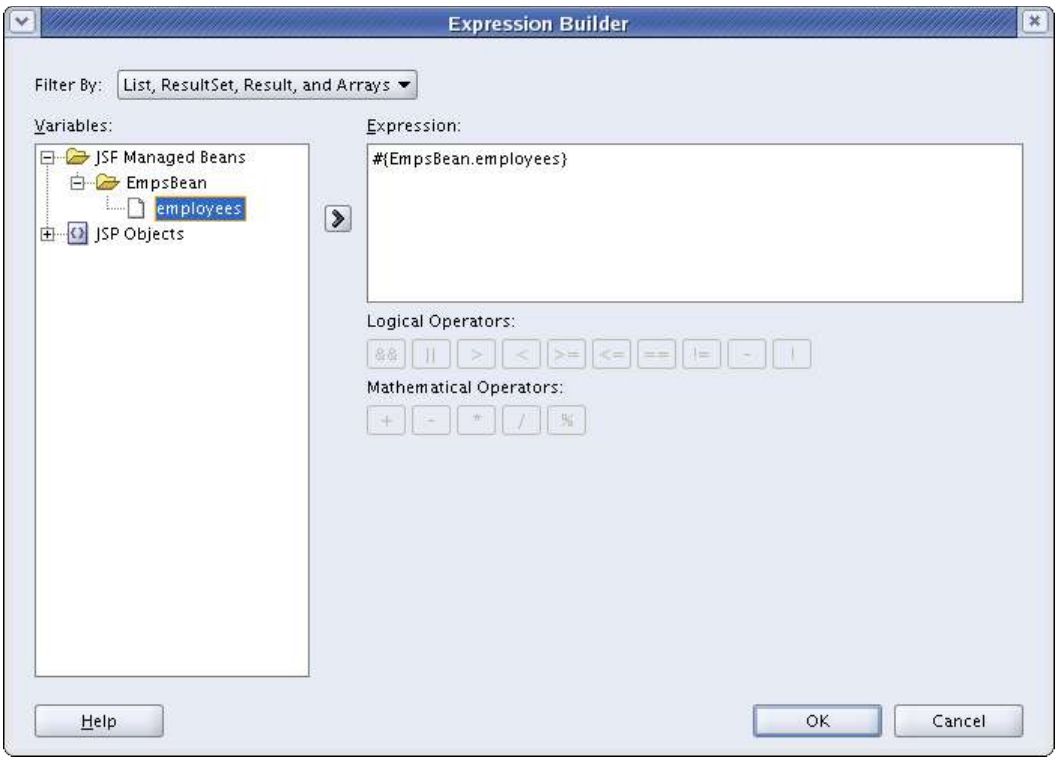
4. Avant de continuer, nous devons ajouter la classe client toplink que nous avons créée dans le projet « Model » comme un Managed Bean dans le fichier faces-config.xml. Double-cliquons sur ce fichier (Web Content -> WEB-INF -> faces-config.xml) pour éditer son contenu
5. L'éditeur par défaut du fichier faces-config.xml est le modeleur des flux des pages JSF (Page Flow Modeler). Nous ne nous servons pas pour cet exemple des transitions entre les pages. Sélectionnons l'onglet « Overview » en bas de la page pour invoquer la console d'édition de faces-config.xml
6. Ajoutons un nouveau « Managed Bean » en cliquant sur la zone « Managed Bean », puis en sélectionnant le bouton « New ...». Spécifions ce qui suit et cliquons sur OK
   - Name : EmpsBean
   - Class : jsftoplink.model.EmployeesClient (naviguer dans l'arborescence de votre projet pour sélectionner cette classe java)
   - Scope : Request



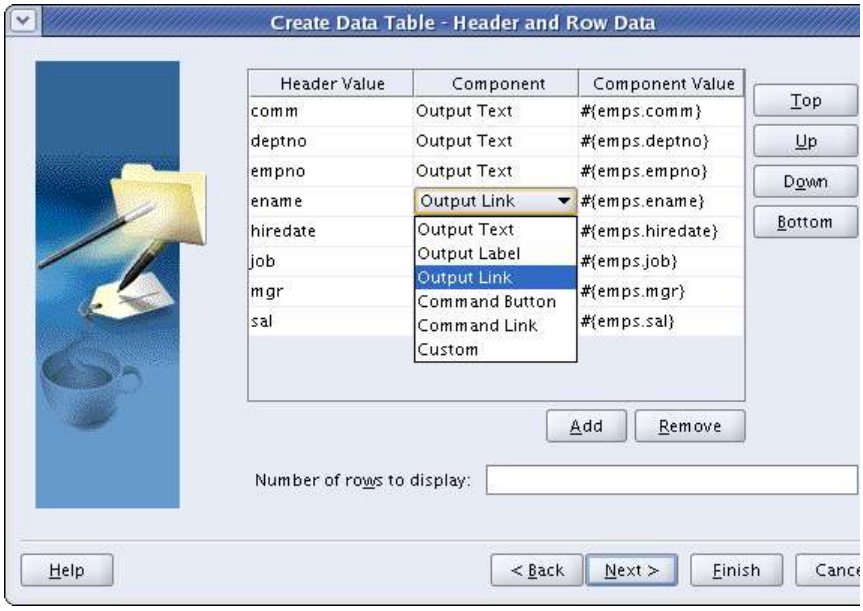
7. Ca y est presque ! Il nous reste à glisser/déposer le composant DataTable UI sur la page JSP et mettre les données dans ce composant (DataBinding). En plus Jdeveloper 10.1.3 a un assistant qui permet d'effectuer ces 2 étapes.
8. Trouvons le composant JSF HTML « Data Table » dans la page JSF HTML de la palette de composants et faisons le glisser/déposer du composant sur la page JSP. Cette action démarre l'assistant.
9. Il faut cliquer sur « Next » pour aller sur la première page de l'assistant. Laissons le bouton radio sur la position : « Bind the Data Table Now » et cliquons sur « Next »
10. Saisissons les valeurs des champs comme suit:
    - Value : #{EmpsBean.employees}
    - Class : jsftoplink.model.Employees
    - Var : emps



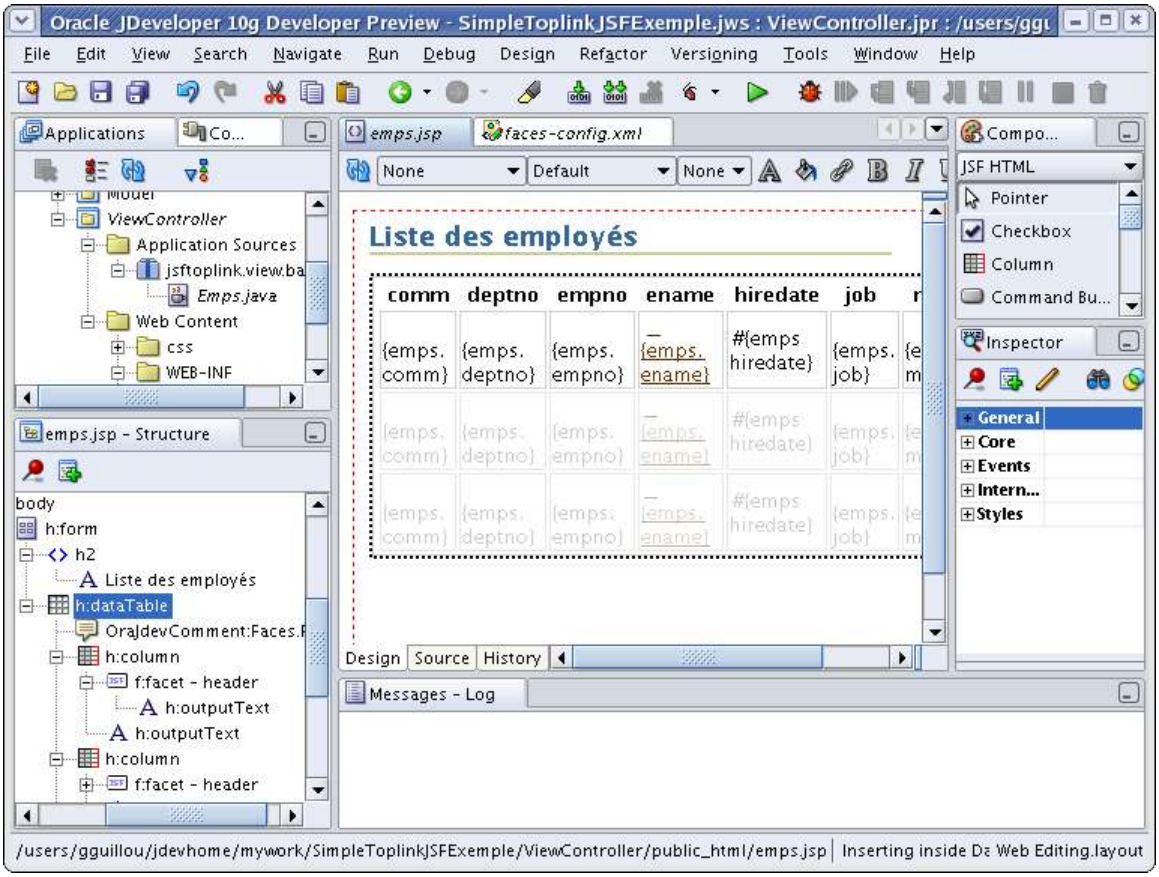
Note : pour le champ « valeur », l'expression builder permet de générer la syntaxe Expression Language (EL) pour accéder à la bonne méthode du Managed Bean.



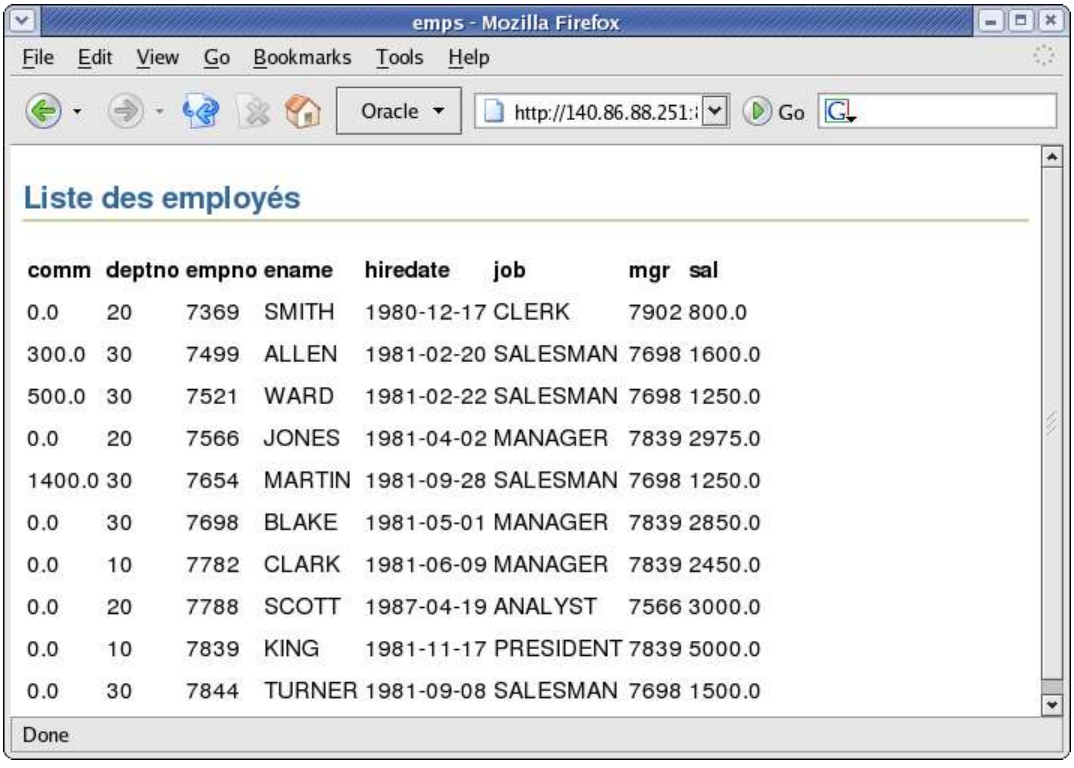
11. En cliquant sur « Next », la page de configuration des colonnes de la table apparaît. Utiliser cette page pour personnaliser l'ordre, les libellés et le type d'affichage de la DataTable.



12. Le bouton « Finish » crée le code DataTable dans la page JSP. Vous verrez apparaître une DataTable avec plusieurs lignes et colonnes.



13. Maintenant exécutons la page ! Avec le bouton droit de la souris, il faut sélectionner emps.jsp et cliquer sur « Run »



Synthèse

Cet exemple simple sert de point de départ pour commencer à créer des applications JavaServer Faces qui utilisent une couche de persistance sur un middle-tier. Cet exemple utilise Toplink, mais n'importe quelle autre technologie peut être utilisée. Il montre aussi l'intérêt des nouvelles fonctions de développement graphique d'Oracle JDeveloper 10.1.3, parmi lesquelles l'éditeur de fichiers de configuration faces-config.xml : « Overview », la boite de dialogue Expression Language (EL) Binder et l'assistant de création de DataTable. Parmi les nouveautés à venir sur la version de production de JDeveloper 10.1.3, plusieurs assistants permettront de créer des UIComponents complexes. N'hésitez pas à faire savoir ce que vous en pensez sur les forums à commencer par OTN !