TUTORIAL

Démarrer un projet JPA 2.1 sous Eclipse 4.4

Ce tutorial a pour objet de présenter la configuration de base d'un projet JPA dans Eclipse, et de créer une première entité JPA. Il utilise une connexion MySQL sur une instance 5.6 et EclipseLink comme implémentation JPA 2.

Sommaire

[TP 1- Installation du serveur MySQL 3](#_Toc422473527)

[TP 2- Création d'une connexion MySQL 4](#_Toc422473528)

[TP 3- Création d'un projet JPA 9](#_Toc422473529)

[TP 4- Création d'une entité JPA 15](#_Toc422473530)

# Installation du serveur MySQL

Si ce n’est pas déjà fait :

Installation d’EasyPHP

* Installer le pack EasyPHP fourni : EasyPHP-DevServer-14.1VC9-install
* Il contient un serveur Apache-PHP, l’application PhpMyAdmin et une instance MySQL 5.6
* Tester l’installation avec l’URL : <http://127.0.0.1/home/>

Création/importation de la base « banque »

* Si votre base « banque » n’existe pas encore, dans PhpMyAdmin, créer une nouvelle base « banque » vide

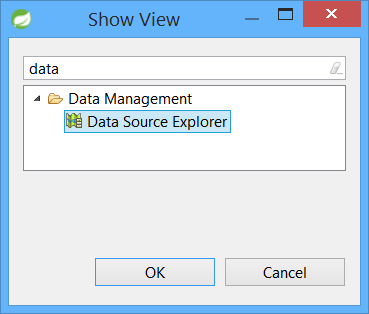
# Création d'une connexion MySQL

Un projet JPA a besoin de se connecter à une base de données. Ci-dessous la façon de créer une connexion à une base MySQL sous Eclipse 4.4

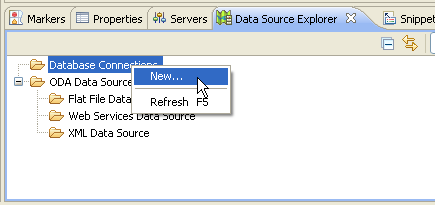
Déposer le driver JDBC pour MySQL fourni « **mysql-connector-java-5.1.35-bin.jar** » dans un dossier de votre choix (sans espaces) Ex : d:\web\lib :



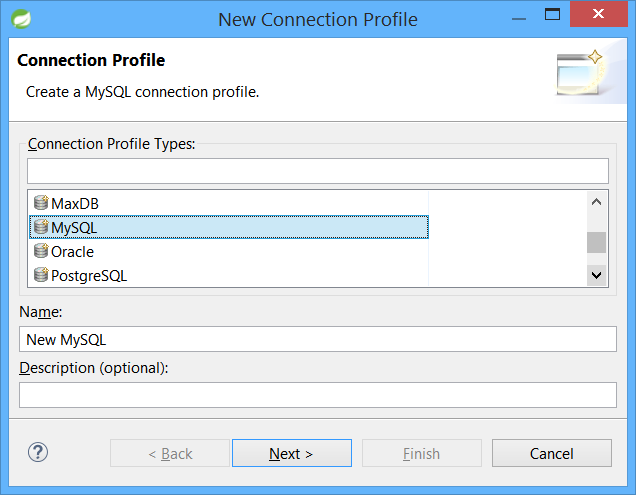
Ouvrir la vue « Data Source Explorer » : Menu Window -> Show View -> Others…



Sélectionner la rubrique New... du menu contextuel de la vue Data Source Explorer.

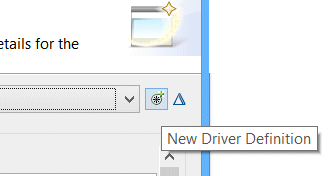


On sélectionne le type de base de données : MySQL dans le panneau qui s'ouvre.

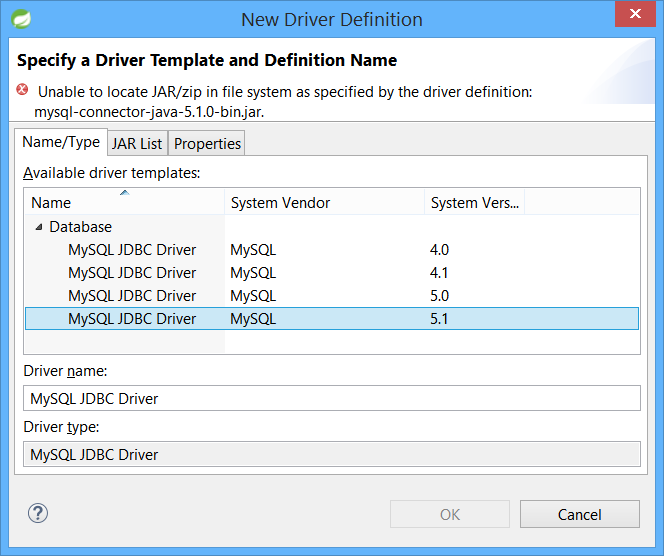


**Création d’un pilote MySQL**

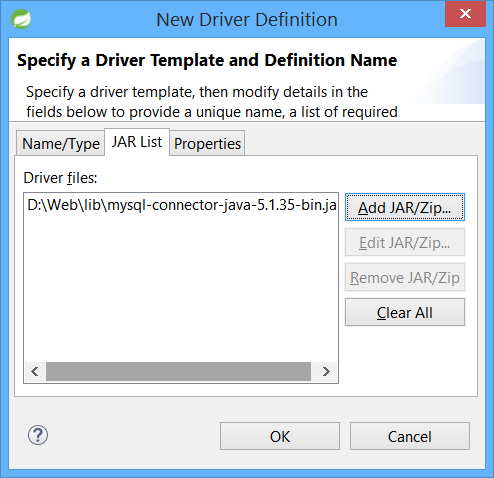
Ecran suivant : il faut choisir et/ou définir le pilote à utiliser. Si aucun driver MySQL n’apparait, le créer avec le bouton :



Puis Choisir le type de base (MySQL 5.1) :



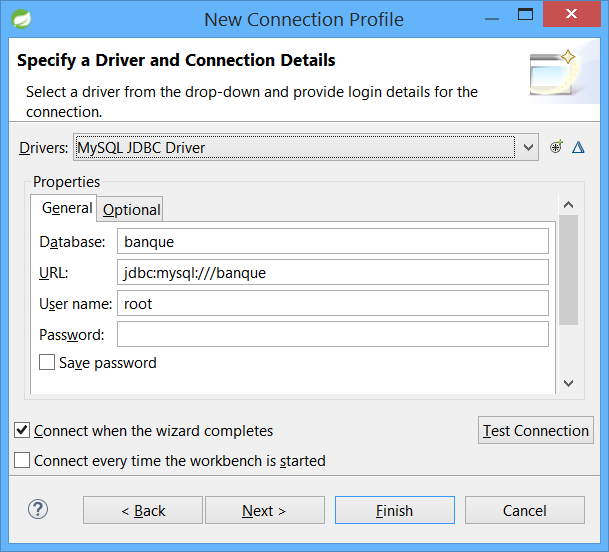
Onglet « JAR List » : sélectionner le fichier jar précédemment installé via le bouton « Add… » et valider avec OK



**URL JDBC de connexion**

Enfin, spécifier l’url de connexion JDBC, le nom de la base (« banque ») et login/mot de passe d’accès.

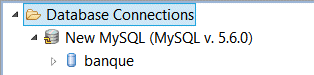
Vous pouvez ensuite tester que tout est OK avec le bouton Test « Connection »



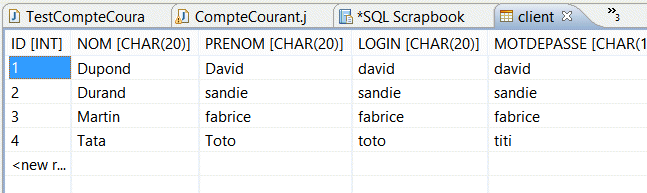
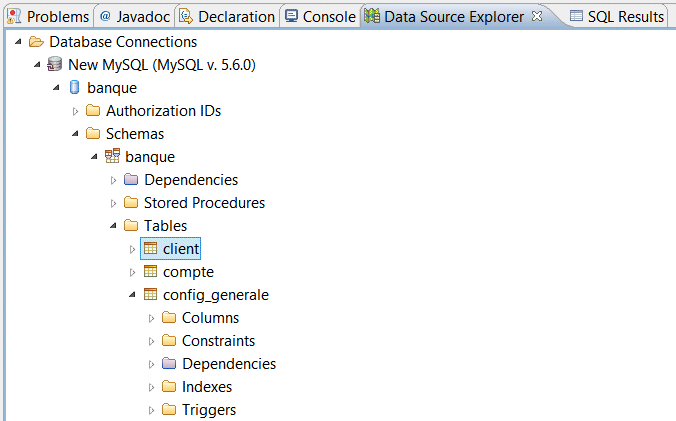
Le dernier écran est juste un rappel des paramètres entrés.

**Utiliser le Datasource Explorer :**

Notre nouvelle connexion apparaît enfin dans la vue Data Source Explorer, comme n'importe quelle autre connexion.



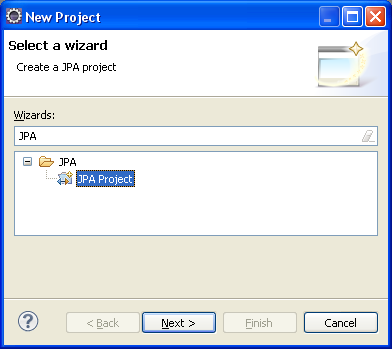
On peut désormais naviguer dans le schéma SQL « banque », effectuer des requêtes et des mises à jour de données. (clic droit sur une table : « Data -> Edit »)



Cette source de données est prête à être utilisée dans le projet JPA que nous allons créer maintenant.

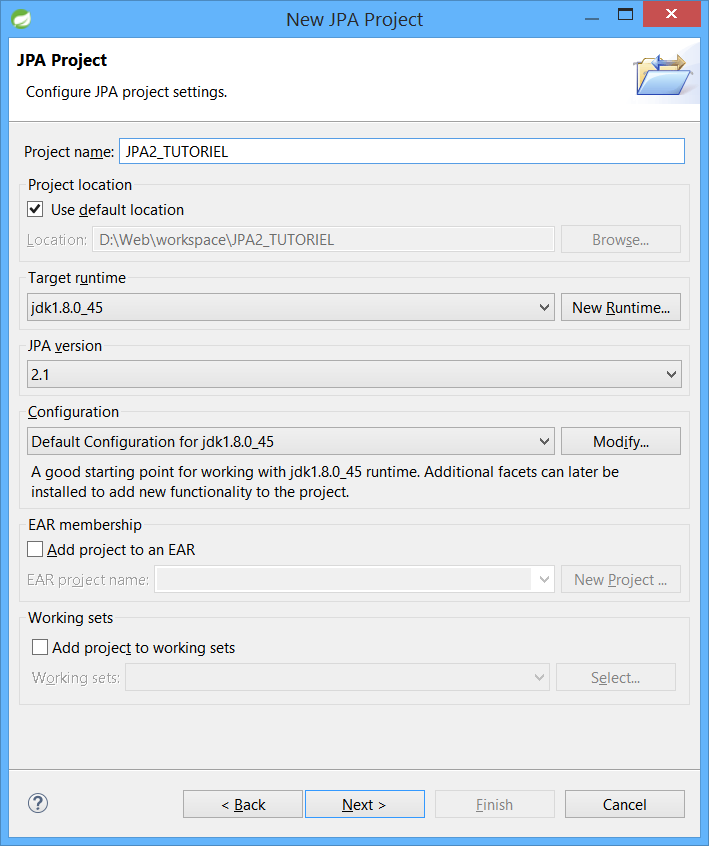
# Création d'un projet JPA

La création d'un projet JPA commence de la même manière que n'importe quel autre projet Eclipse. Le panneau New Project permet de sélectionner JPA Project.



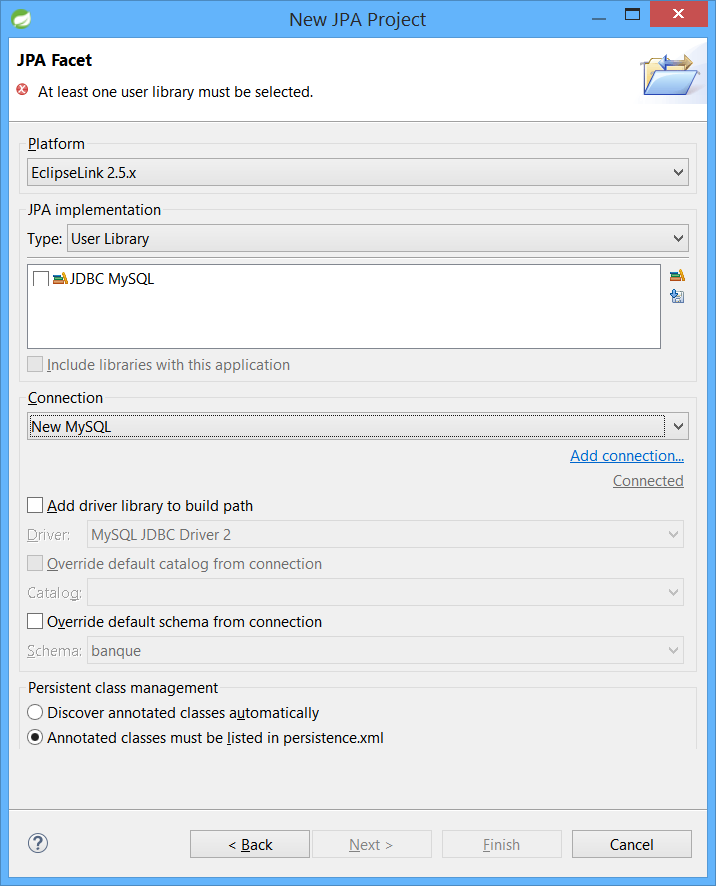
S'ouvre alors un deuxième écran qui permet de préciser les choses.

* Le nom de ce projet et la version de Java utilisée.
* La version de l'API JPA utilisée. (JPA 2.1 par défaut)
* Enfin, il est possible d'ajouter ce projet à un EAR (point non couvert ici)



L'étape suivante permet de préciser éventuellement les répertoires source de notre projet.

Puis choisir l'implémentation JPA que notre projet va utiliser. Il s'agira typiquement d'Hibernate ou d'Eclipselink, cette dernière implémentation étant l'implémentation de référence.

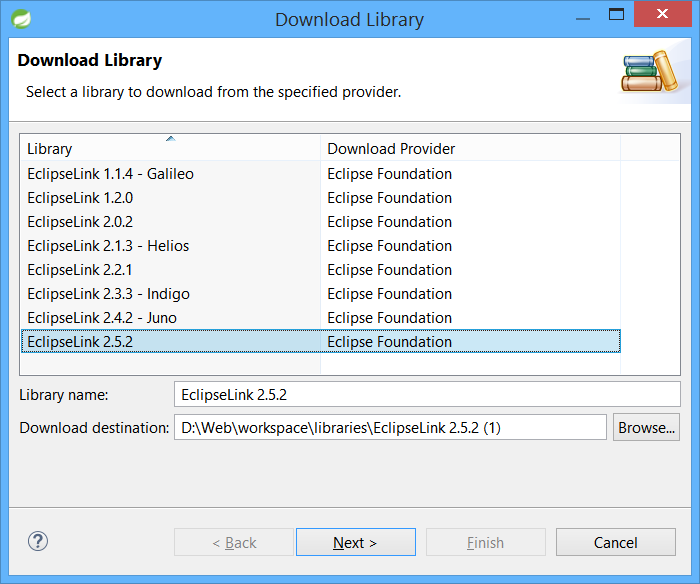


L'implémentation choisie pour notre exemple est EclipseLink 2.5.x, qui n'est pas installée dans les dépendances de notre projet. Pour ce faire, il faut cliquer sur l'icône en forme de disquette à droite dans le cadre JPA implementation.

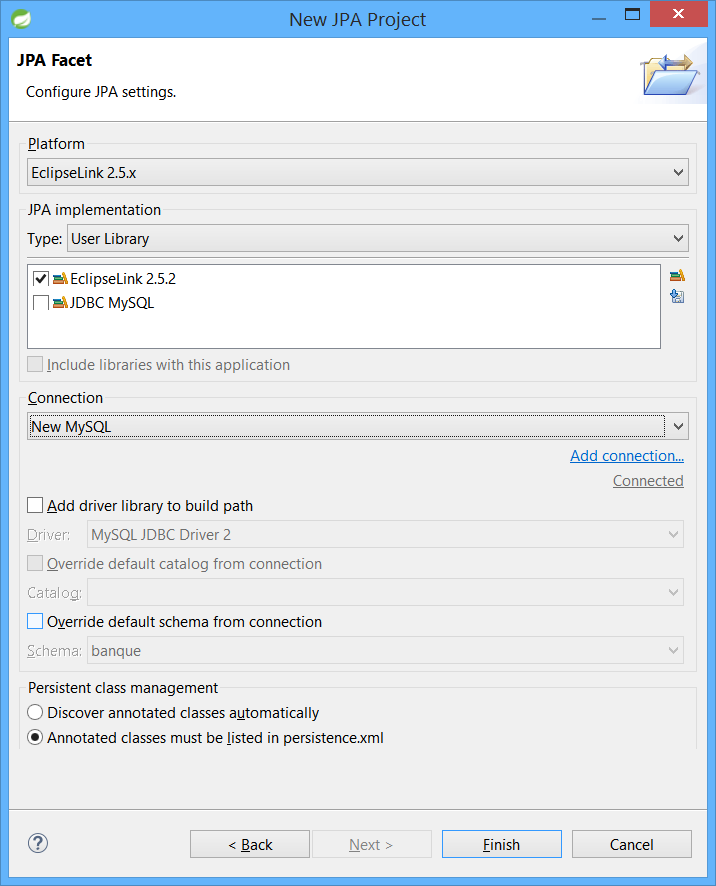
Cet écran nous permet de choisir également la base de données utilisée. On sélectionne ici New MYSQL, connexion que nous venons de créer. Il est alors possible de définir ici un autre jeu de paramètres que celui utilisé par cette connexion, notamment le schéma cible de l'application.

Enfin, on peut choisir de laisser EclipseLink découvrir les entités JPA de notre projet, ou de les lister dans le fichier persistence.xml. C'est cette deuxième option que l'on choisit ici.

Avant de terminer, télécharger la librairie EclipseLink 2.5.2 (Accepter la licence d’utilisation et attendre la fin du téléchargement)

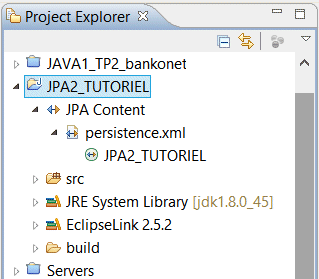


Une fois ce téléchargement terminé, notre projet est prêt à être créé.



Eclipse bascule alors sur la perspective JPA

La structure de notre projet est la suivante, visible dans la vue Project Explorer.

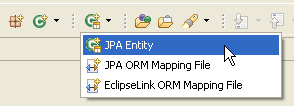


Notre projet est maintenant prêt à accueillir nos premières entités.

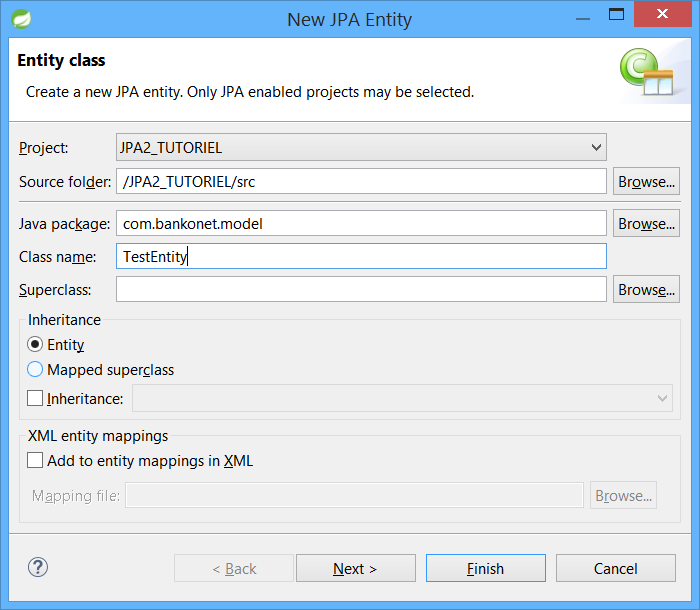
# Création d'une entité JPA

On se propose dans cette partie de créer une première entité JPA. En première approche, une entité JPA est un bean Java classique, annoté, et qui doit posséder une clé primaire. On peut générer de telles classes automatiquement avec les outils fournis par Eclipse.

On commence la création d'une entité JPA en sélectionnant la rubrique JPA Entity sous l'icône New Entity. Attention, cette frise d'icônes n'est visible, par défaut, que sur la perspective JPA.



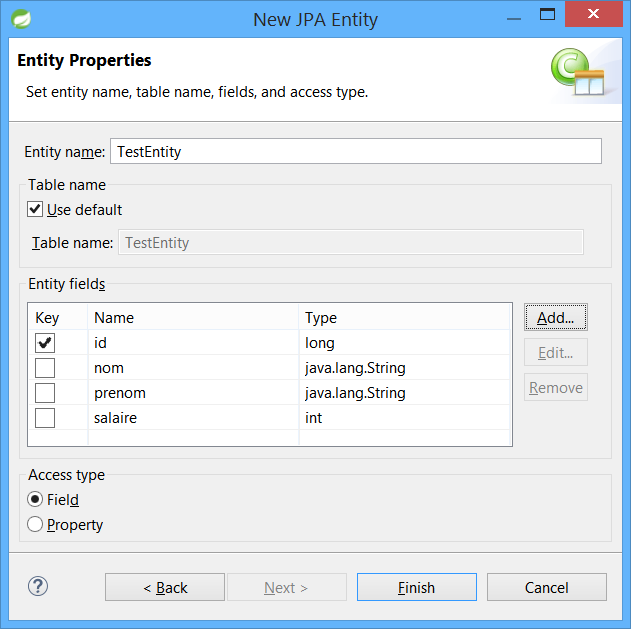
Le premier panneau qui s'ouvre alors nous permet de fixer la classe de notre entité, et éventuellement sa super-classe. Si elle possède une super-classe, alors on peut préciser le type de stratégie pour le mapping de l'héritage.



Le deuxième panneau permet de fixer trois choses.

* Le nom de l'entité JPA que l'on est en train de créer. Par défaut il s'agit du nom complet de la classe. Si ce nom doit être utilisé dans des requêtes JPQL, il est souvent utile d'en choisir un plus simple.
* Le nom de la table associée à cette entité.
* La liste des champs persistants de cette classe, c'est-à-dire ceux qui sont associés à des colonnes de la table dans laquelle sera enregistrée cette entité.

On peut également fixer la façon dont cette classe sera annotée : par ses champs ou ses getters.

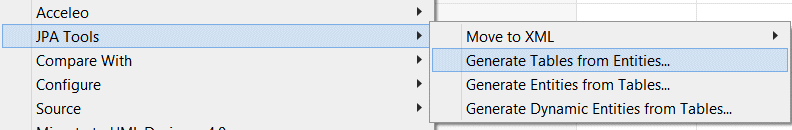


En cliquant sur le bouton Finish, Eclipse crée notre classe, telle qu'elle a été paramétrée.

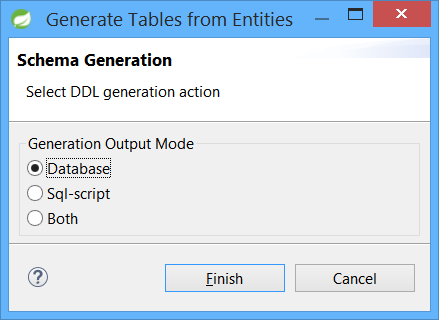
Lancer un build du projet (clic droit sur le projet « -> Build Project ») : une erreur apparait :



La table correspondante n'existe pas encore. Il serait facile de la créer manuellement, mais Eclipse nous propose de le faire pour nous. Il faut pour cela aller chercher la rubrique JPA Tools du menu contextuel de notre projet JPA. Cette rubrique possède une sous-rubrique Generate Tables From Entities..., qui va nous faire le travail.



Là encore Eclipse nous demande si l'on veut créer nos tables en base directement, ou créer un fichier de commandes SQL à exécuter, sans toucher à la base.



Dans notre exemple on choisit de créer directement les tables en base, ce qu'Eclipse va nous faire grâce au couple JPA / EclipseLink

