

Workshop N° 2

Mise en œuvre des entités JPA

1. Création de la base de données :

Commencez par créer une base de données contenant une table Learner.

Le script de création de la base est le suivant :

```
CREATE DATABASE JpaBase;

USE JpaBase;

create table Learner(

    LearnerId integer not null auto_increment,

    Name varchar(30),

    Email varchar(30),

    City varchar(20),

    primary key (LearnerId)

);

insert into Learner(Name,Email,City) values ("Foulen Ben Foulen","test@test.tn","Tunis");

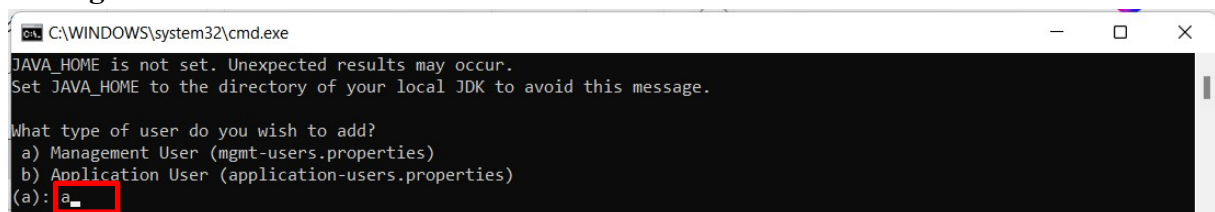
insert into Learner(Name,Email,City) values ("Salah Ben Salah","exemple@test.tn","Tunis");
```

2. Configuration de Wildfly :

1.1. Créer un utilisateur pour administrer Wildfly

Pour créer un compte utilisateur, il faut :

- Démarrer Wildfly
- Accéder au dossier *bin* de votre installation Wildfly
- Exécuter le script *add-user.bat*
- L'invite de commande ci-dessous sera alors lancée vous demandant d'indiquer le type d'utilisateur que vous voulez ajouter. Sélectionnez le premier type (a) relatif au type **Management User**.



```
C:\WINDOWS\system32\cmd.exe
JAVA_HOME is not set. Unexpected results may occur.
Set JAVA_HOME to the directory of your local JDK to avoid this message.

What type of user do you wish to add?
a) Management User (mgmt-users.properties)
b) Application User (application-users.properties)
(a): a
```

- Par la suite, on vous demande d'indiquer le nom d'utilisateur et son mot de passe. Pour notre cas on va utiliser :

* Username: myadmin

* Password: myadmin!2022

Il est possible d'utiliser un username et un password de votre choix

```
C:\WINDOWS\system32\cmd.exe
JAVA_HOME is not set. Unexpected results may occur.
Set JAVA_HOME to the directory of your local JDK to avoid this message.

What type of user do you wish to add?
a) Management User (mgmt-users.properties)
b) Application User (application-users.properties)
(a): a

Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing property files.
Username : myadmin
Password recommendations are listed below. To modify these restrictions edit the add-user.properties configuration file.

- The password should be different from the username
- The password should not be one of the following restricted values {root, admin, administrator}
- The password should contain at least 8 characters, 1 alphabetic character(s), 1 digit(s), 1 non-alphanumeric symbol(s)
Password :
Re-enter Password :
```

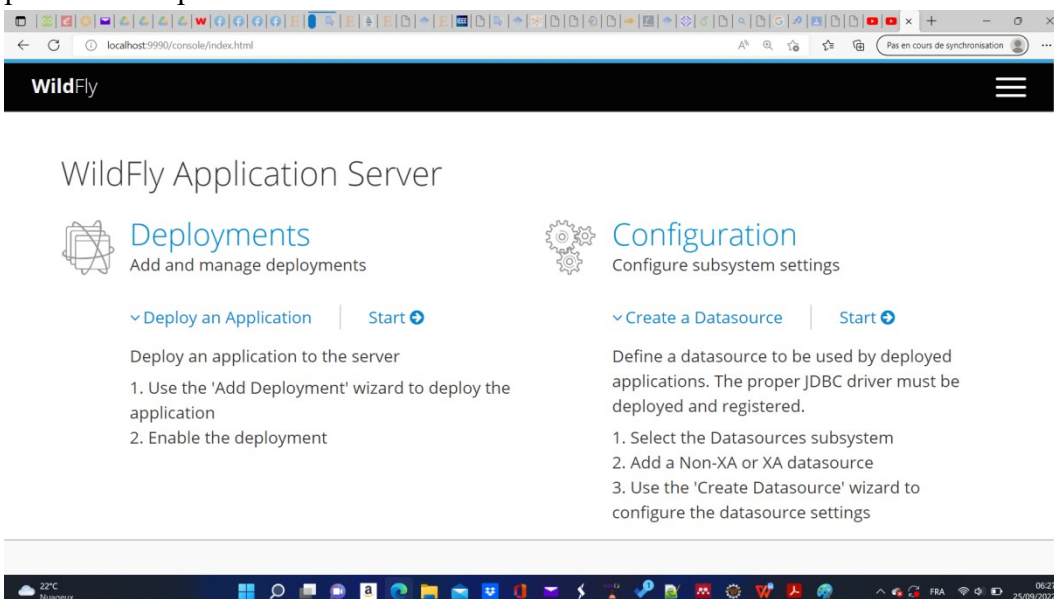
- Par la suite, on vous demande d'indiquer les groupes auxquels cet utilisateur sera associé. Ne rien choisir et cliquer sur entrée. Dans ce cas le nouveau utilisateur sera automatiquement associé au group **ManagementRealm**. Confirmez en saisissant **yes**.

```
What groups do you want this user to belong to? (Please enter a comma separated list, or leave blank for none)[ ]:
About to add user 'myadmin' for realm 'ManagementRealm'
Is this correct yes/no? yes
```

- Par la suite, on vous demande si le nouveau utilisateur sera utilisé pour assurer une connexion entre AS process ce qui n'est pas le cas. Il est possible de répondre par **no**.

```
Is this new user going to be used for one AS process to connect to another AS process?
e.g. for a slave host controller connecting to the master or for a Remoting connection for server to server Jakarta Enterprise Beans calls.
yes/no? no
```

- Pour vérifier la création du nouveau compte utilisateur, accédez à l'interface d'administration <http://127.0.0.1:9990/console/index.html> et saisissez le login et le mot de passe du compte créé. L'interface d'administration ci-dessous sera alors affichée.



1.2. Configuration d'une source de données au niveau de Wildfly

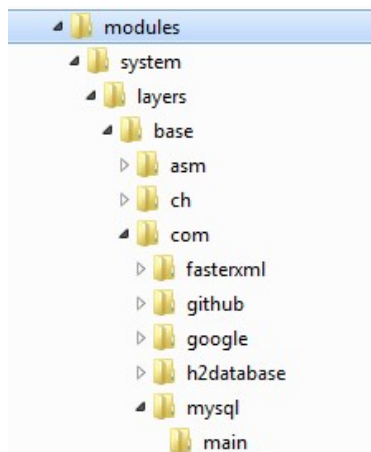
1.2.1. Créer un module dans Wildfly pour la base de données MySQL :

On va commencer par créer un **module de base de données MySQL** dans wildfly afin qu'il soit disponible pour les applications utilisant une source de données MySQL.

Étapes à suivre :

- Accédez à l'installation de WildFly sur votre disque.
- Sous l'installation, vous pouvez voir le répertoire des **modules**. Ouvrez le répertoire et accédez au dossier **base** en suivant le chemin *modules/system/layers/base*
- Ici vous pouvez voir le répertoire **com**. Créez un répertoire **mysql** sous **com**.
- Créez un répertoire **main** sous **mysql**.
- Sous **main**,
 - o Ajoutez le connecteur **mysql-connector-java-8.0.23.jar**
 - o Créez un fichier nommé **module.xml**.

La structure du répertoire devrait ressembler à ceci :

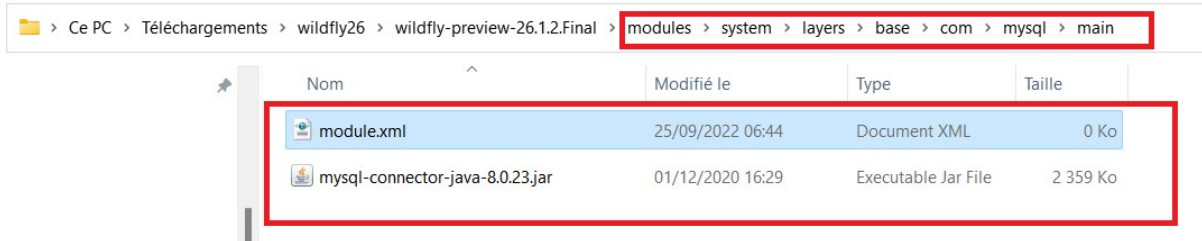


Maintenant, ajoutez le code ci-dessous au fichier **module.xml** pour qu'il soit reconnu comme un module dans WildFly.

```
<module xmlns="urn:jboss:module:1.9" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java-8.0.23.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

- **name = "com.mysql"** : est l'identifiant qui sera utilisé pour identifier ce module
- **path = "mysql-connector-java-8.0.23.jar"** est l'emplacement du fichier jar du connecteur mysql relatif au répertoire *main*.

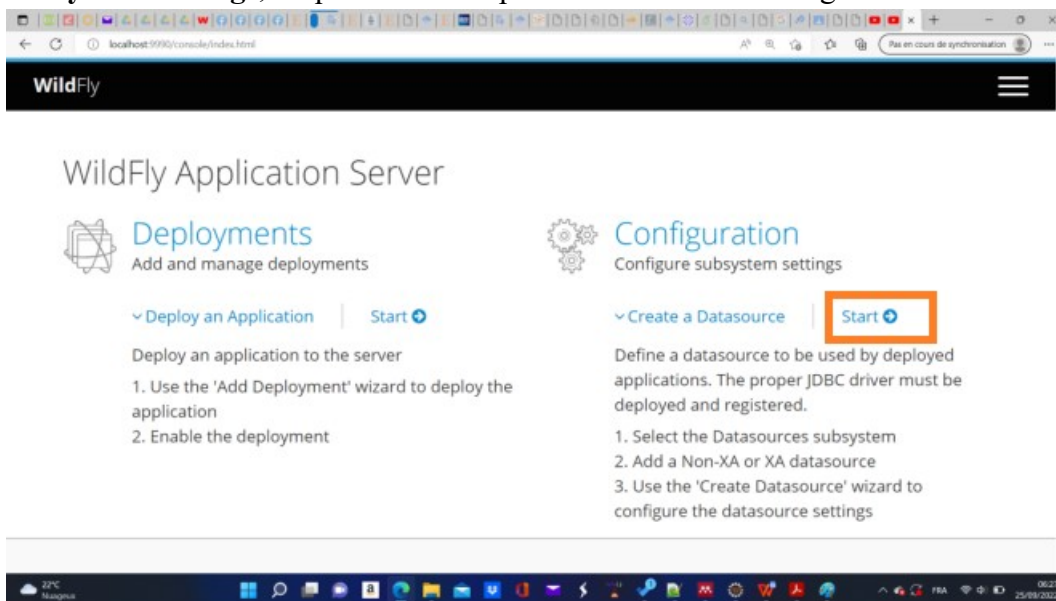
Jusqu'à présent, nous avons créé un module de base de données dans wildfly.



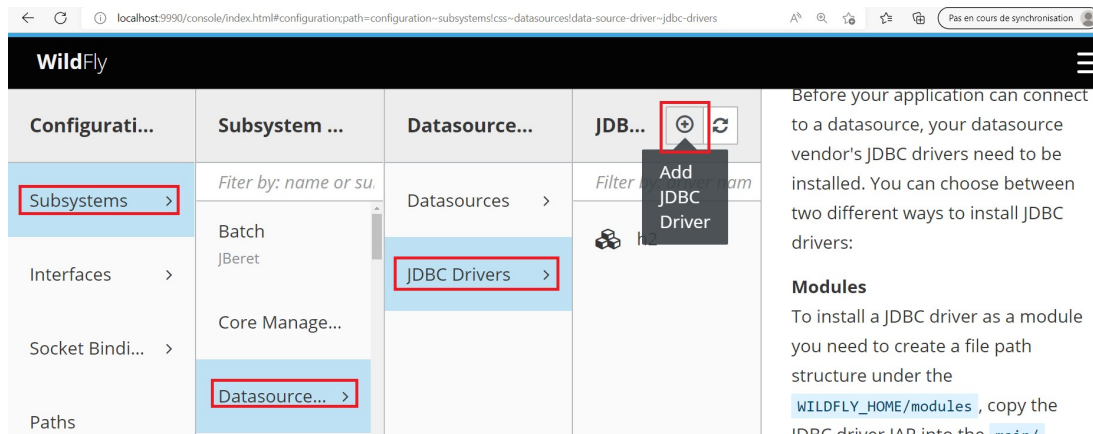
Redémarrez le serveur Wildfly afin que ces changements seront pris en considération.

1.2.2. Installer le connecteur JDBC de MySQL sous Wildfly

- Accédez à l'interface d'administration de Wildfly et connectez-vous avec le nouveau compte utilisateur créé. Dans l'interface qui s'affiche et au niveau de la rubrique **Configure subsystem settings**, cliquez sur **Start** pour commencer la configuration.



- Ensuite, accédez à **Subsystems>Datasources & Drivers>JDBC Drivers > Add**.



- Dans l'interface d'ajout du **JDBC Driver**, indiquez les informations suivantes :

- * Driver Name : **mysqlDriver**
- * Driver Module Name : **com.mysql**
- * Driver Class Name :
com.mysql.cj.jdbc.Driver
- * Driver XA Datasource Class Name :
com.mysql.cj.jdbc.MysqlXADataSource

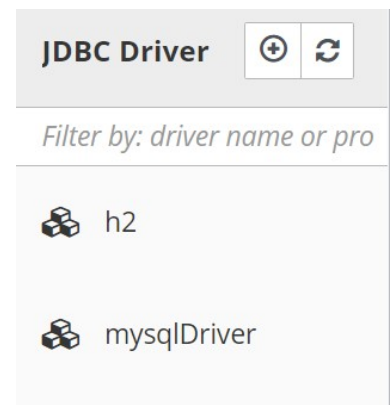
The 'Add JDBC Driver' dialog box is shown with the following fields filled:

- Driver Name *: mysqlDriver
- Driver Module Name *: com.mysql
- Module Slot: (empty)
- Driver Class Name: com.mysql.cj.jdbc.Driver
- Driver Datasource Class Name: (empty)
- Driver XA Datasource Class Name: com.mysql.cj.jdbc.MysqlXADataSource

Required fields are marked with *

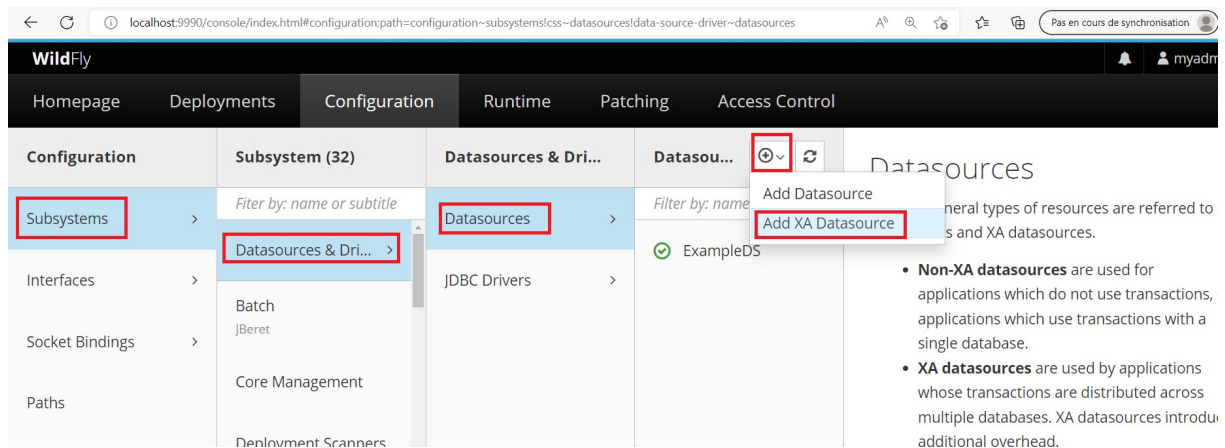
Buttons: Cancel, Add

En terminant l'assistant, le nouveau JDBC Driver mysqlDriver sera ajouté à la liste des drivers configurés sous Wildfly

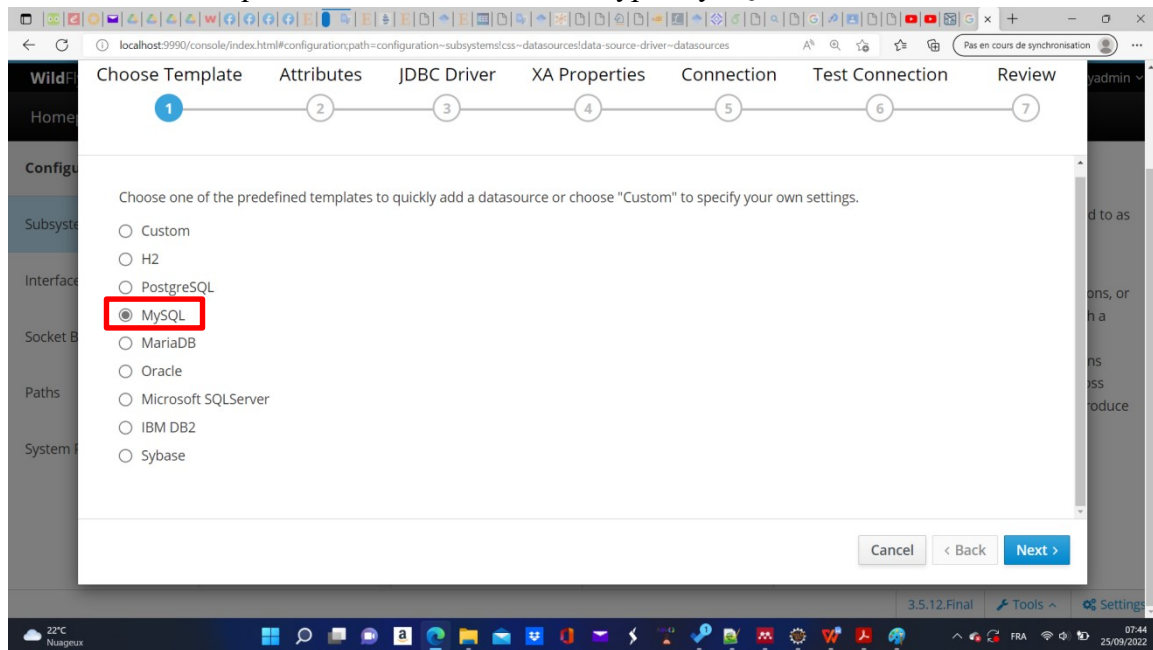


1.2.3. Configuration de la source de données MySQL sous Wildfly

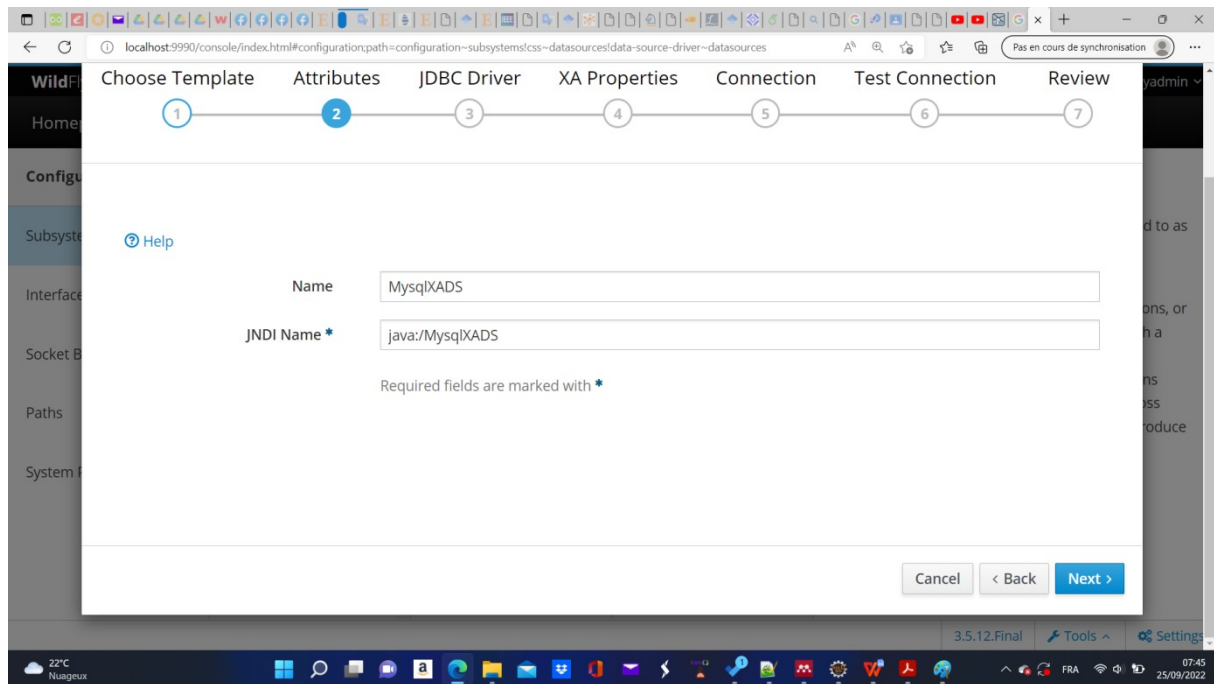
- Accédez à **Subsystems > Datasources & Drivers > Datasources > Add XA Datasource**.



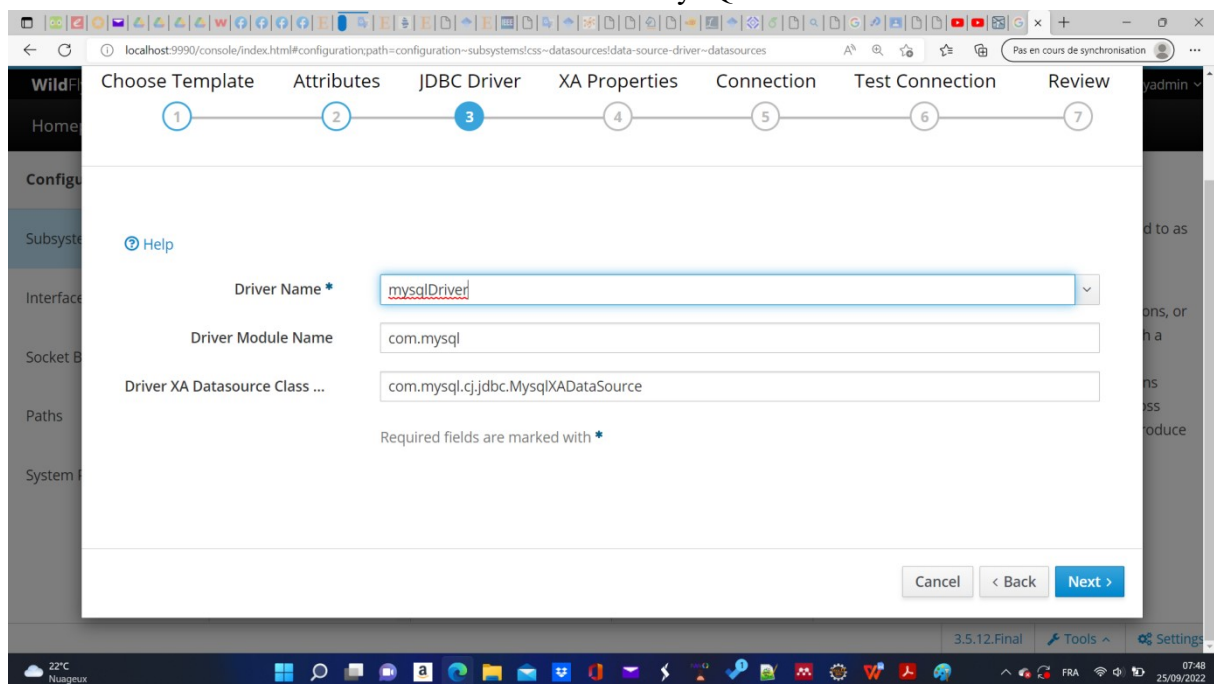
- Dans l'interface qui s'affiche, sélectionnez le type MySQL



- Ensuite indiquez le nom de la source de données (MysqlXADS) et son **JNDI Name** (java:/MysqlXADS). Il est possible d'utiliser d'autres noms selon votre choix. Le JNDI Name sera utilisé par votre Web Service pour se connecter à la source de données.

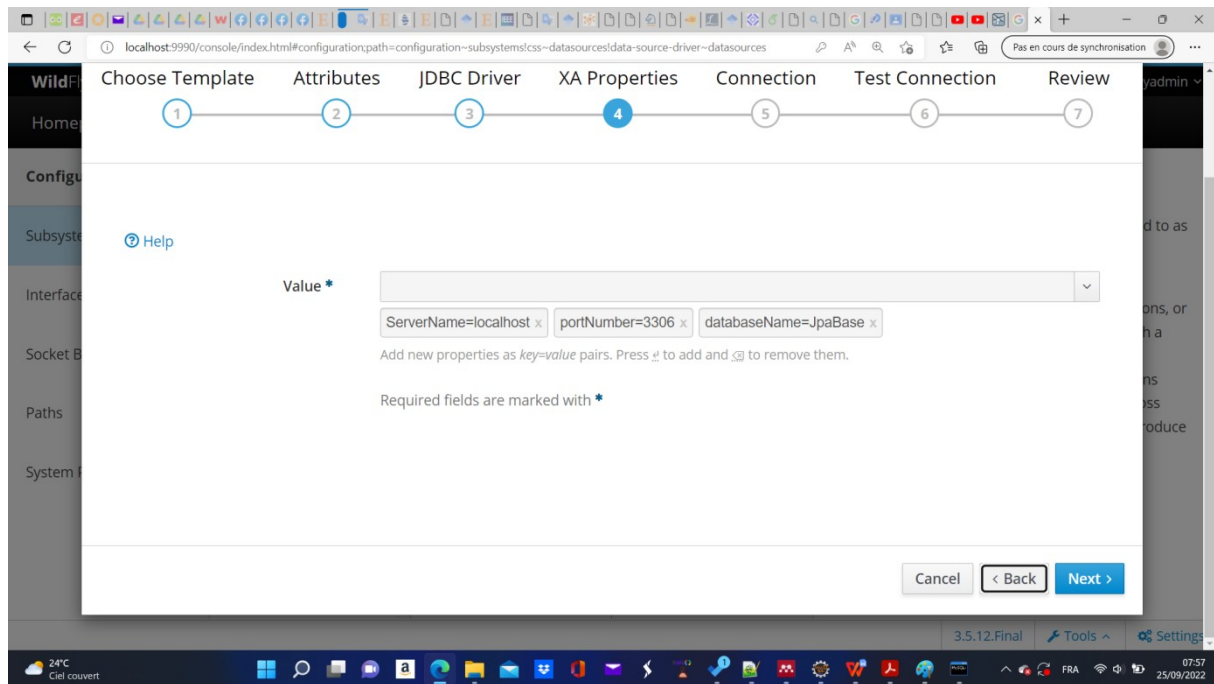


- Dans l'interface suivante sélectionnez **mysqlDriver** en tant que nom du connecteur JDBC à utiliser pour se connecter à cette source de données (le nom relatif au connecteur JDBC qui a été créé lors de l'étape précédente) et **com.mysql** en tant que nom du module relatif à ce driver. Indiquez aussi la classe **com.mysql.cj.jdbc.MysqlXADataSource** en tant que classe de source de données associée à notre connecteur MySQL.

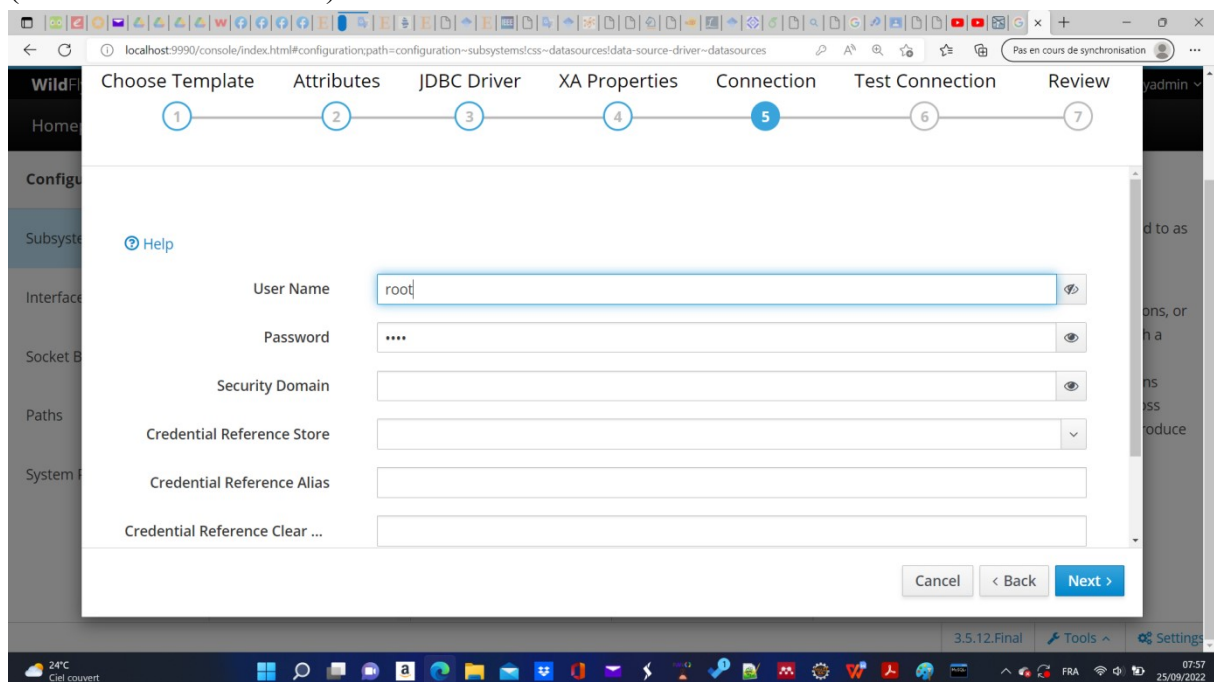


- Dans l'interface suivante, ajoutez les propriétés relatives à la connexion à notre base de données JpaBase initialement créée. Les propriétés à ajouter sont :

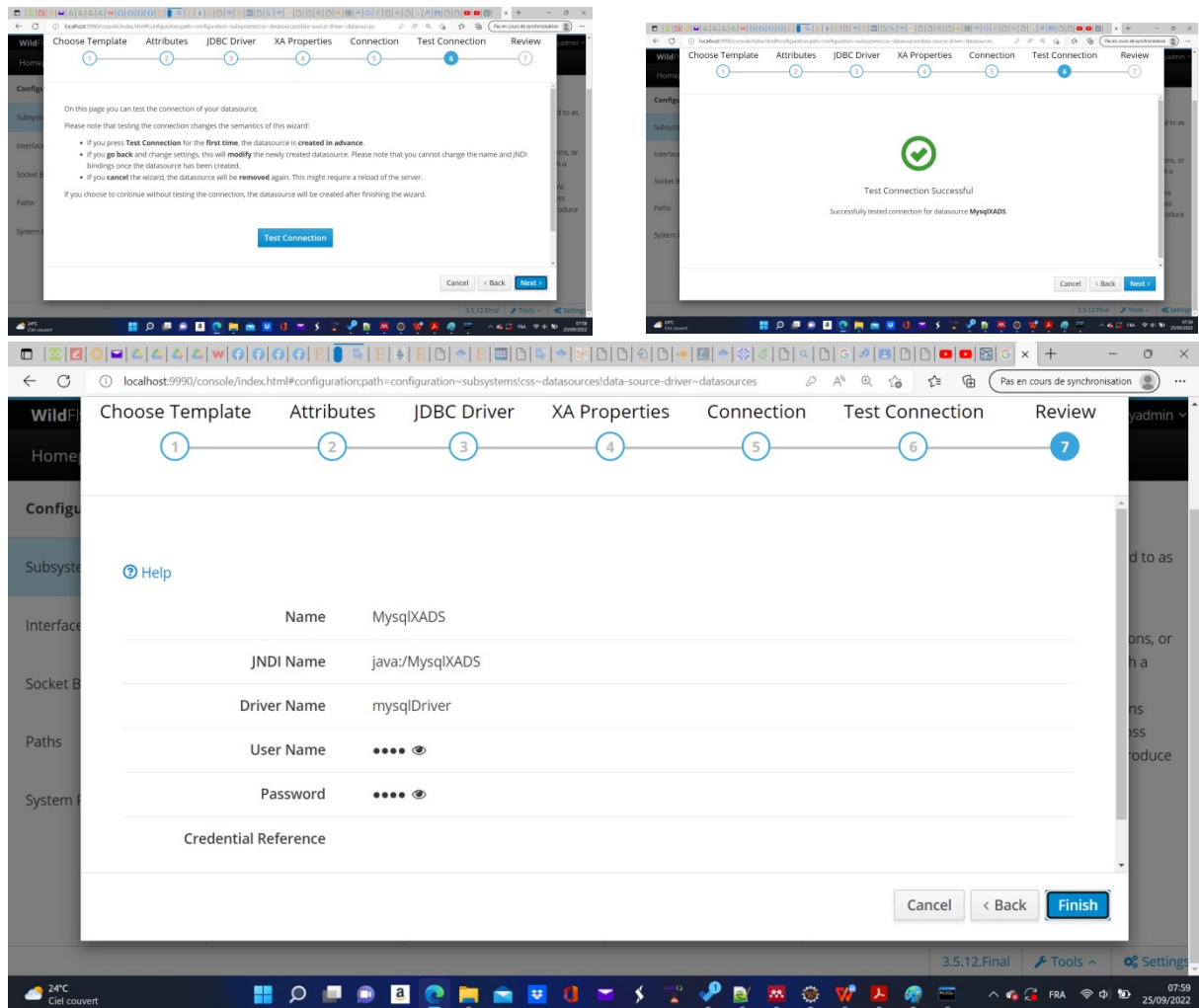
serverName=localhost
 portNumber=3306
 databaseName=JpaBase



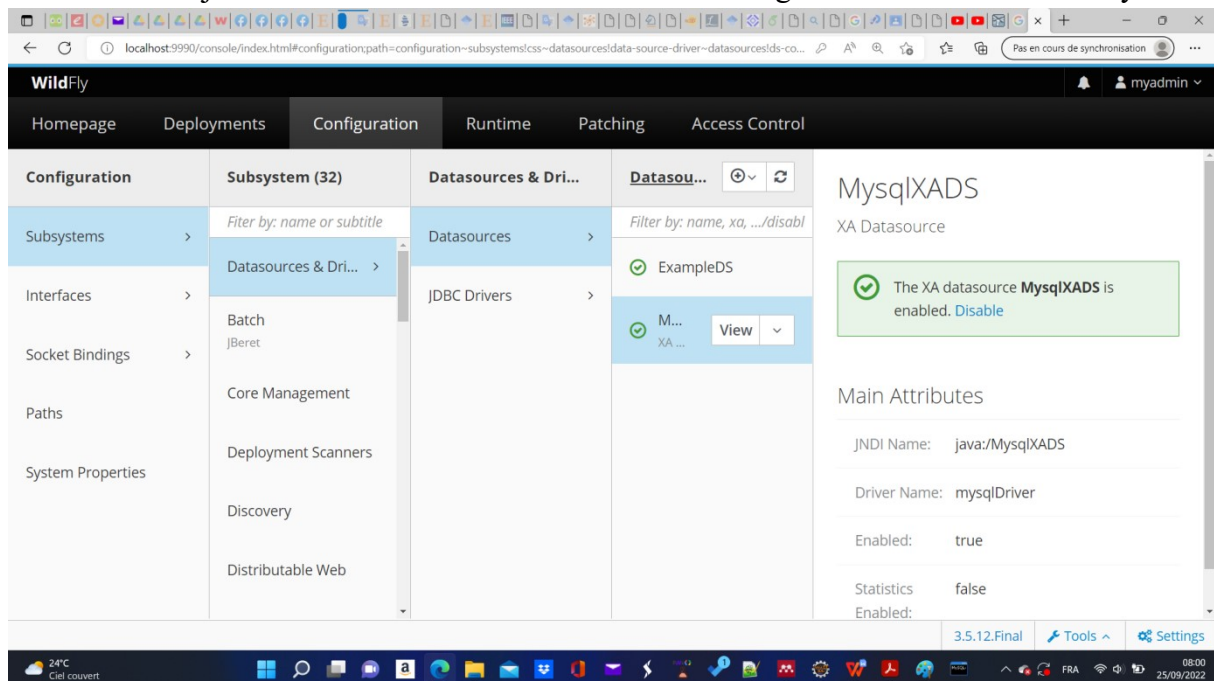
- Dans l'interface suivante, indiquez les paramètres de connexion à votre serveur MySQL (User Name et Password).



- Testez ensuite la connexion et terminez l'assistant.

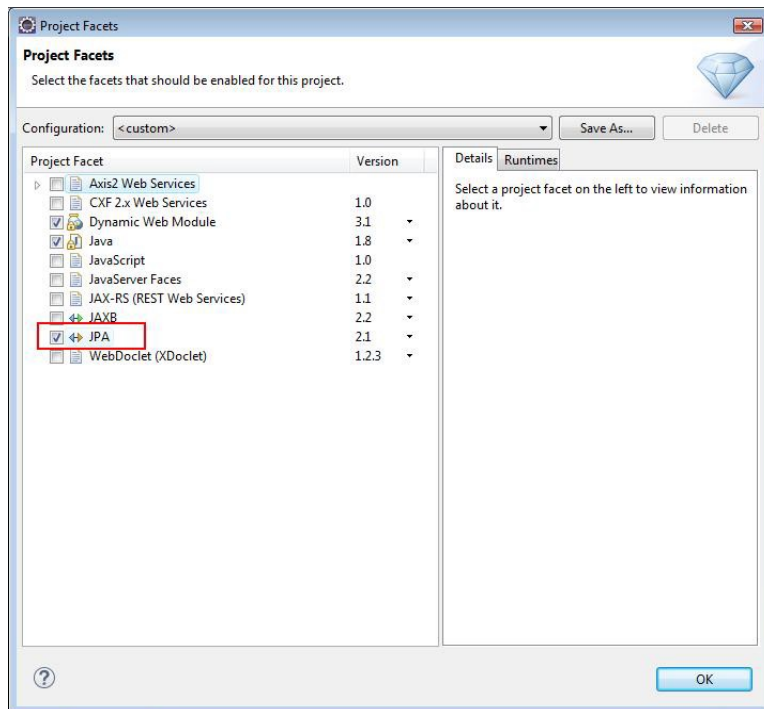


- Recharger la nouvelle configuration du serveur. Remarquez qu'une nouvelle source de données a été ajoutée à la liste des sources de données configurées au niveau de Wildfly.



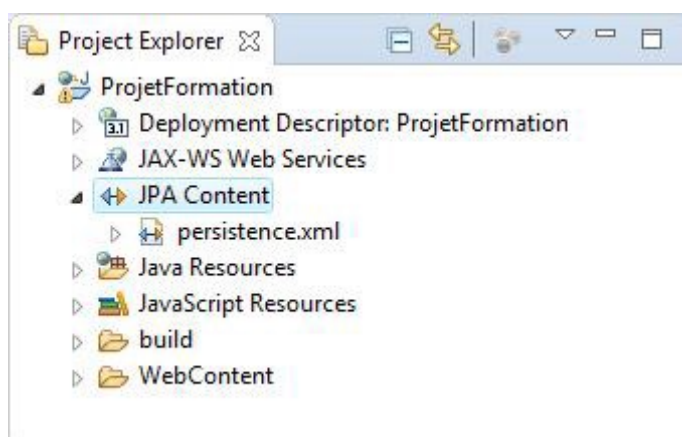
3. Génération et utilisation des entités JPA

Au niveau de Eclipse activer la facette **JPA** pour votre projet web comme indiqué par la figure suivante et validez.

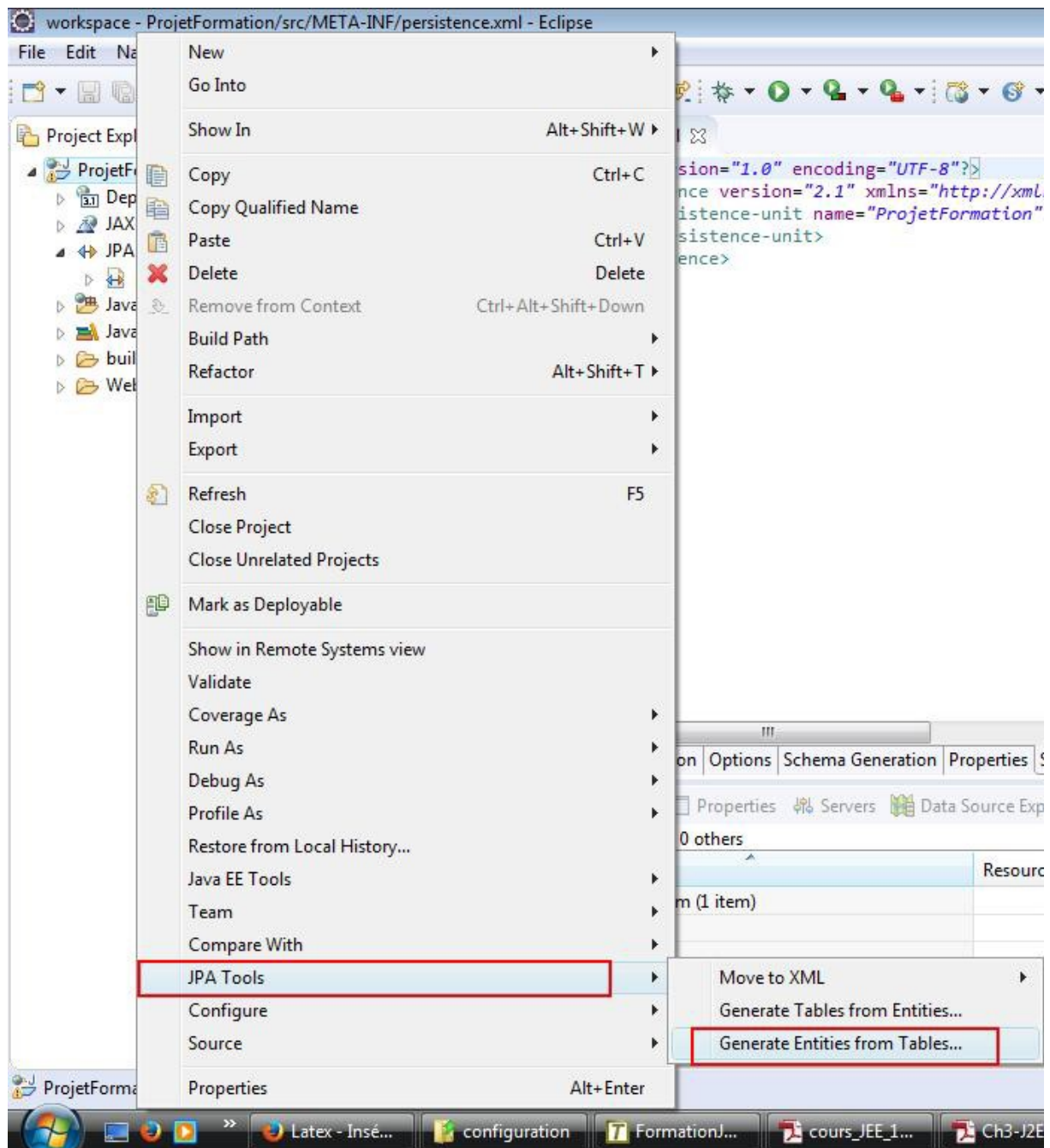


3.1 Génération des entités JPA

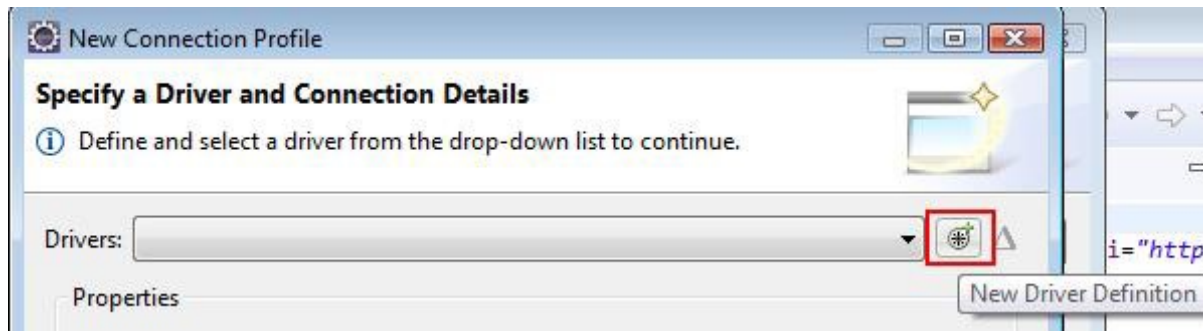
Si on examine la structure du projet créé, on remarque qu'il existe une sous arborescence intitulée **JPA content**.



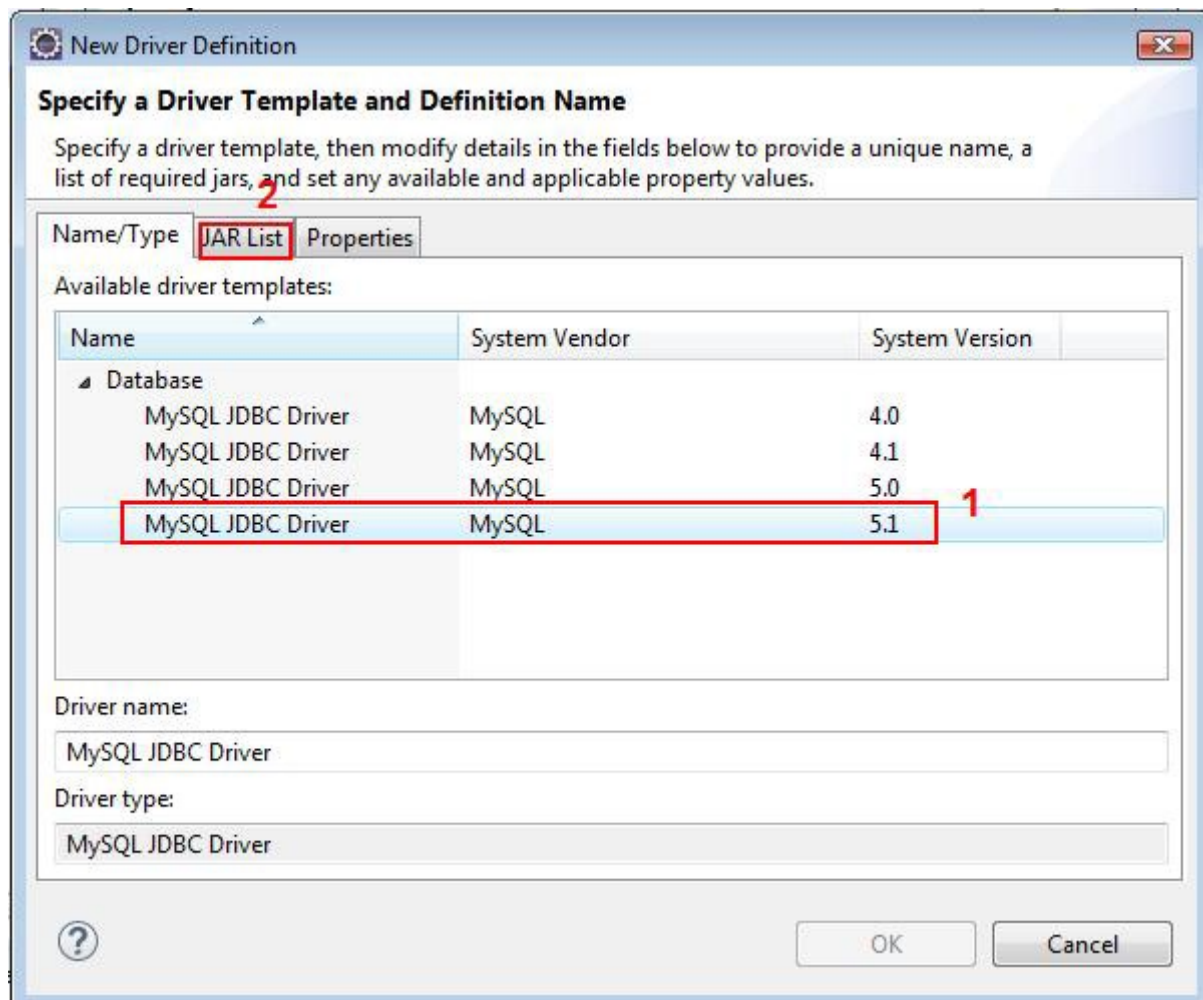
Pour générer les beans entités relatifs aux tables de la base, il suffit de faire : **Clic droit sur le projet → JPA Tools → Generate entities from tables**



Dans l'interface suivante, cliquez sur l'icône d'ajout d'une connexion comme indiqué dans la figure suivante.



Dans l'interface suivante, sélectionnez le nom du connecteur à utiliser (celui correspondant à la version 5.1), ensuite fournissez à partir de la tabulation **Jar List** le connecteur à partir de son emplacement physique. Validez.



Dans l'interface suivante, modifiez le nom de la base (dans notre cas c'est **JpaBase**), saisissez le mot de passe pour accéder au serveur MySQL et testez la connexion créée. Un message est affiché indiquant si la connexion est réussite ou non.

Terminez l'assistant de création d'une nouvelle connexion.

Revenons à l'assistant de génération des entités, sélectionnez le schéma "**JpaBase**" et la table Learner de la base. Validez.

L'interface suivante modélise les relations qui existent entre les tables de la base. Si la base a été correctement créée alors les associations seront correctement générées. Validez.

Dans l'interface suivante, sélectionnez **"identity"** en tant que mode de génération de la valeur de la clé primaire et **"Eager"** en tant que mode de génération d'une propriété de type Collection résultante d'une jointure entre deux tables. Terminez l'assistant.

The screenshot shows the 'Generate Custom Entities' wizard, specifically the 'Customize Defaults' step. The window title is 'Generate Custom Entities'. The subtitle is 'Customize Defaults'. Below the subtitle, it says: 'Optionally customize aspects of entities that will be generated by default from database tables. A Java package should be specified.'

The 'Mapping defaults' section contains the following options:

- Key generator: **identity** (selected in a dropdown)
- Sequence name: (empty text field)
- Entity access: **Field** (selected radio button), Property (unselected)
- Associations fetch: Default (unselected), **Eager** (selected radio button), Lazy (unselected)
- Collection properties type: java.util.Set (unselected), **java.util.List** (selected radio button)
- ☐ Always generate optional JPA annotations and DDL parameters

The 'Domain java class' section contains the following options:

- Source folder: MyFirstEJBWS/src/main/java (with a 'Browse...' button)
- Package: tn.jpa.entities (with a 'Browse...' button)
- Superclass: (empty text field, with a 'Browse...' button)
- Interfaces: java.io.Serializable (with an 'Add...' button and a 'Remove' button)

At the bottom of the wizard, there are four buttons: '?', '< Back', 'Next >', and 'Finish' (highlighted in blue), and a 'Cancel' button.

On remarque aussi qu'un fichier **persistence.xml** a été créé contenant les paramètres de configuration d'une nouvelle **unité de persistance** `<persistence-unit`

`name="Unit_Name">`. A ce stade il faut modifier le fichier de persistance afin d'associer la source de données précédemment configurée sous Wildfly (identifiée par son JNDI name **java:/MysqlXADS**) à l'unité de persistance qui vient d'être créée. Procédez comme indiqué dans l'interface ci-dessous.

The screenshot shows the 'Persistence Unit Connection' configuration window in Wildfly. The window has a title bar with several tabs, including '*persistence.xml' which is highlighted with an orange box. The main content area is titled 'Persistence Unit Connection' and contains the following fields:

- Transaction type:** A dropdown menu set to 'JTA', highlighted with an orange box.
- Database:** A section containing:
 - JTA data source:** A text field containing 'java:/MysqlXADS', highlighted with an orange box.
 - Non-JTA data source:** An empty text field.
- JDBC connection properties:** A section with a link 'Populate from connection...' and four input fields: 'Driver', 'URL', 'User', and 'Password'.

At the bottom of the window is a tabbed interface with the following tabs: 'General', 'Connection' (highlighted with an orange box), 'Options', 'Schema Generation', 'Properties', 'Schema Generation', and 'Source'.