

TP07

Les composants « EJB Entity »

Objectifs

Réalisation de la persistance des données en utilisant un EJB Entity

1. Préparer l'environnement de développement EJB

- Installer un serveur EJB **JBoss WildFly** (moteur EJB)
Configurer Eclipse avec des outils JBoss **WildFly**

2. Persister les données :

- Configurer une source de données (datasource) au niveau du serveur WildFly
- Ecrire un EJB Entity (JPA Entity)
- Configurer la procédure de persistance et la communication avec la source de données

A. Préparer l'environnement de développement EJB (Ajout des outils du serveur WildFly à eclipse)

1. Télécharger, dans le dossier «D:\Atelier_JEE\Servers», l'archive de la version **18** du serveur (JBoss) **WildFly** à travers le lien suivant :

<https://download.jboss.org/wildfly/18.0.1.Final/wildfly-18.0.1.Final.zip>

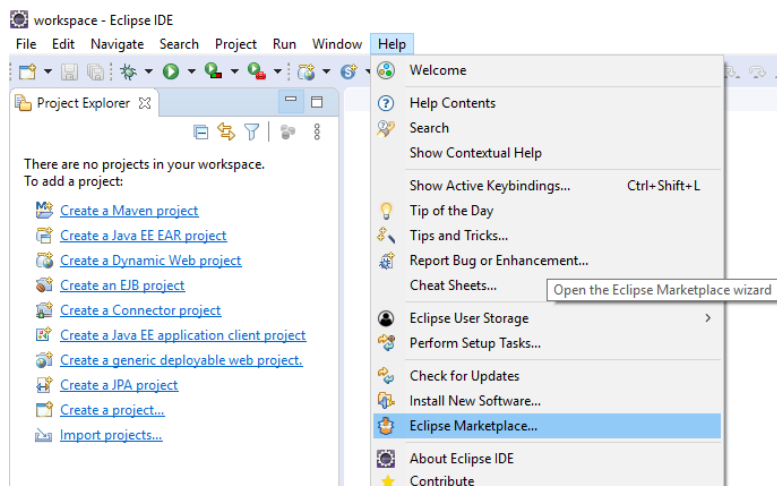
2. Extraire l'archive dans le même dossier «D:\Atelier_JEE\Servers».

3. Pour ajouter le plugin **JbBoss (WildFly)** à **eclipse**:

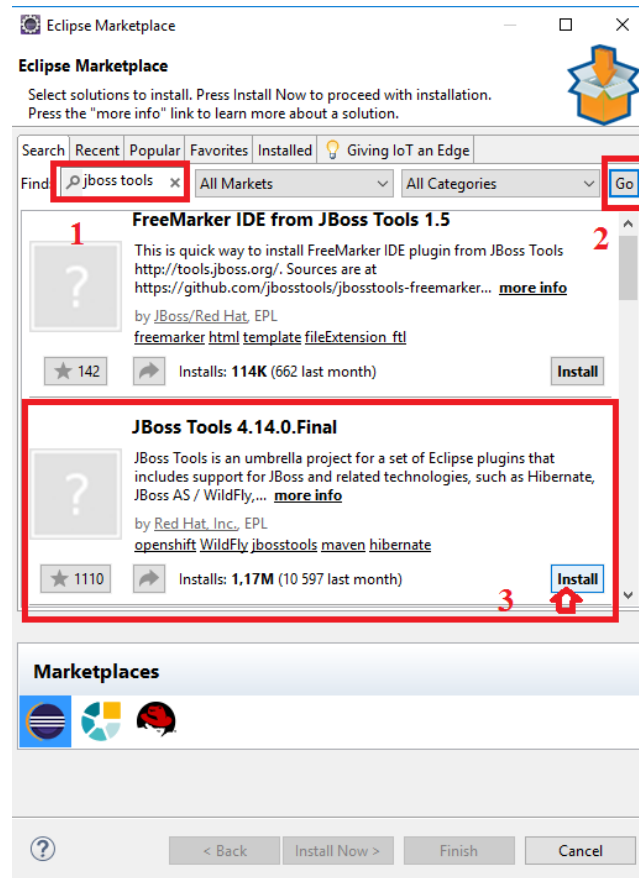
a. Ouvrir **eclipse** dans le workspace «D:\Atelier_JEE\Workspace».

b. Aller au menu **Help**.

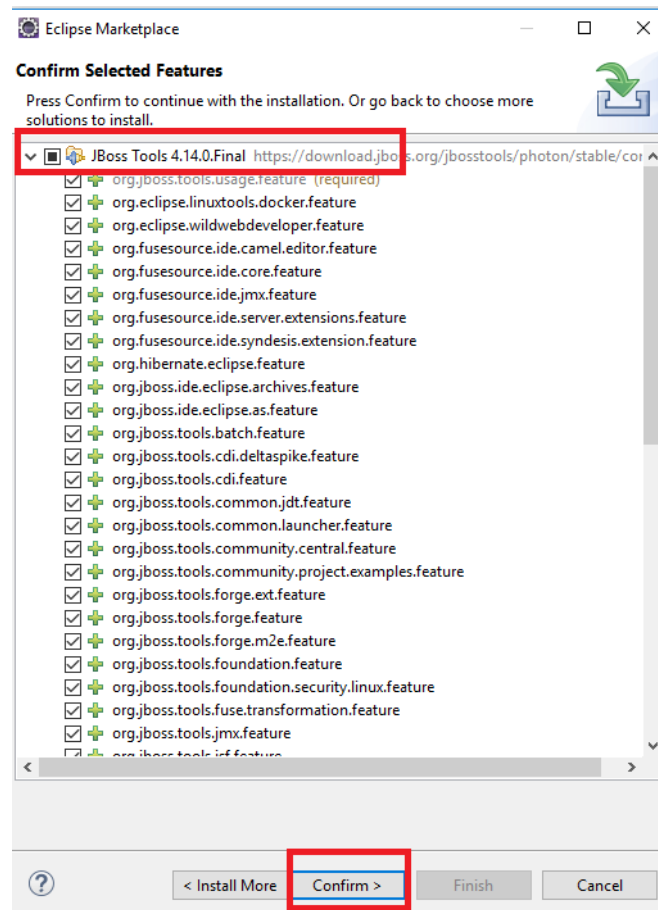
c. Aller à la commande «**Eclipse Marketplace..** ».



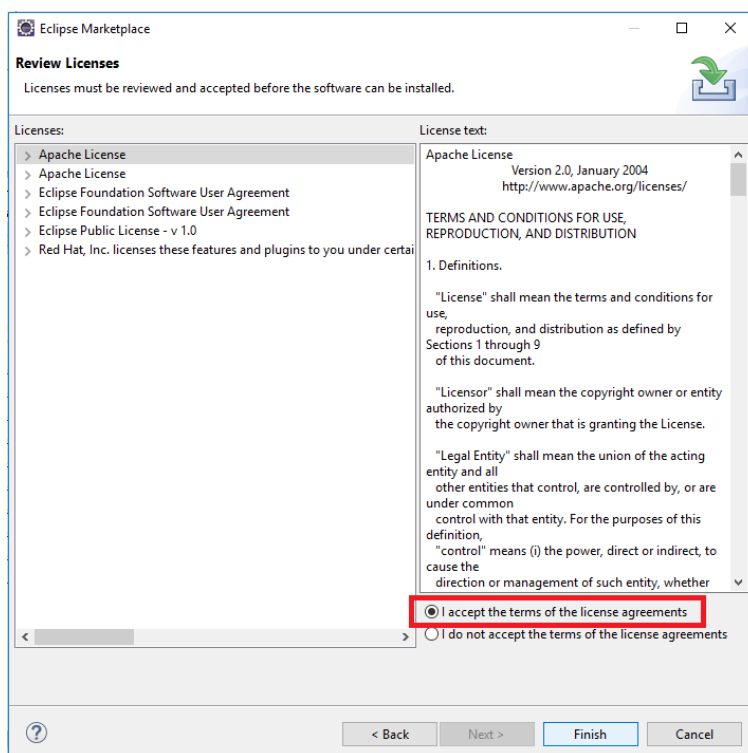
- d. Chercher le plugin «JBoss Tools» dans le magasin de eclipse et lancer l'installation :



- e. Confirmer l'installation des outils de JBoss :



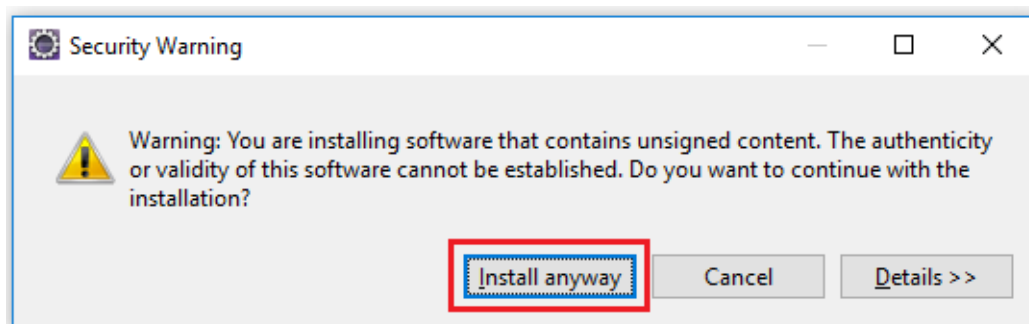
f. Accepter les termes de licence :



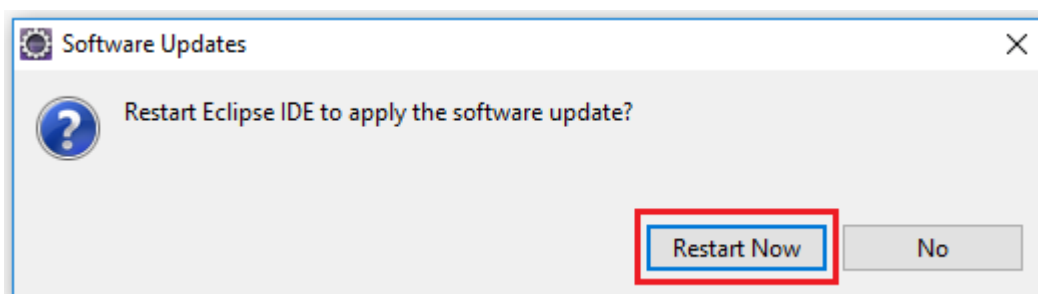
g. Attendre l'installation des outils de JBoss (suivre l'avancement d'installation en bas à droite) :



h. Continuer l'installation du contenu non signé..

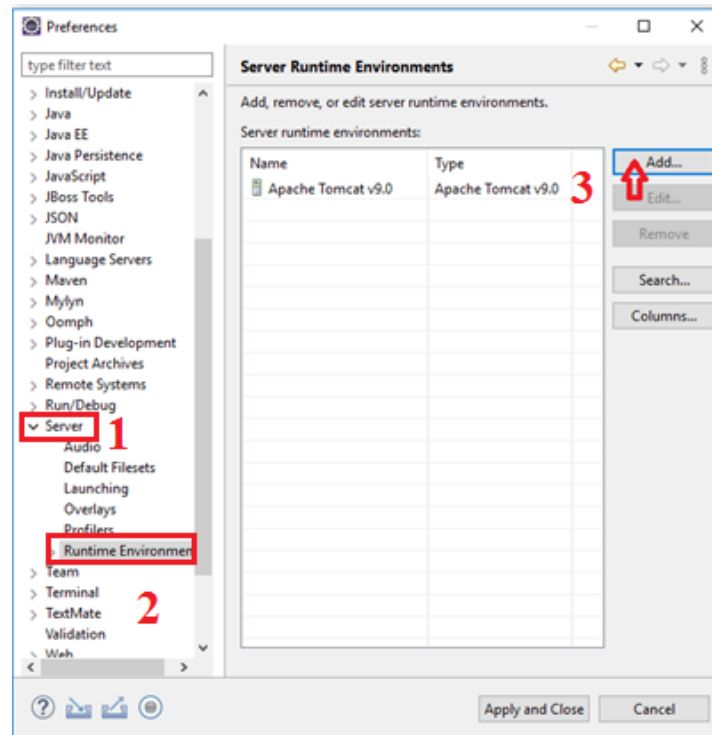


i. Une fois l'installation es terminée, redémarrer eclipse pour la mise à jour :

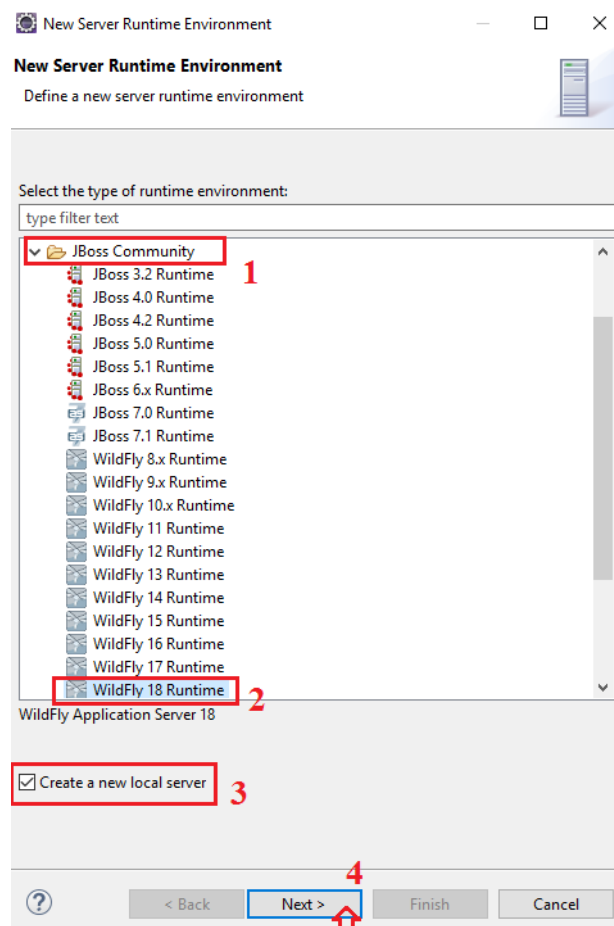


4. Pour utiliser le serveur **WildFly 18** avec **eclipse** :

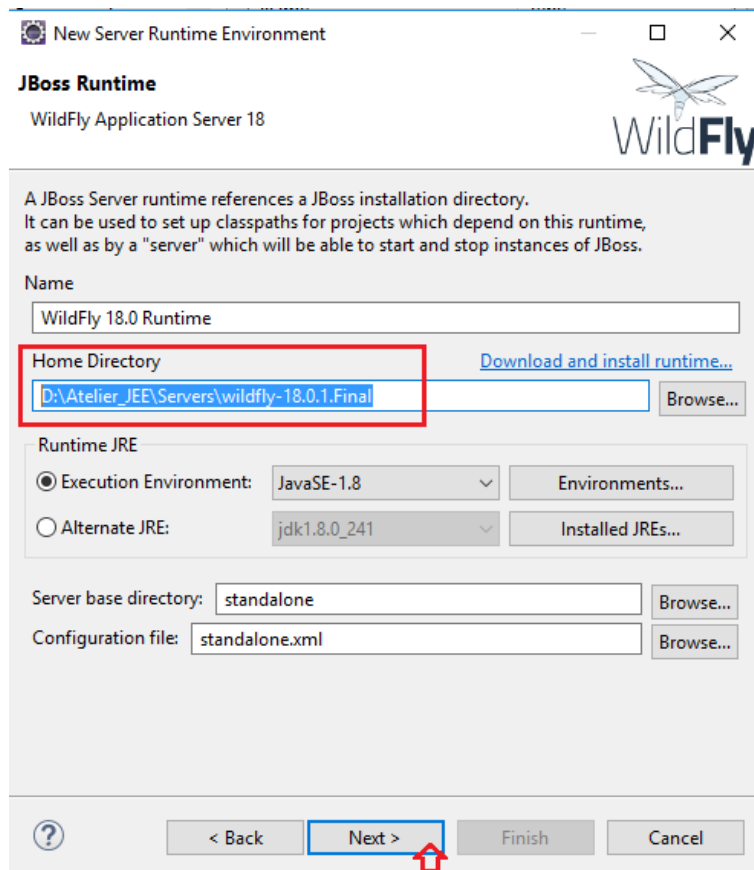
- a. Aller au menu « **Window** » puis à la commande « **Preferences** »
- b. Dans la liste située à gauche, choisir « **Server** » puis l'option «**Runtime Environments** » et appuyer sur le bouton «**Add**»:



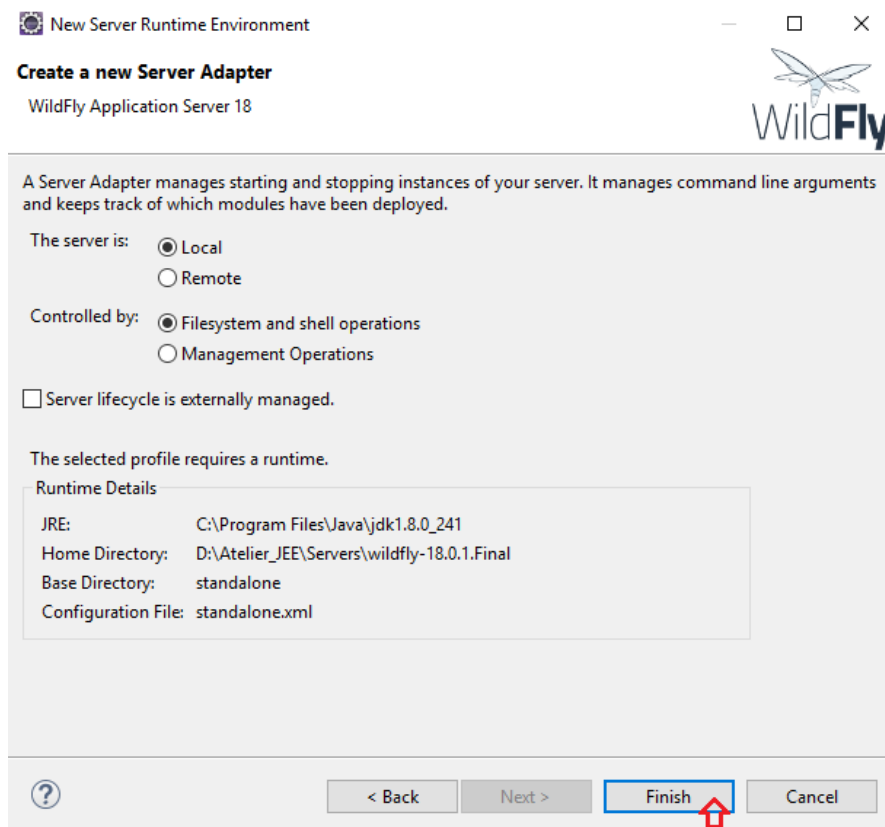
- c. Sélectionner la version **18** de **WildFly** (sous «**JBoss Community**»), cocher l'option «**create a new local server**» et appuyer sur «**Next**» :



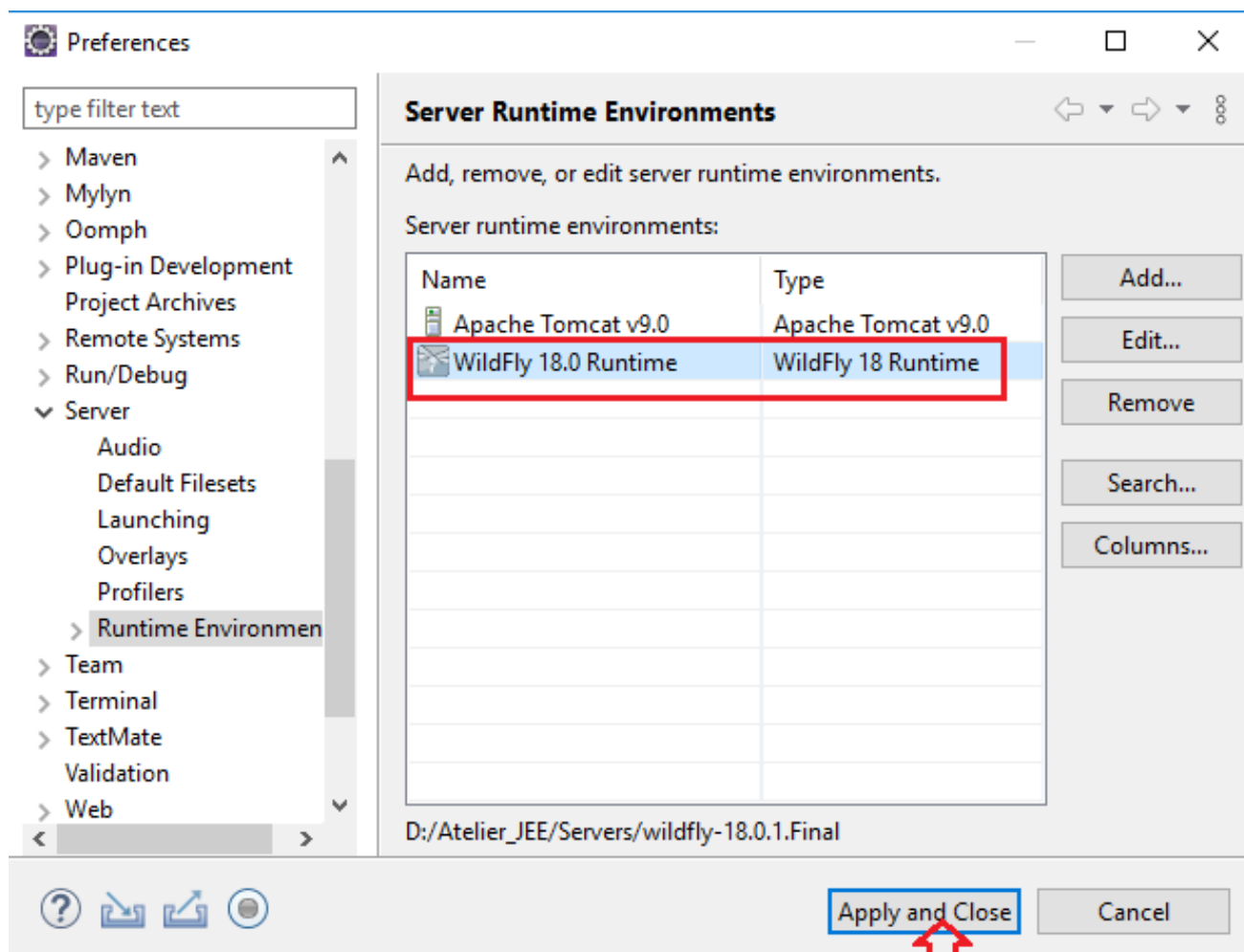
- d. Spécifier le dossier d'installation de **WildFly 18** : aller à travers le bouton « **Browse** », sélectionner le dossier racine du **WildFly 18** déjà créé (**D:\Atelier_JEE\Servers\wildfly-18.0.1.Final**) et appuyer sur «**Next**»:



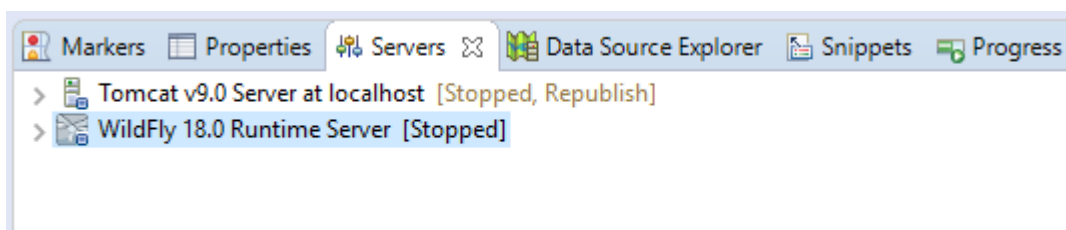
- e. Appuyer sur « **Finish** » :



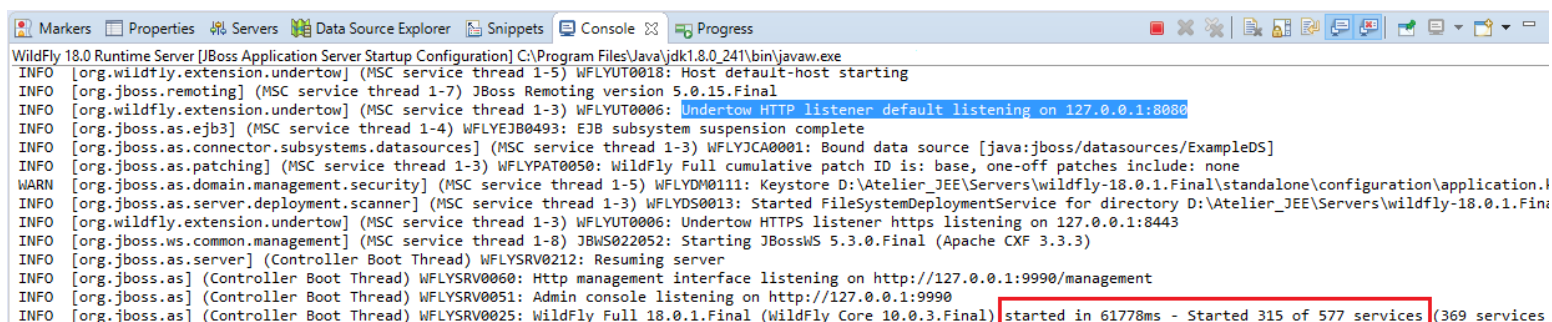
f. Appuyer sur « **Apply and Close** » pour achever la procédure d'ajout du serveur **WildFly** à l'EDI **eclipse**.



5. Remarquer l'ajout de **WildFly 18** dans le volet «**Servers**» de **eclipse**:



6. Pour démarrer **WildFly**, Cliquer par le bouton droit sur le serveur puis Choisir la commande « **Start** ». Si tout va bien, vous recevez dans la vue **Console** un message pareil :



B. Configuration d'une source de données (datasource)

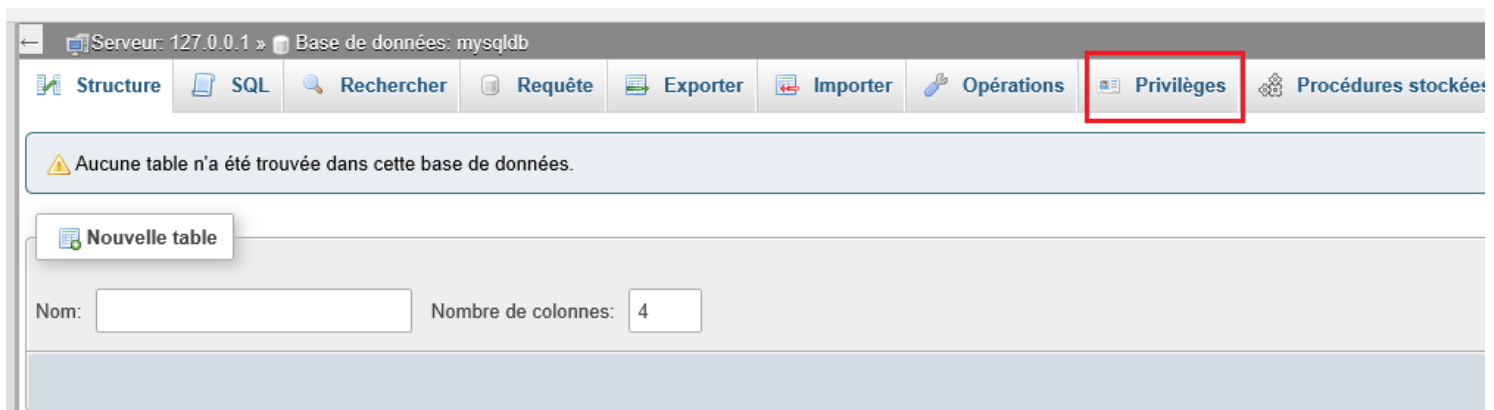
Une source de données (**datasource**) est une unité logicielle (gérée par le serveur **WildFly**) qui constitue un intermédiaire entre le programme **JAVA** et la base de données en question.

7. **Etape1** : Créer une base de données MySQL nommée «**MySQLDB**» (en utilisant l'interface web « **phpmyadmin** » de **XXAMP**)

8. **Etape2** : Créer un compte utilisateur ayant les propriétés suivantes :

- Nom d'utilisateur =**admin**
- Mot de passe =**admin**

- Accéder au volet « **Privilèges** » :



- Ajouter un compte d'utilisateur



- Donner tous les privilèges au compte (**admin, admin**) :

Serveur: 127.0.0.1

Bases de données SQL État Comptes utilisateurs Exporter Importer Paramètres Réplication Variables

Ajouter un compte d'utilisateur

Informations pour la connexion

Nom d'utilisateur : Saisir une valeur: admin

Nom d'hôte : Tout hôte %

Mot de passe : Saisir une valeur: **** Strength: Extrêmement faible

Saisir à nouveau : ****

Extension d'authentification : Authentification MySQL native

Générer un mot de passe: Générer

Base de données pour ce compte d'utilisateur

☐ Créer une base portant son nom et donner à cet utilisateur tous les privilèges sur cette base.

☐ Accorder tous les privilèges à un nom passe-partout (utilisateur_%).

☒ Donner tous les privilèges sur la base de données mysqldb.

Privilèges globaux ☒ Tout cocher

NB : les noms de privilèges sont exprimés en anglais.

☒ Données ☒ Structure ☒ Administration ☐ Limites de ressources

☒ SELECT ☒ CREATE ☒ GRANT

NB : une valeur de 0 (zéro) lève la limite.

- Appuyer sur «**Exécuter**» et remarquer l'ajout du nouvel utilisateur:

Serveur: 127.0.0.1 Base de données: mysqldb

Structure SQL Rechercher Requête Exporter Importer Opérations Privilèges Procédures stoc

Utilisateurs ayant accès à « mysqldb »

Nom d'utilisateur	Nom d'hôte	Type	Privilèges	« Grant »	Action
<input type="checkbox"/> admin	%	global	ALL PRIVILEGES	Oui	Éditer les privilèges Exporter
		spécifique à cette base de données	ALL PRIVILEGES	Non	Éditer les privilèges Exporter
<input type="checkbox"/> root	127.0.0.1	global	ALL PRIVILEGES	Oui	Éditer les privilèges Exporter
<input type="checkbox"/> root	::1	global	ALL PRIVILEGES	Oui	Éditer les privilèges Exporter
<input type="checkbox"/> root	localhost	global	ALL PRIVILEGES	Oui	Éditer les privilèges Exporter

☐ Tout cocher Avec la sélection : Exporter

9. Etape3 : Définir un module pour mySQL :

a. Accéder au dossier :

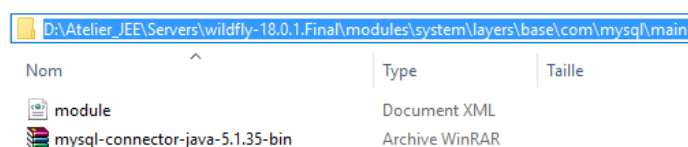
«D:\Atelier_JEE\Servers\wildfly-18.0.1.Final\modules\system\layers\base\com »

b. Créer un sous-dossier «mysql»

c. Créer un sous-dossier «main» dans «mysql»

d. Placer dans le dossier «main» les deux fichiers suivants (données en pièce jointe :

- mysql-connector-java-5.1.35-bin.jar (pilote de mysql)
- Module.xml (fichier XML pour définir un nom de module «com.mysql» associé au pilote de mysql).

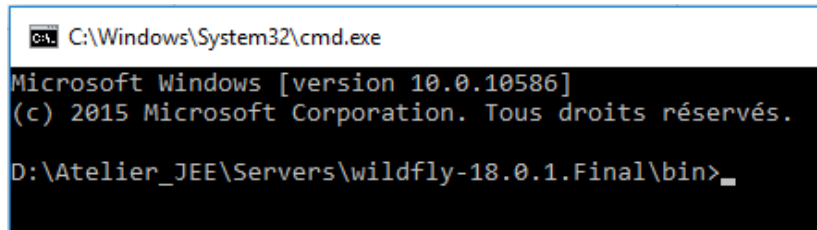


Voici un aperçu du fichier « **module.xml** » :

```
<module xmlns="urn:jboss:module:1.1" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java-5.1.35-bin.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
    <module name="javax.servlet.api" optional="true"/>
  </dependencies>
</module>
```

10. Etape 4 : Définir un driver (jdbc-driver) :

- a. Accéder , en mode CONSOLE, au dossier « **bin** » du serveur WildFly :

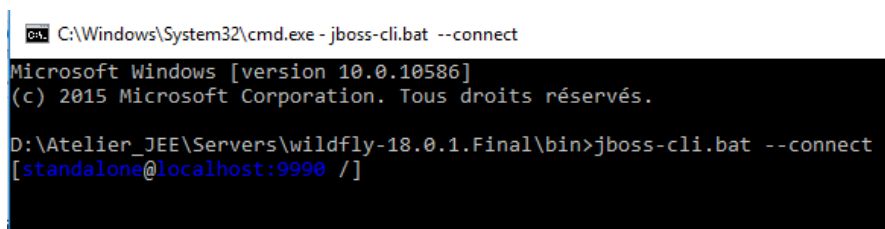


```
C:\Windows\System32\cmd.exe
Microsoft Windows [version 10.0.10586]
(c) 2015 Microsoft Corporation. Tous droits réservés.

D:\Atelier_JEE\Servers\wildfly-18.0.1.Final\bin>
```

- b. Se connecter comme un client Jboss en tapant la commande suivante :

jboss-cli.bat --connect



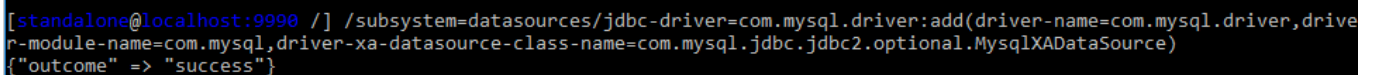
```
C:\Windows\System32\cmd.exe - jboss-cli.bat --connect
Microsoft Windows [version 10.0.10586]
(c) 2015 Microsoft Corporation. Tous droits réservés.

D:\Atelier_JEE\Servers\wildfly-18.0.1.Final\bin>jboss-cli.bat --connect
[standalone@localhost:9990 /]
```

- c. Enregistrer un pilote (**driver-jdbc**) nommé « **com.mysql.driver** » associé au module « **com.mysql** » déjà créé en tapant la commande suivante :

/subsystem=datasources/jdbc-driver=com.mysql.driver:add(driver-name=com.mysql.driver,driver-module-name=com.mysql,driver-xa-datasource-class-name=com.mysql.jdbc.jdbc2.optional.MysqlXADataSource)

- d. Si tout va bien vous recevez un message qui indique le succès de l'opération :



```
[standalone@localhost:9990 /] /subsystem=datasources/jdbc-driver=com.mysql.driver:add(driver-name=com.mysql.driver,driver-module-name=com.mysql,driver-xa-datasource-class-name=com.mysql.jdbc.jdbc2.optional.MysqlXADataSource)
{"outcome" => "success"}
```

11. Etape 5 : Définir un compte administrateur pour accéder à la console d'administration de WildFly :

- a. Taper **ctr+c** pour retourner, en mode CONSOLE, au dossier « **bin** » du serveur WildFly.
- b. Taper la commande suivante :

add-user.bat

```
D:\Atelier_JEE\Servers\wildfly-18.0.1.Final\bin>add-user.bat

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

c. Taper « **a** » pour choisir un utilisateur de type « **Management User** » :

```
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 : _
```

d. Taper « **jbossadmin** » (comme un nom d'utilisateur) puis taper de même « **jbossadmin** » (comme mot de passe) :

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

- 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-alf
)
Password :
WFLYDM0098: The password should be different from the username
Are you sure you want to use the password entered yes/no?
```

e. Taper « **yes** » puis retaper le mot de passer pour confirmer :

```
Password :
WFLYDM0098: The password should be different from the username
Are you sure you want to use the password entered yes/no? yes
Re-enter Password :
What groups do you want this user to belong to? (Please enter a comma separated list, or leave blank for none)[ ]:
```

f. Taper la touche « **ENTREE** » du clavier :

```
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 'jbossadmin' for realm 'ManagementRealm'
Is this correct yes/no?
```

g. Taper « **yes** »

```
About to add user 'jbossadmin' for realm 'ManagementRealm'
Is this correct yes/no? yes
Added user 'jbossadmin' to file 'D:\Atelier_JEE\Servers\wildfly-22.0.1.Final\standalone\configuration\mgmt-users.prop
ities'
Added user 'jbossadmin' to file 'D:\Atelier_JEE\Servers\wildfly-22.0.1.Final\domain\configuration\mgmt-users.properties'
Added user 'jbossadmin' with groups to file 'D:\Atelier_JEE\Servers\wildfly-22.0.1.Final\standalone\configuration\mg
groups.properties'
Added user 'jbossadmin' with groups to file 'D:\Atelier_JEE\Servers\wildfly-22.0.1.Final\domain\configuration\mgmt-g
ps.properties'
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 EJB calls
yes/no?
```

h. Taper « **yes** »

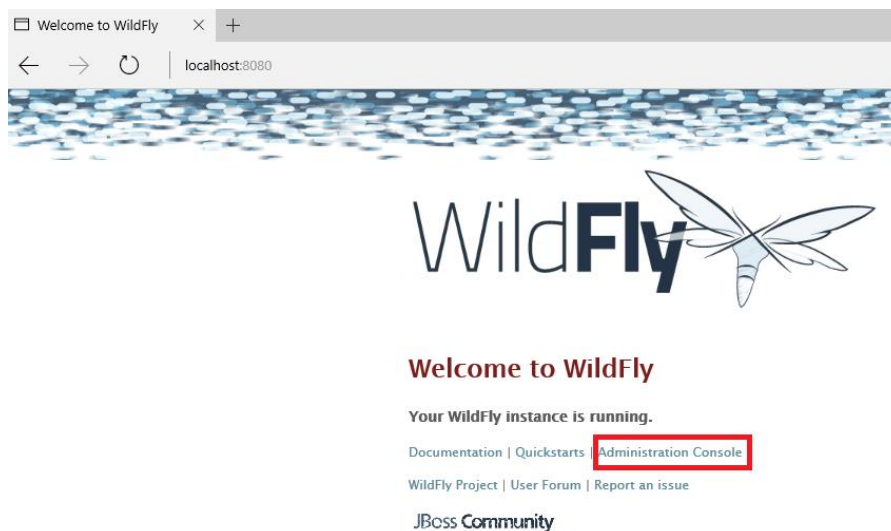
```
yes/no? yes
To represent the user add the following to the server-identities definition <secret value="amJvc3NhZG1pbG==" />
Appuyez sur une touche pour continuer...
```

i. Taper une touche pour terminer.

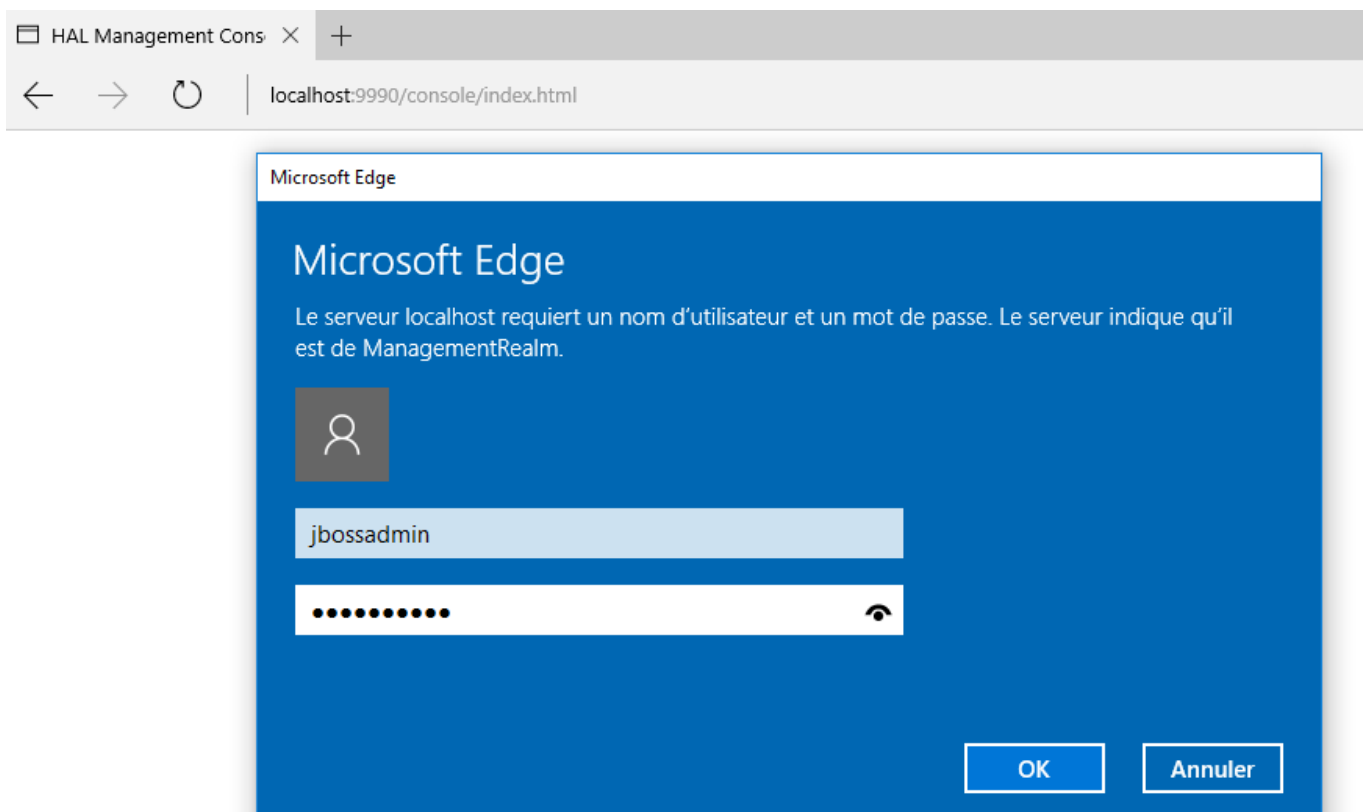
j. Redémarrer le serveur **WildFly** et accéder à la page d'accueil en tapant l'url suivant :

<http://localhost:8080/>

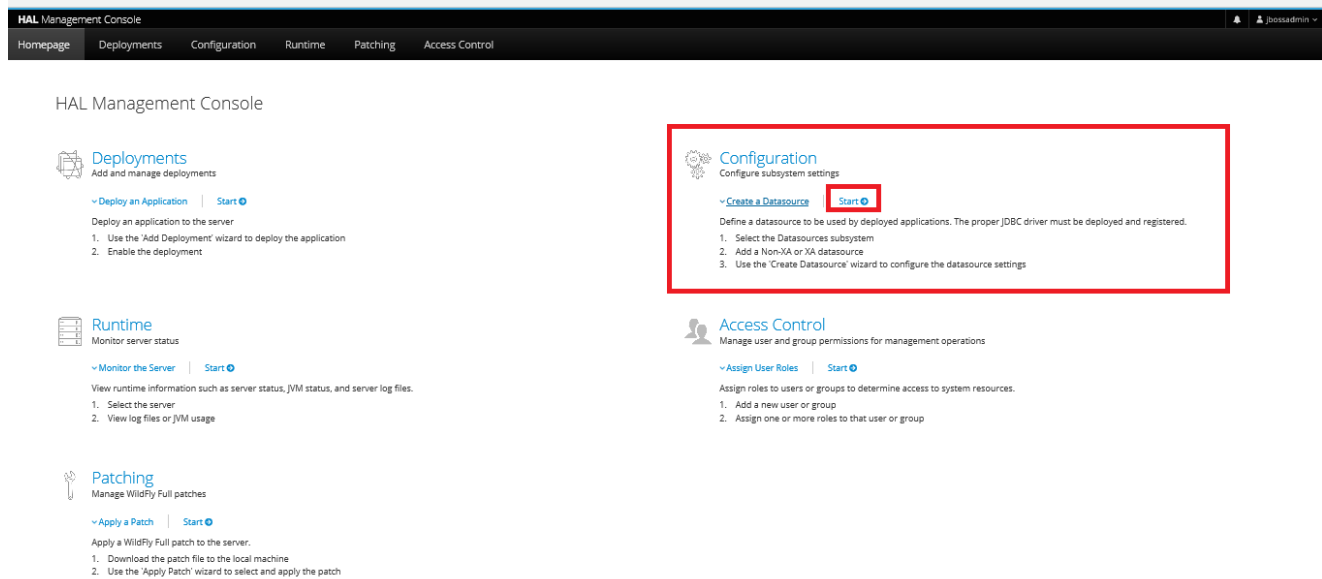
12. Accéder à la console d'administration :



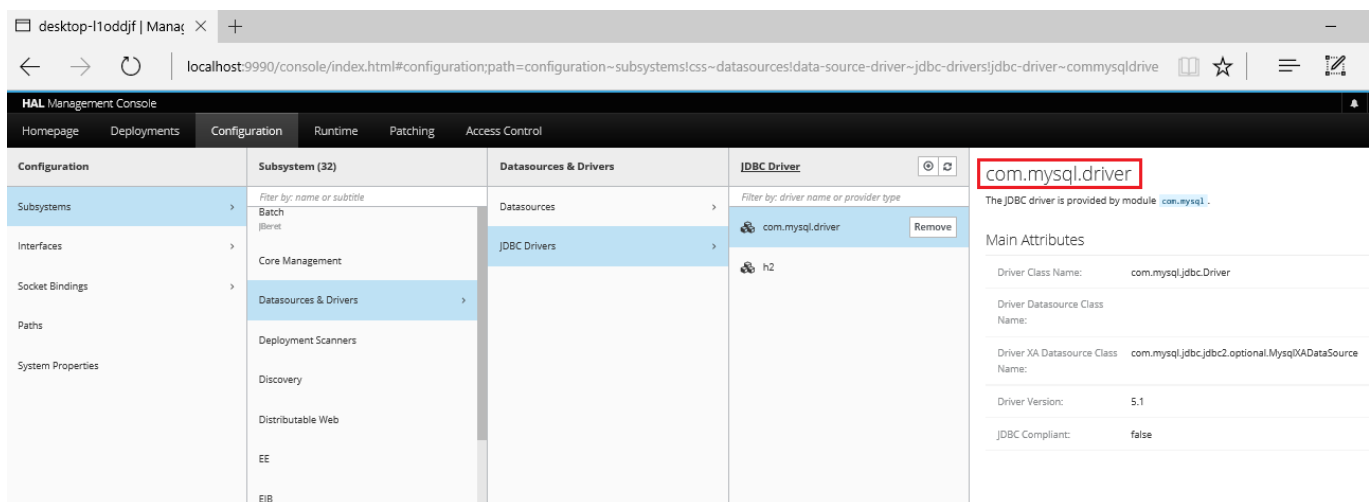
13. Saisir les paramètres de connexion (**jbossadmin**, **jbossadmin**) pour l'utilisateur déjà créé :



14. Accéder au lien « Start » dans le volet de « configuration » pour commencer la création d'une datasource :



15. Vérifier l'existence du driver « **com.mysql.driver** » nouvellement créé :

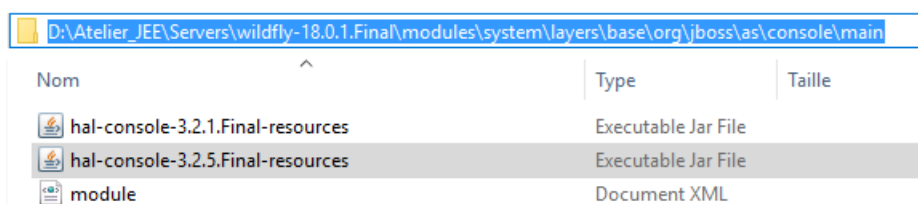


Cet affichage est bien testé avec le navigateur Microsoft Edge. Il se peut qu'avec d'autres navigateurs WEB la liste des drivers JDBC n'apparaissent pas. Ceci est dû à un BUG dans la console de WildFly18. (Si tout va bien passer directement à l'étape 6)

Pour remédier à ce problème :

1. Copier le fichier « **hal-console-3.2.5.Final-resources.jar** » (donné en pièce jointe) dans le dossier :

D:\Atelier_JEE\Servers\wildfly-18.0.1.Final\modules\system\layers\base\org\jboss\as\console\main



2. Editer le fichier «module.xml», situé dans le même dossier, pour utiliser le nouveau fichier JAR copié :

```
<module name="org.jboss.as.console" xmlns="urn:jboss:module:1.5">
  <properties>
    <property name="jboss.api" value="private"/>
  </properties>

  <resources>
    <resource-root path="hal-console-3.2.5.Final-resources.jar"/>
  </resources>

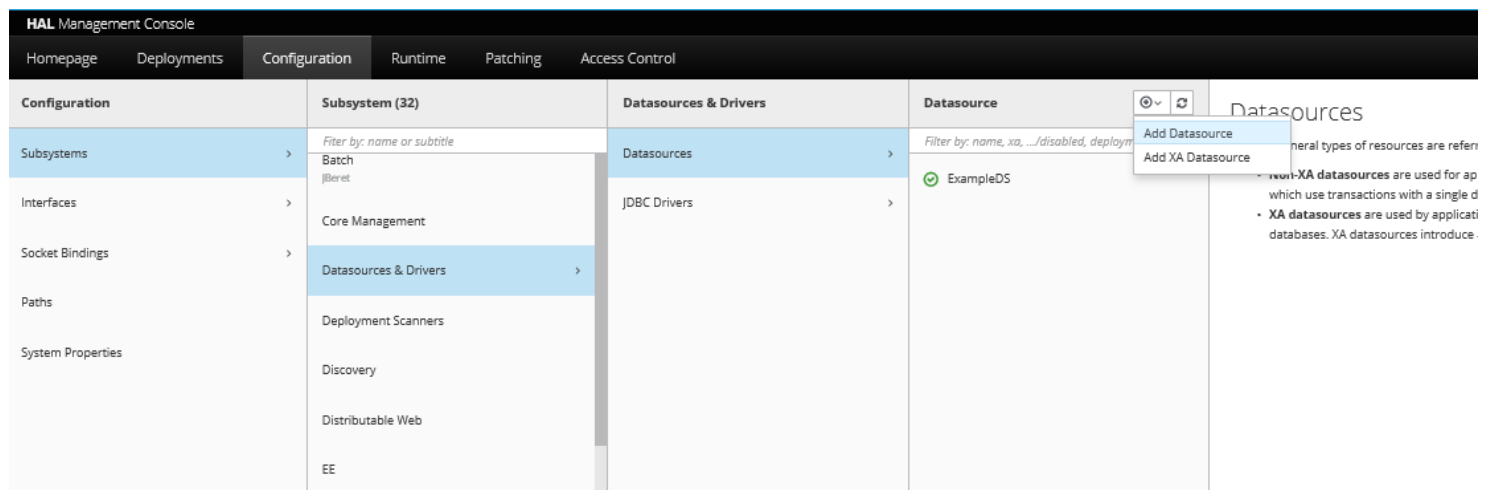
  <dependencies>
    <!-- Do not add dependencies to the console as resources in any
         dependencies could be served up through the HTTP interface. -->
  </dependencies>
</module>
```

3. Redémarrer le serveur WildFly 18.

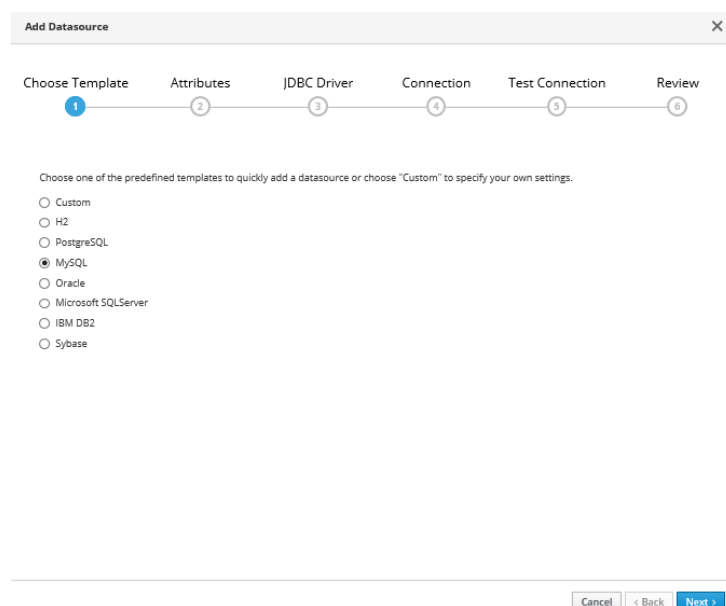
4. forcer le rafraîchissement du cache du navigateur (vider les données de navigation) .

5. Accéder, maintenant, à la console d'administration de WildFly pour visualiser les drivers JDBC et continuer les étapes qui suivent.

16. **Etape6** : Ajouter une nouvelle source de données (Add Datasource) :



a. Sélectionner le type « **MySQL** » puis appuyer sur « **Next** » :



b. Attribuer le nom «**MySqlDS**» à la source de données puis appuyer sur « **Next** » :

Add Datasource

X

Choose Template1Attributes2JDBC Driver3Connection4Test Connection5Review6

Help

Name *MySqlDS

JNDI Name *java:/MySqlDS

Required fields are marked with *

Cancel< BackNext >

c. Choisir le driver-jdbc « **com.mysql.driver** » puis appuyer sur « **Next** » :

Add Datasource

X

Choose Template1Attributes2JDBC Driver3Connection4Test Connection5Review6

Help

Driver Name *com.mysql.driver

Driver Module Namecom.mysql

Driver Class Namecom.mysql.jdbc.Driver

Required fields are marked with *

Cancel< BackNext >

d. Spécifier l'url de connexion à la base de données «**MySQLDB** » puis appuyer sur « **Next** » :

Add Datasource

Choose Template

Attributes

JDBC Driver

Connection

Test Connection

Review

1

2

3

4

5

6

Help

Connection URL

jdbc:mysql://localhost:3306/mysqlpdb

User Name

admin

Password

admin

Security Domain

Cancel

< Back

Next >

e. Tester la connexion :

Add Datasource

Choose Template

Attributes

JDBC Driver

Connection

Test Connection

Review

1

2

3

4

5

6

On this page you can test the connection of your datasource.

Please note that testing the connection changes the semantics of this wizard:

- If you press **Test Connection** for the **first time**, the datasource is **created in advance**.
- If you **go back** and change settings, this will **modify** the newly created datasource. Please note that you cannot change the name and JNDI bindings once the datasource has been created.
- If you **cancel** the wizard, the datasource will be **removed** again. This might require a reload of the server.

If you choose to continue without testing the connection, the datasource will be created after finishing the wizard.

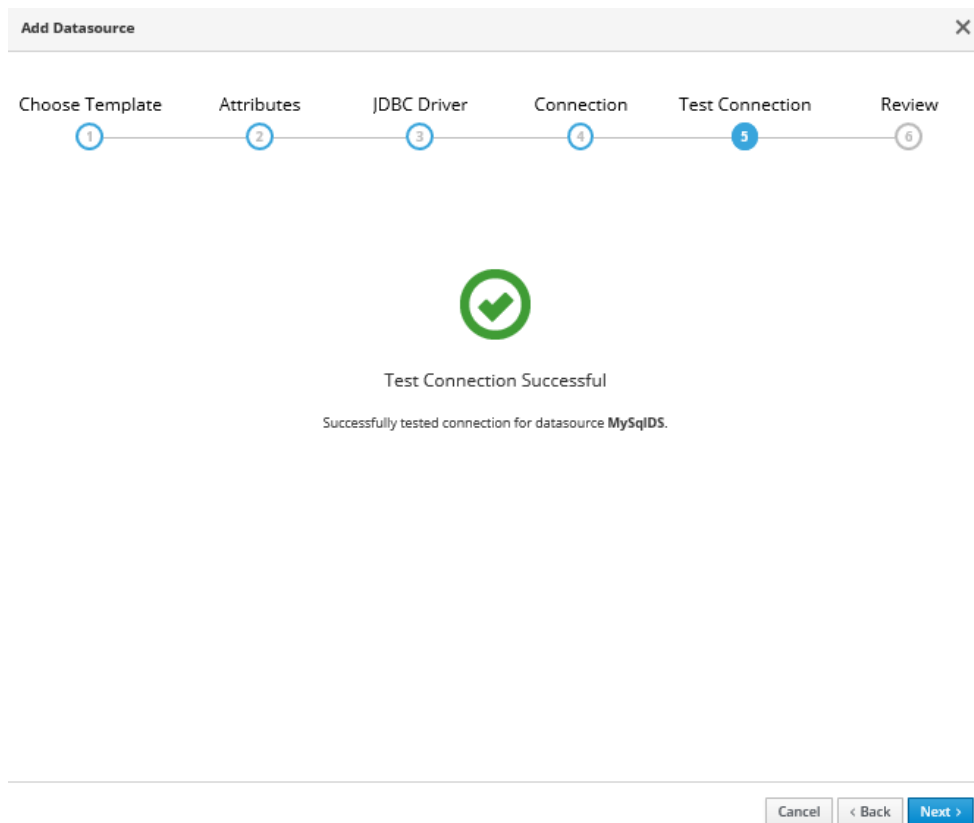
Test Connection

Cancel

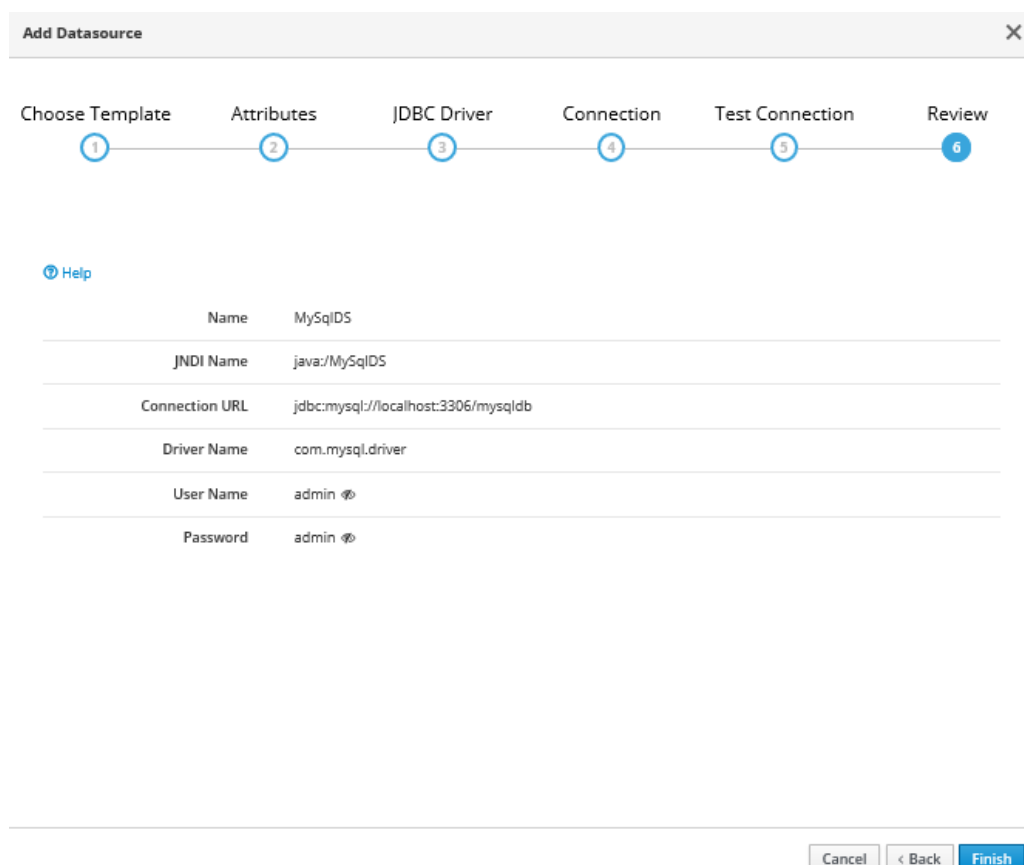
< Back

Next >

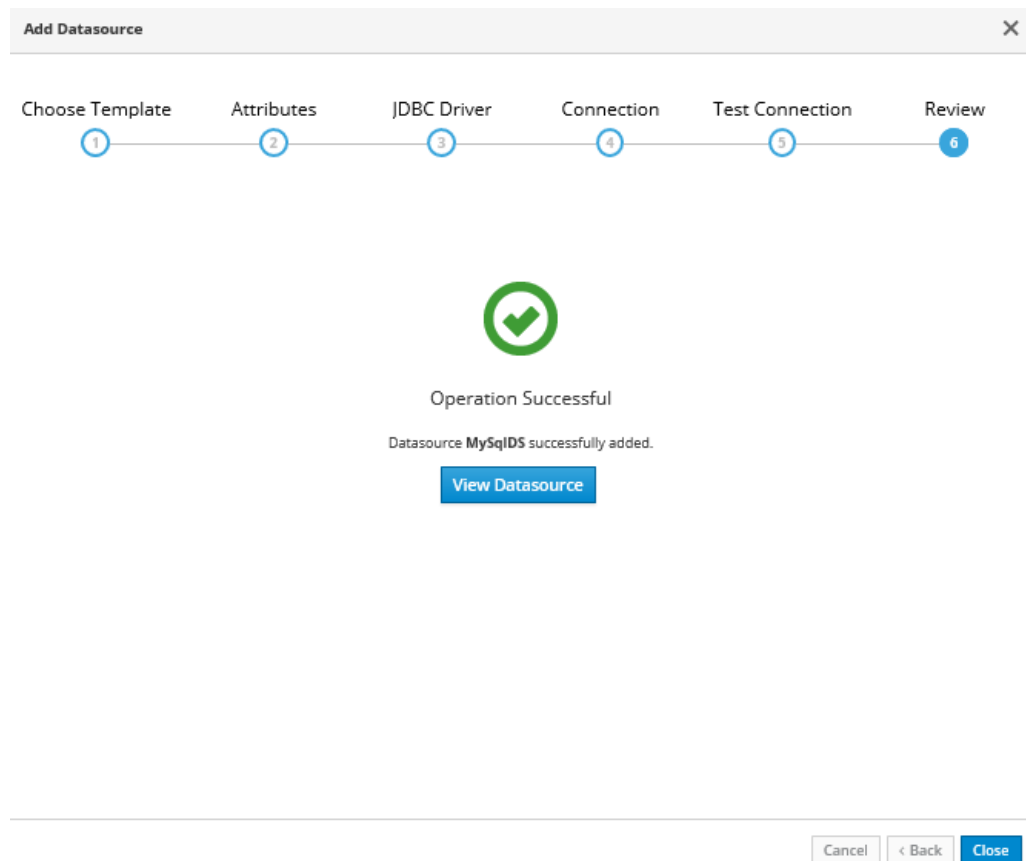
f. Si tout va bien, vous recevez un message qui indique le succès de la connexion de la source de données «**MySqlDS**» à la base «**MySqlDB**» :



g. Appuyer sur «**Next**» pour afficher les propriétés de la source de données «**MySqlDS**» puis appuyer sur «**Finish**» pour terminer :



h. Appuyer «**View Datasource**» sur pour visualiser la source de données ainsi créée :



i. Voici la source de données ainsi créée :

Configuration	Subsystem (32)	Datasources & Drivers	Datasource
Subsystems	Filter by: name or subtitle	Datasources	Filter by: name, xa, .../disabled, c
Interfaces	Batch		ExampleDS
Socket Bindings	JBeret		MySQLDS View
Paths	Core Management	JDBC Drivers	
System Properties	Datasources & Drivers		
	Deployment Scanners		
	Discovery		
	Distributable Web		
	EE		
	FIR		

MySQLDS

Datasource

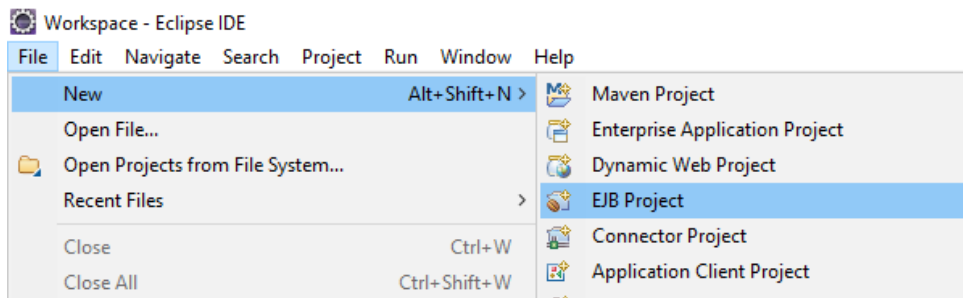
The datasource **MySQLDS** is enabled. [Disable](#)

Main Attributes

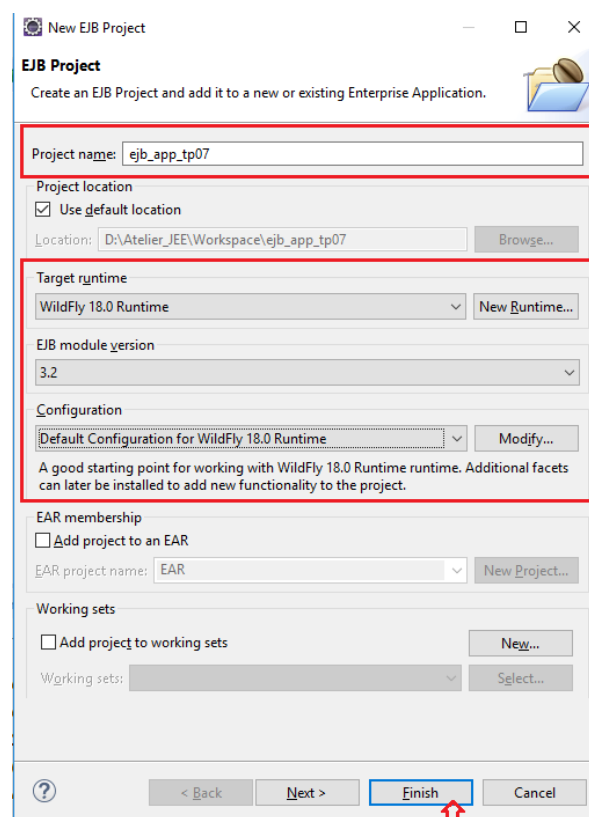
JNDI Name:	java:/MySQLDS
Driver Name:	com.mysql.driver
Connection URL:	jdbc:mysql://localhost:3306/mysqlpdb
Enabled:	true
Statistics Enabled:	false

C. Premier projet EJB

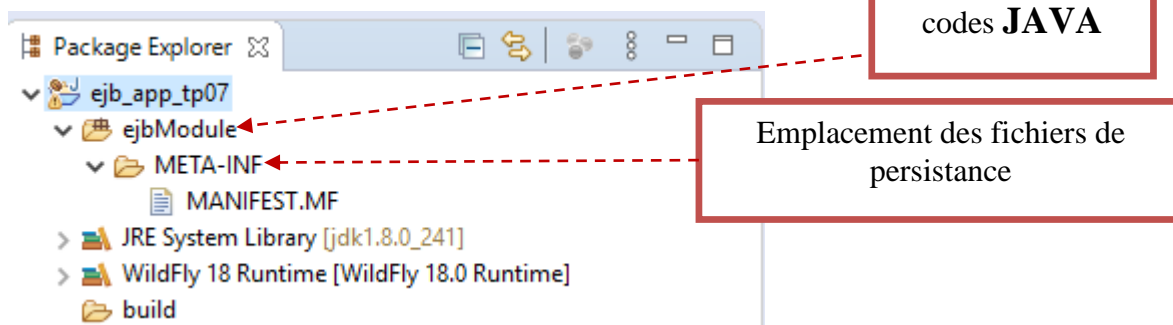
17. Dans **eclipse**, accéder au « **File/New/EJB Project** » :



18. Créer un nouveau projet EJB nommé «**ejb_app_tp07**» associé au serveur **WildFly 18** et appuyer sur « **Finish** » :

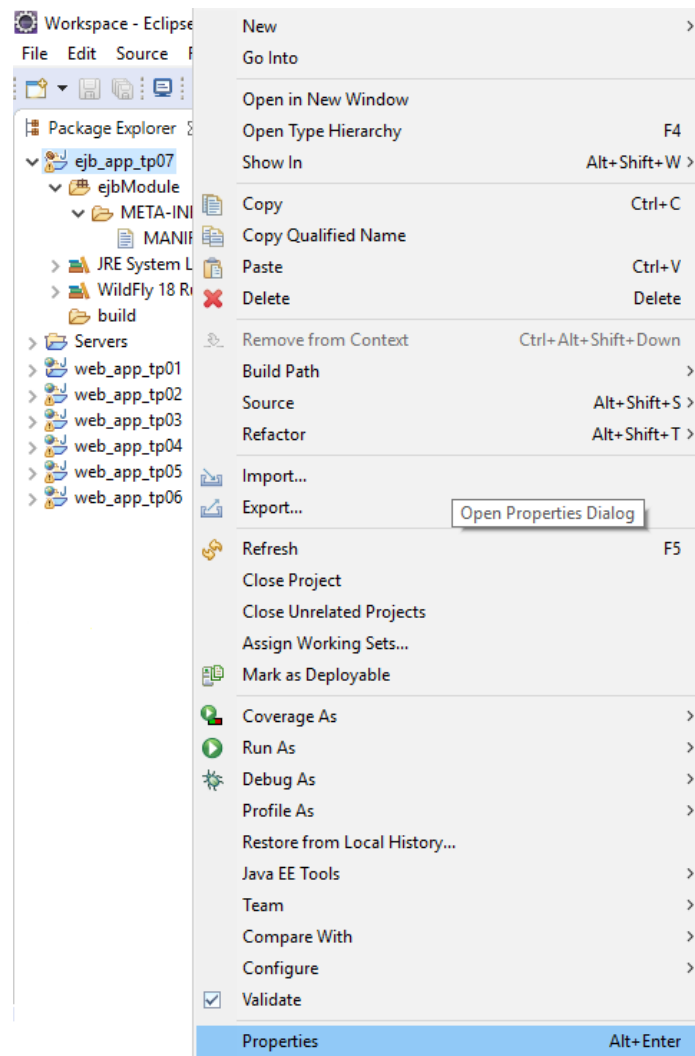


19. Voici la structure du projet ainsi créé :

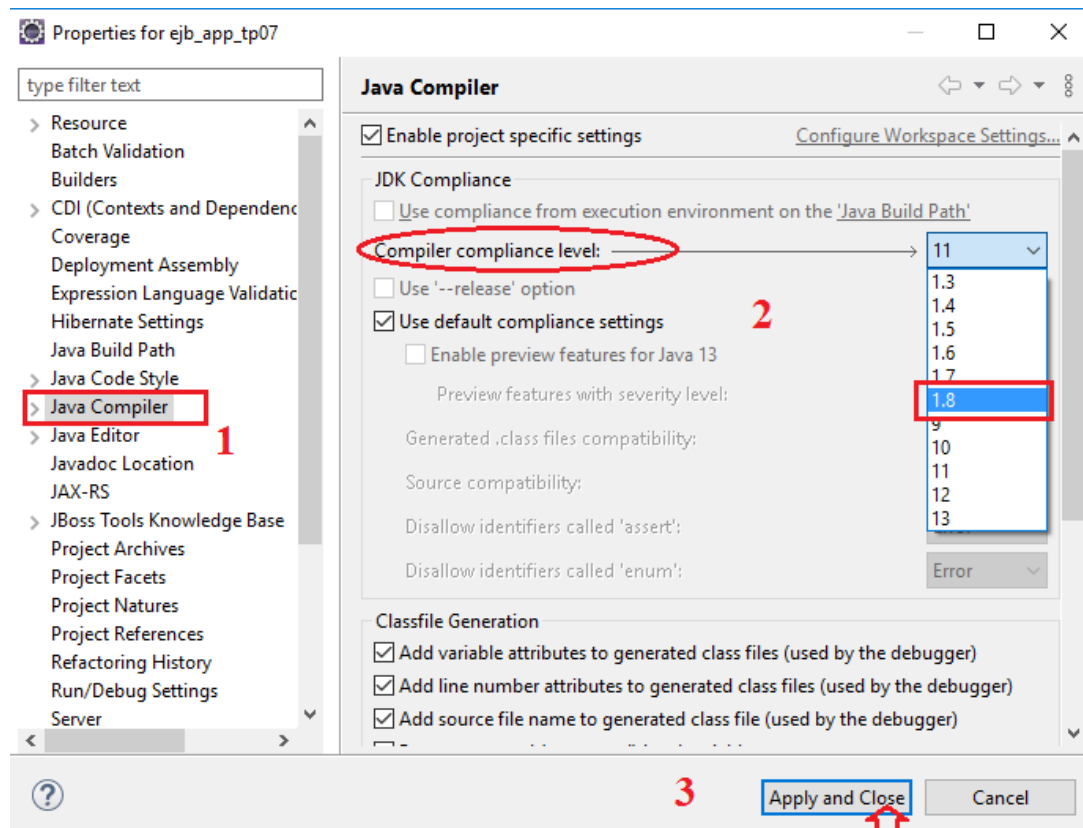


20. Rendre le niveau de conformité du compilateur à **1.8** :

a. Accéder aux propriétés du projet :

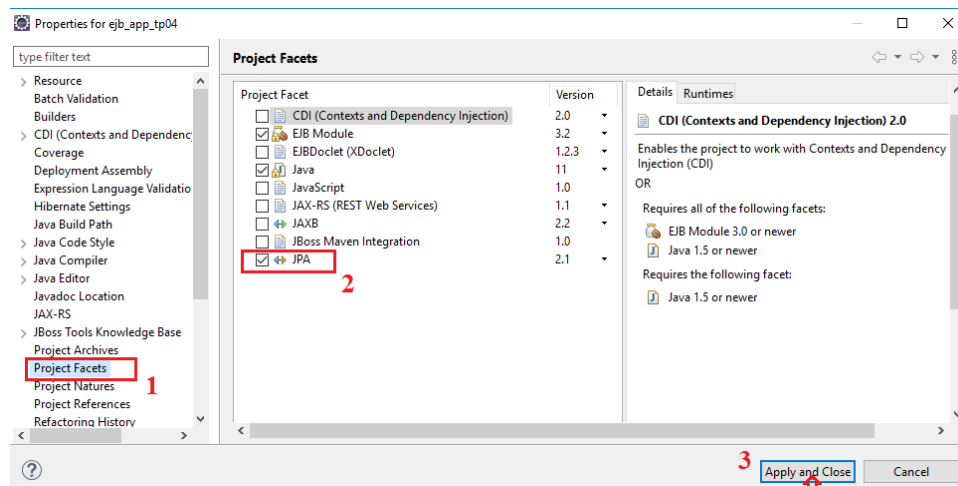


b. Rendre le niveau de conformité du compilateur à **1.8** :

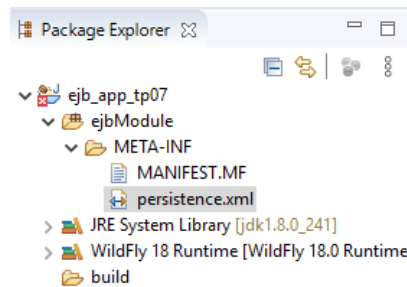


21. Ajouter les composants relatifs à JPA :

- a. Accéder aux propriétés du projet.
- b. Ajouter les éléments JPA (**Facet JPA**) au projet :



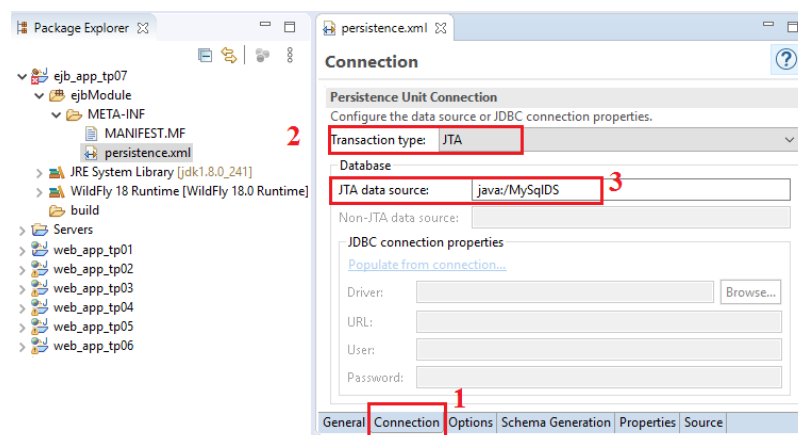
c. Remarquer l'ajout d'un fichier «**persistence.xml**» sous «**META-INF**» :

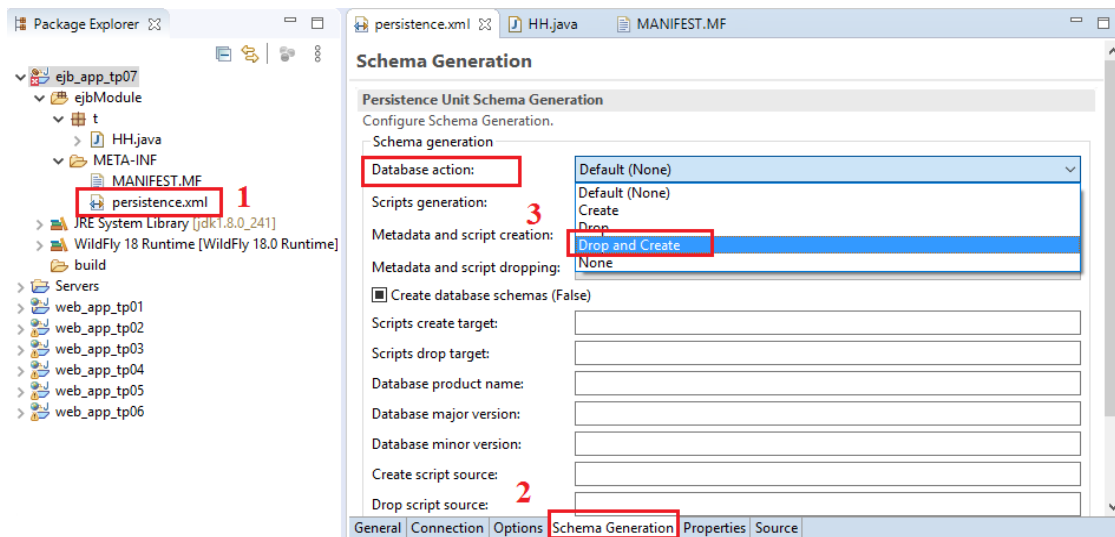


Ignorer la croix rouge signalée sur le non du projet due à la différence en la version du JDK installée et le niveau de conformité de la facette utilisée par le projet EJB.

22. Sélectionner le fichier de «**persistence.xml**» et passer au mode graphique pour spécifier :

- Le type de transaction : **JTA**
- Le nom JNDI de la source de données : **java:/MySqlDS**
- L'action à réaliser sur la base de données : **Drop and Create** (Supprimer le contenu de la BD et régénérer les tables à chaque exécution du projet)





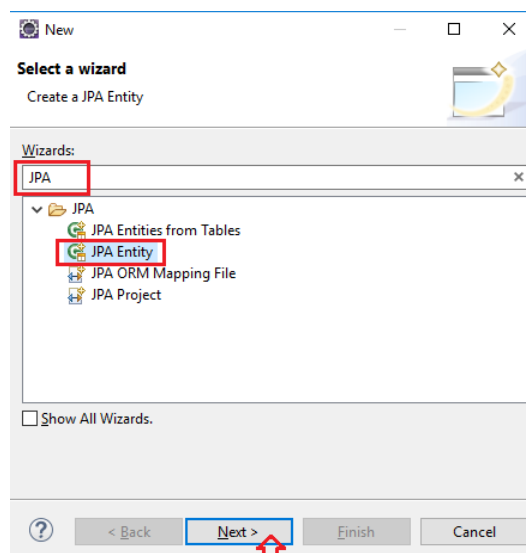
23. Voici ainsi l'aperçu du fichier « **persistence.xml** » :

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1"
  xmlns="http://xmlns.jcp.org/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
  <persistence-unit name="ejb_app_tp07"
    transaction-type="JTA">
    <jta-data-source>java:/MySqlDS</jta-data-source>
    <properties>
      <property name="javax.persistence.schema-generation.database.action"
        value="drop-and-create"/>
    </properties>
  </persistence-unit>
</persistence>
```

D. Premier composant EJB Entity (entité JPA)

24. Après avoir préparé la structure du projet, passer maintenant à implémenter une entité JPA :

- Créer, sous « **ejbModule** » du projet, un package « **domaine** ».
- Sélectionner le package « **domaine** » puis choisir la commande « **New/Other** » et chercher l'assistant « **JPA Entity** » :



c. Créer une entité JPA nommée «**Produit**» dans le package «**domaine**» :

New JPA Entity

Entity class

Create a new JPA entity. Only JPA enabled projects may be selected.

Project: ejb_app_tp07

Source folder: /ejb_app_tp07/ejbModule

Java package: domaine

Class name: Produit

Superclass:

Inheritance

☒ Entity

☐ Mapped superclass

☐ Inheritance:

XML entity mappings

☐ Add to entity mappings in XML

Mapping file:

< Back Next > Finish Cancel

d. Déclarer deux attributs :

- id de type **Long** (clé primaire au niveau de la BD)
- **designation** de type **String**
- **prix** de type **double**
- **quantite** de type **int**

New JPA Entity

Entity Properties

Set entity name, table name, fields, and access type.

Entity name: Produit

Table name

☒ Use default

Table name

Entity fields

Type: Long

Name: id

Key

Access type

☒ Field

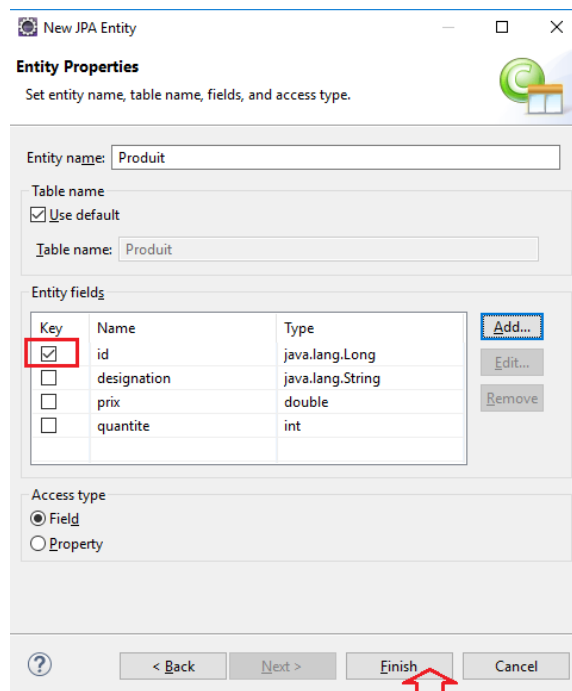
☐ Property

Add... Edit... Remove

OK Cancel

< Back Next > Finish Cancel

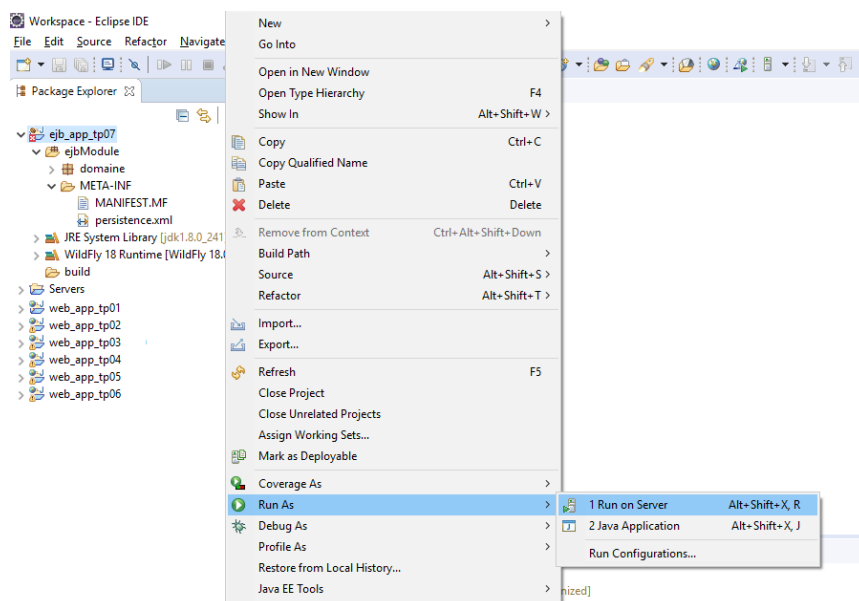
e. marquer l'attribut « **id** » comme clé et finaliser.



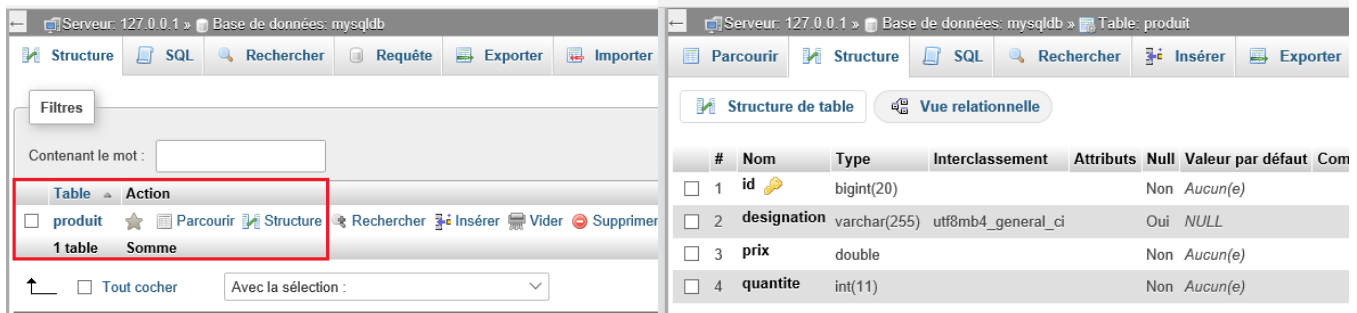
f. Remarquer la déclaration de l'entité JPA dans le fichier « **persistence.xml** » pour être pris en compte dans la procédure de persistance dans la base de données.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <persistence version="2.1"
3   xmlns="http://xmlns.jcp.org/xml/ns/persistence"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
6     http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
7   <persistence-unit name="ejb_app_tp07"
8     transaction-type="JTA">
9     <jta-data-source>java:/MySqlDS</jta-data-source>
10    <class>domaine.Produit</class>
11    <properties>
12      <property name="javax.persistence.schema-generation.database.action"
13        value="drop-and-create"/>
14    </properties>
15  </persistence-unit>
16 </persistence>
```

25. Passer, maintenant, à exécuter le projet avec le serveur **WildFly** :



26. Aller à la base de données pour visualiser la création de la table « produit » :



27. Personnaliser la structure de la table «**produit**» en utilisant des annotations dans le code de l'entité JPA «**Produit**» comme suit :

a. Rendre automatique l'incrémentation de la clé primaire :

```
@Id
@GeneratedValue (strategy = GenerationType.IDENTITY)
private Long id;
```

b. Limiter la taille de la colonne «**designation**» à 50 :

```
@Column (length=50)
private String designation;
```

c. Ajouter une colonne «**dateAchat**» de type **Date** :

```
@Temporal(TemporalType.DATE)
private java.util.Date dateAchat;
public java.util.Date getDateAchat() {
    return dateAchat;
}
public void setDateAchat(java.util.Date dateAchat) {
    this.dateAchat = dateAchat;
}
```

d. Enregistrer les modifications et relancer l'exécution du projet, puis visualiser la mise à jour de la structure de la table «**produit**» :

