

## Java EE

TP n°1 : Utilisation de l'api EJB 3.0  
Création d'un projet de type EJB.

## **Sommaire**

1. Objet .....	3
2. Pré requis.....	3
3.Préparation de l'environnement de développement .....	3
4. Développement de l'application EJB 3.0 .....	39
5. Tester l'application .....	59

## 1. Objet

- ✓ Installer MySQL (le serveur et l'interface graphique de l'administrateur).
- ✓ Créer un schéma.
- ✓ Installer WildFly.
- ✓ Créer un compte administrateur au niveau de WildFly.
- ✓ Créer un compte utilisateur au niveau de WildFly.
- ✓ Ajouter une nouvelle **datasource** (afin de se connecter à la base de données MySQL) au niveau de WildFly.
- ✓ Ajouter WildFly au niveau d'Eclipse Luna.
- ✓ Développer un projet EJB 3.0 (EJB Session StateLess et EJB Entity).
- ✓ Déployer le projet EJB dans WildFly.
- ✓ Développer un projet Java pour tester les services offerts par l'EJB Session.

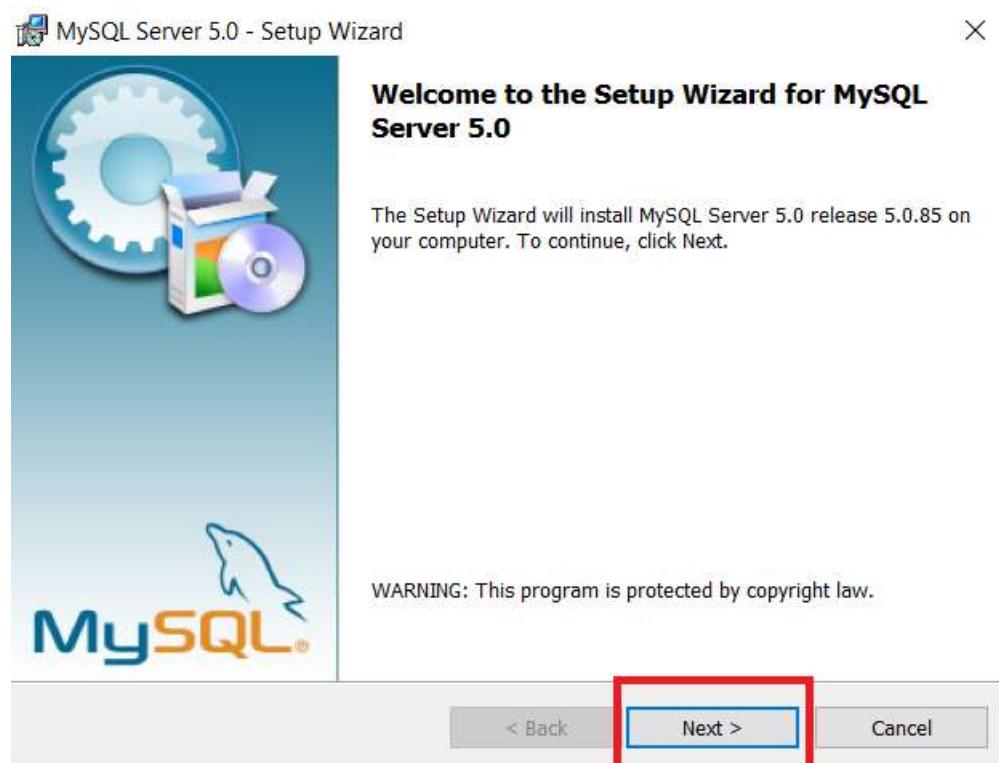
## 2. Pré requis

- ✓ JDK 1.7.
- ✓ Eclipse Luna.
- ✓ WildFly 8.2.0.
- ✓ MySQL 5.0 (driver mysql-connector-java-5.1.17-bin.jar).

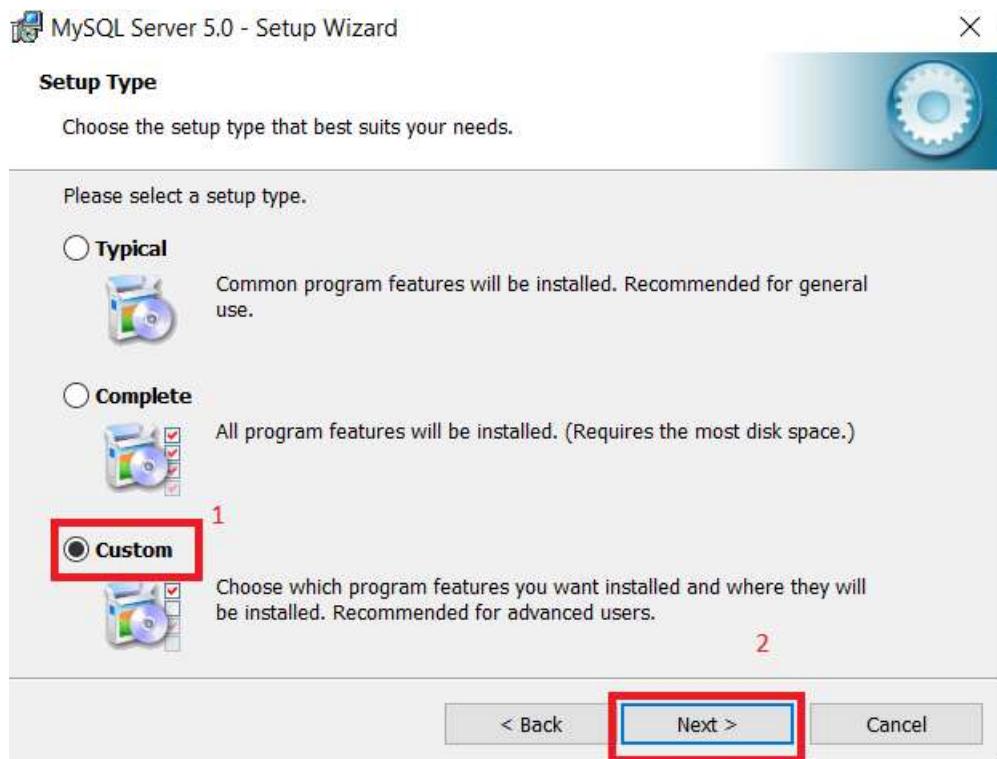
## 3. Préparation de l'environnement de développement

### Installation de la base de données MySQL 5.0:

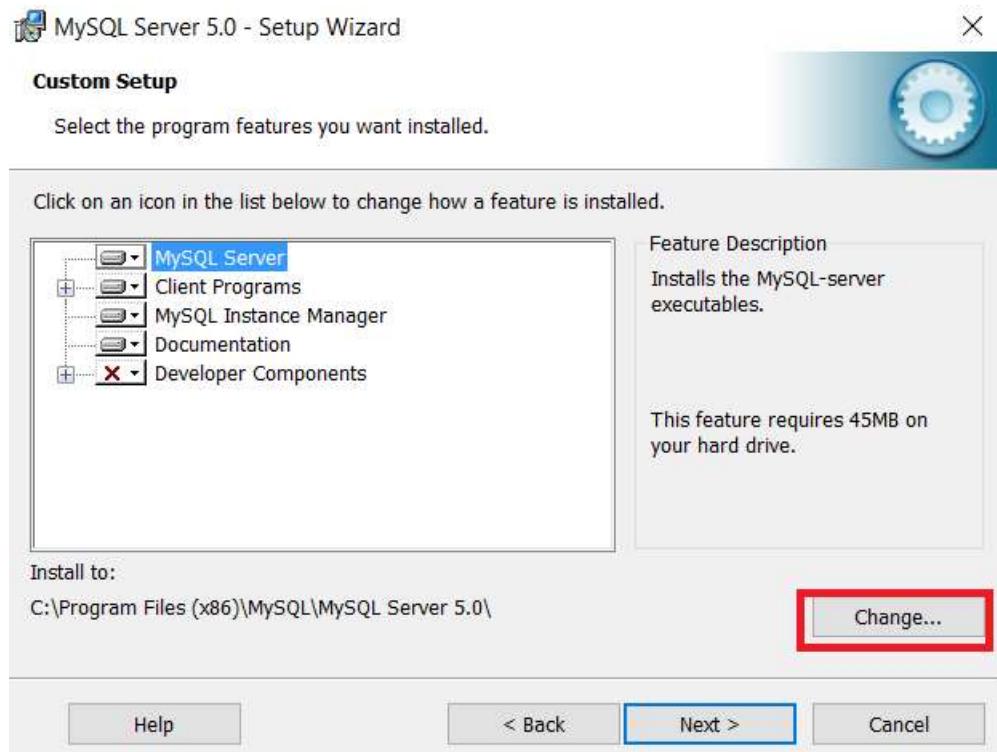
Double cliquer sur le fichier **setup.exe** et suivre les étapes suivantes :



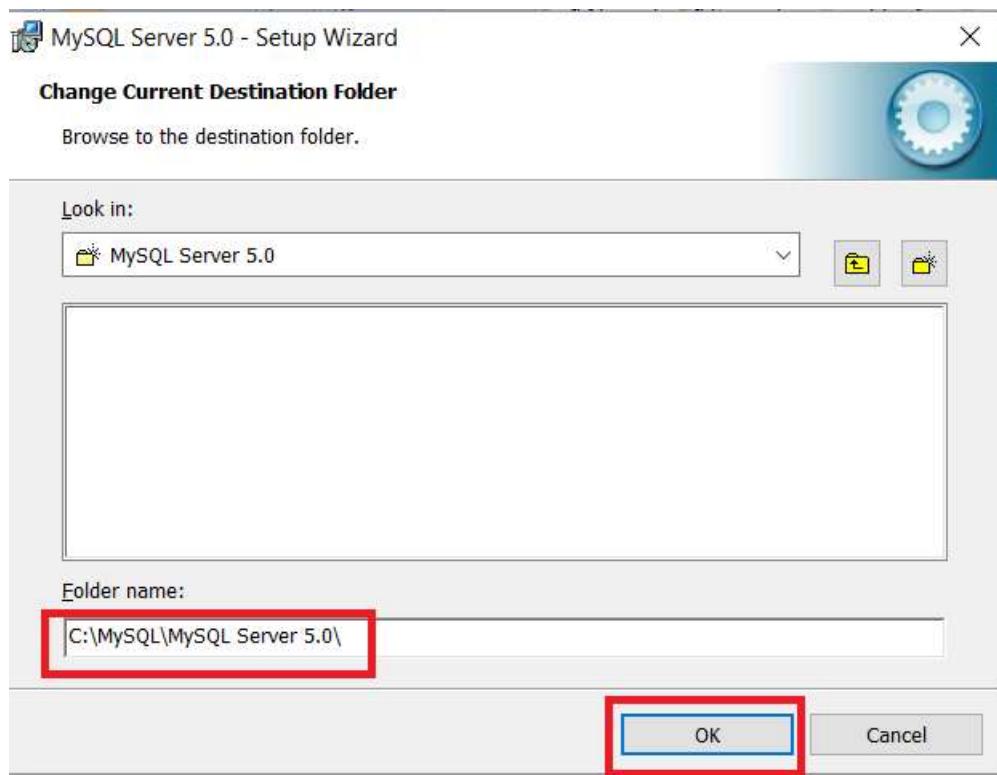
cliquer sur **Next >**:



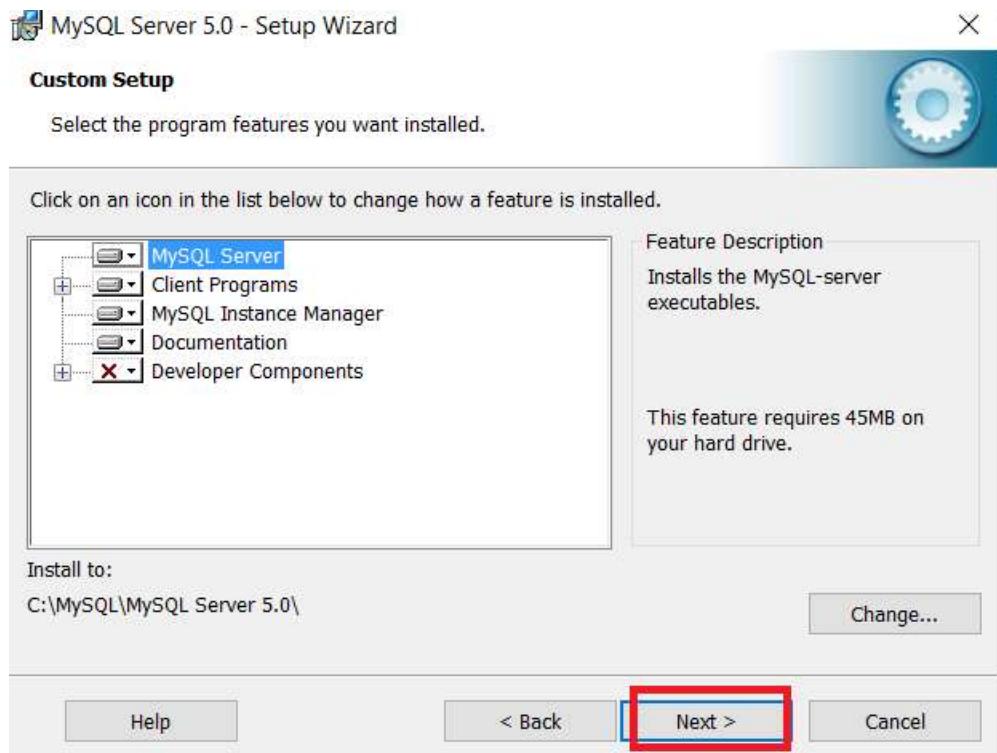
Cocher **Custom** et cliquer sur le bouton **Next >** :



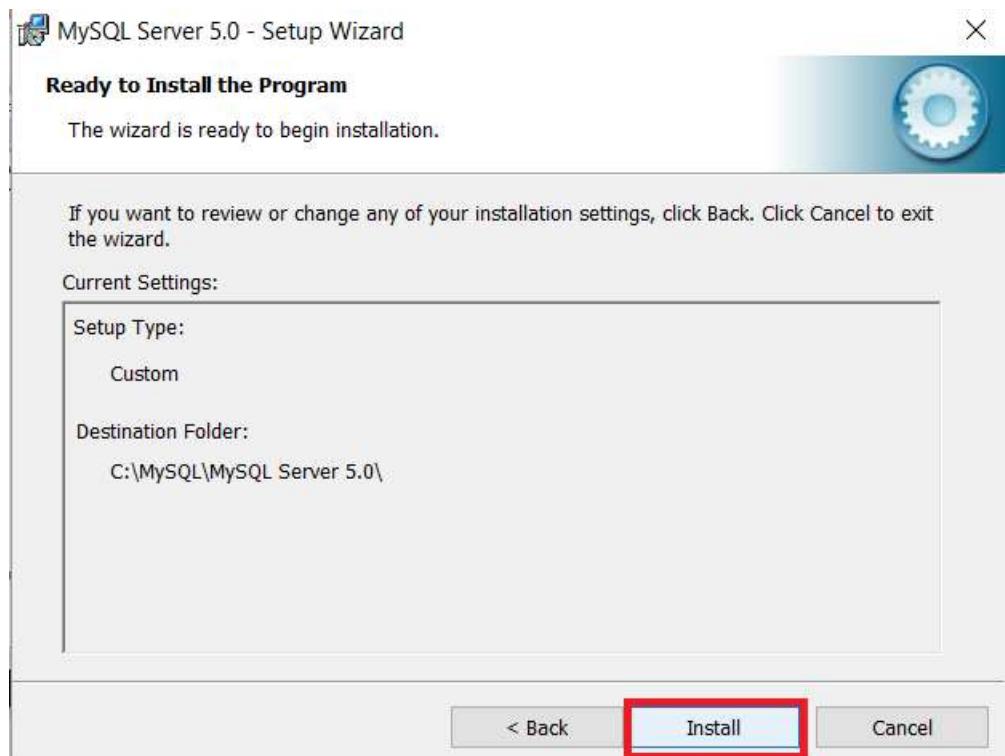
Cliquer sur **Change...** :



Entrer le dossier dans lequel sera installé MySQL (vous devriez avoir le droit en lecture et en écriture sur ce répertoire) et cliquer sur le bouton **OK** :



Cliquer sur le bouton **Next>** :



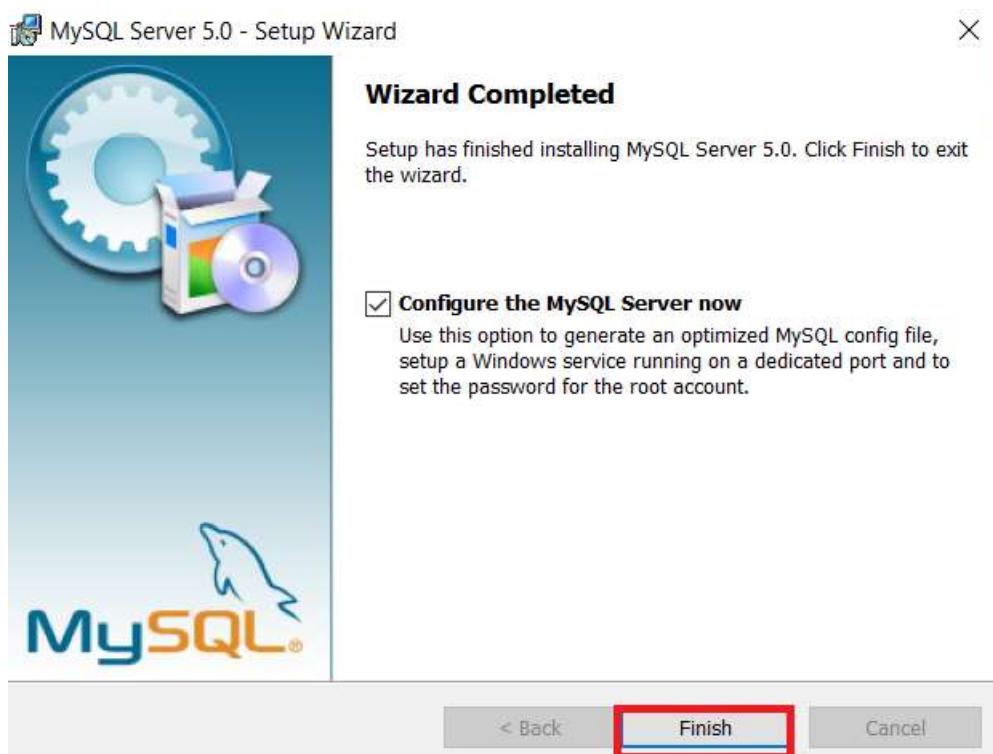
Cliquer sur **Install** :

Une fois l'écran suivant est affiché, cliquer sur le bouton **Next>** :

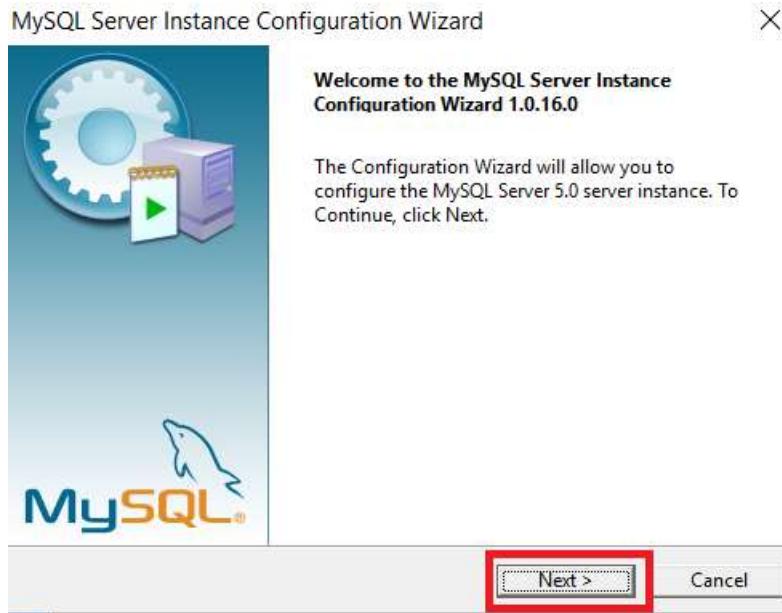




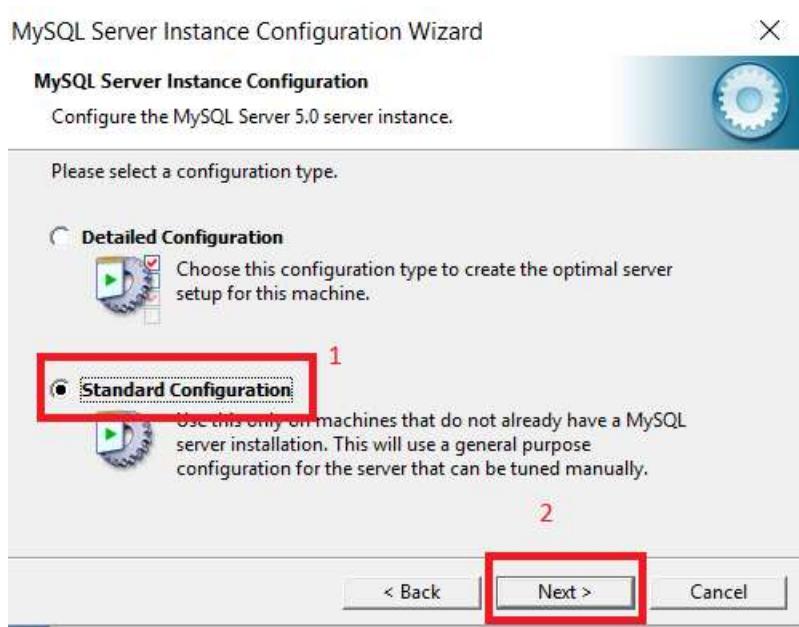
Cliquer sur **Next>** :



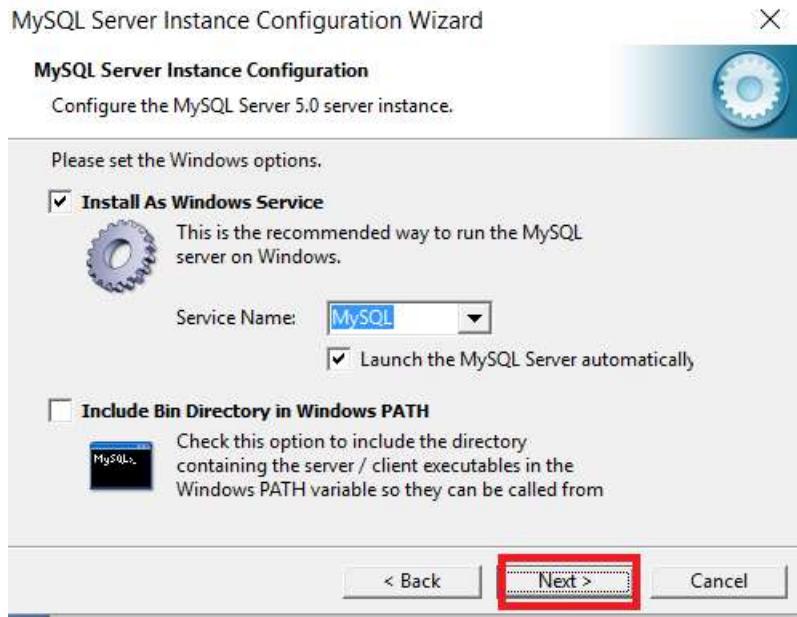
Cliquer sur **Finish** :



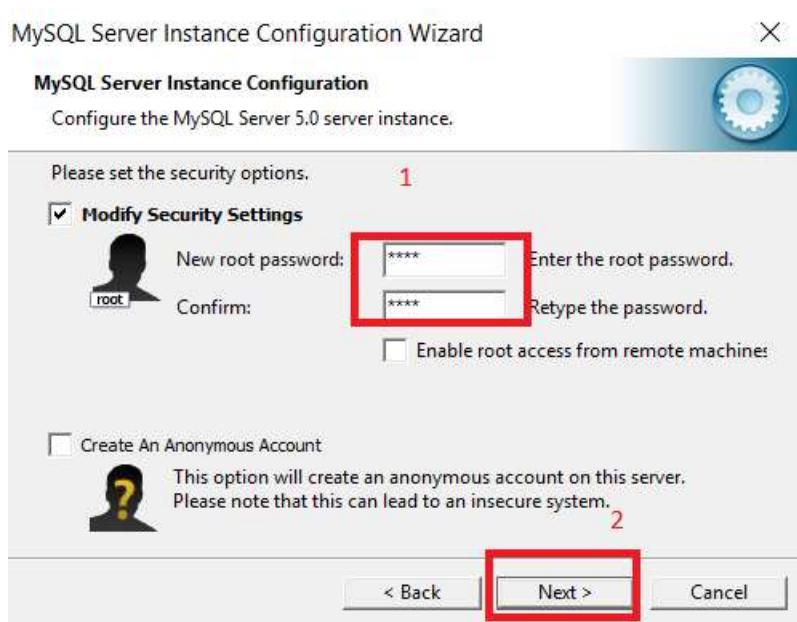
Cliquer sur le bouton **Next>** :



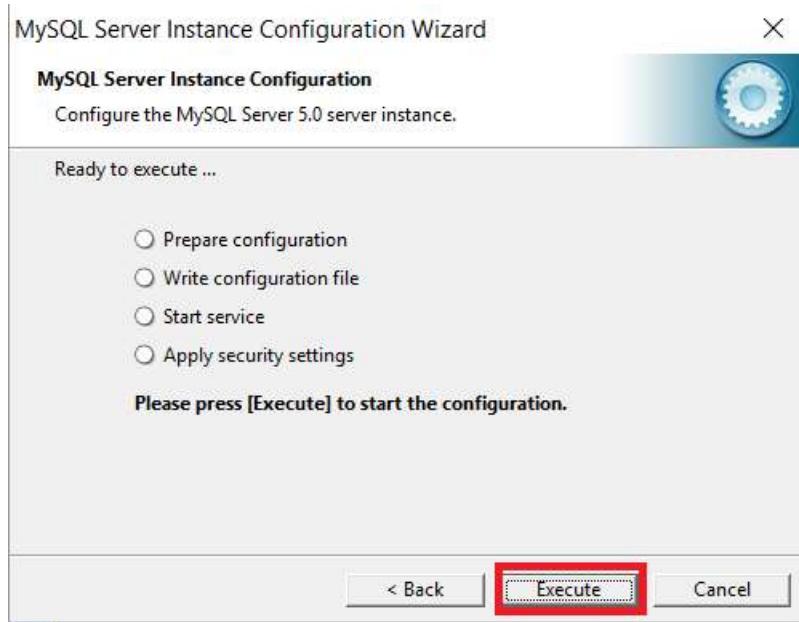
Cocher "**Standard Configuration**" et cliquer sur le bouton **Next>** :



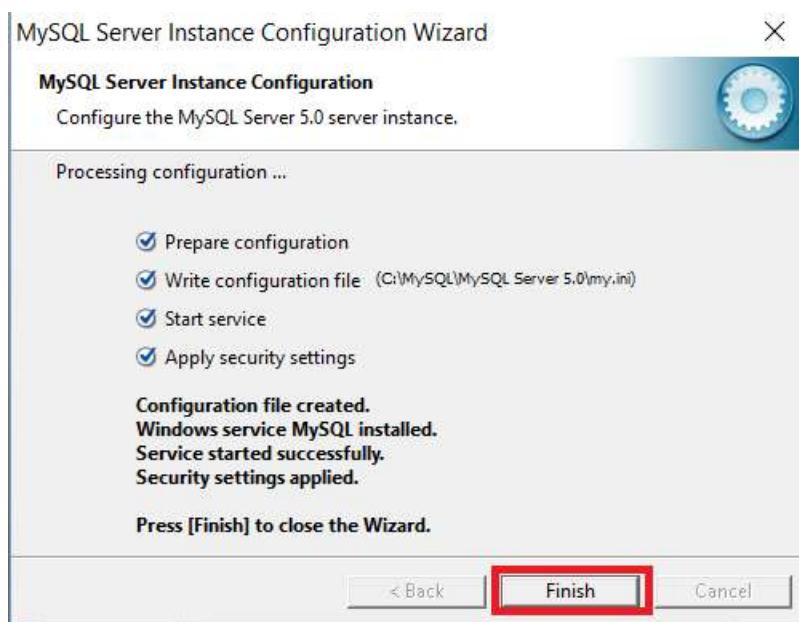
Cliquez sur le bouton **Next >** :



Entrer le mot de passe du compte **root** et cliquer sur **Next >** :



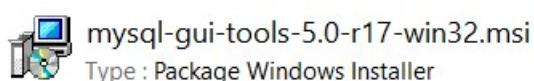
Cliquer sur le bouton **Execute** :

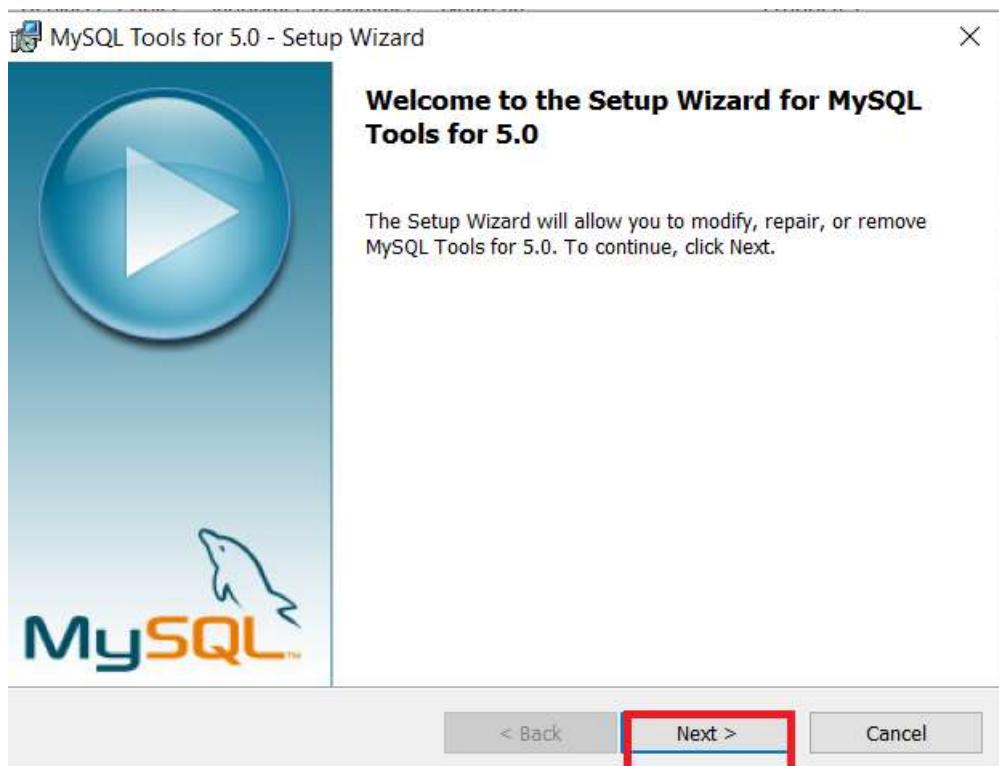


Une fois l'installation terminée, cliquer sur le bouton **Finish**.

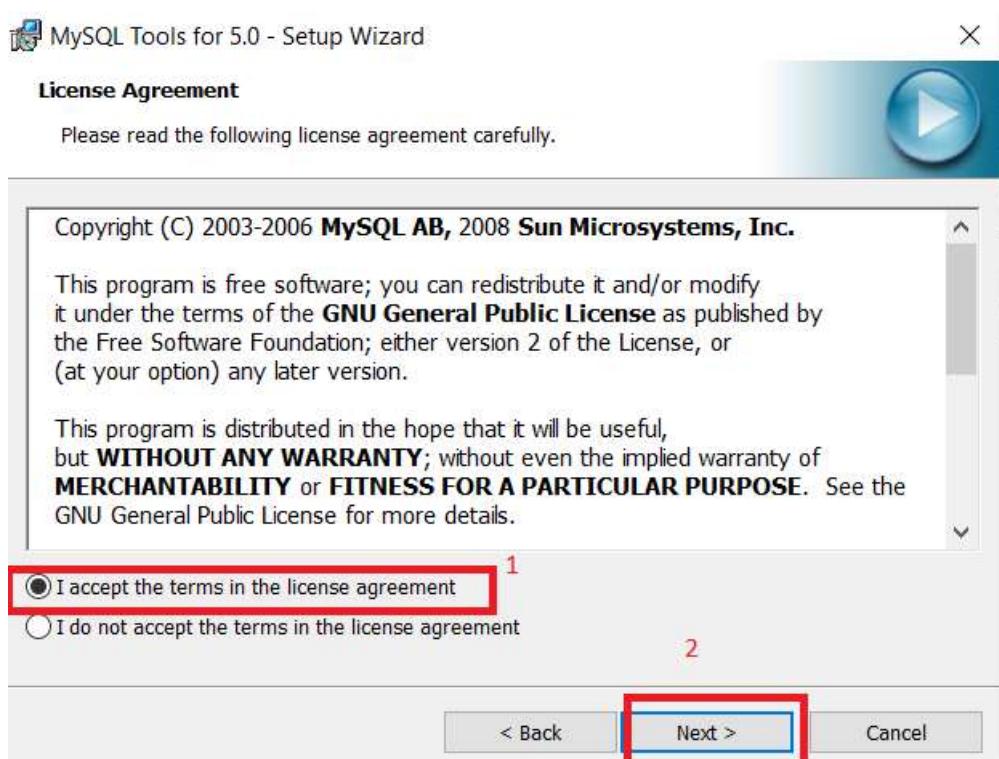
#### Installation de l'administrateur (interface client) de la base de données MySQL 5.0:

Cliquer sur l'exe suivant et suivre les étapes suivantes :

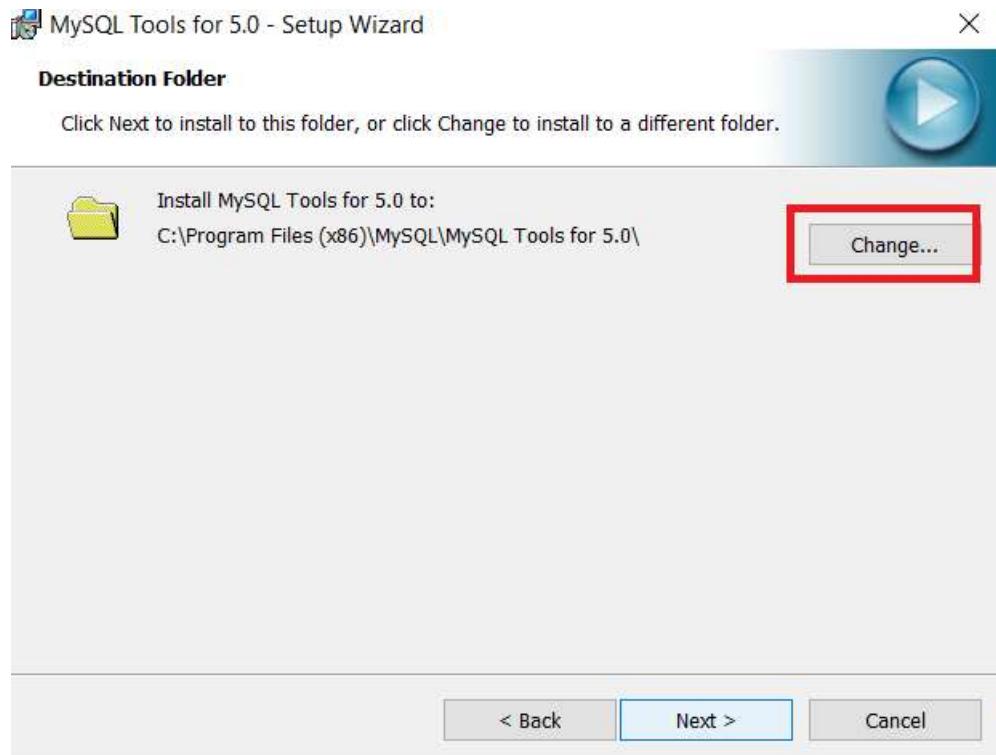




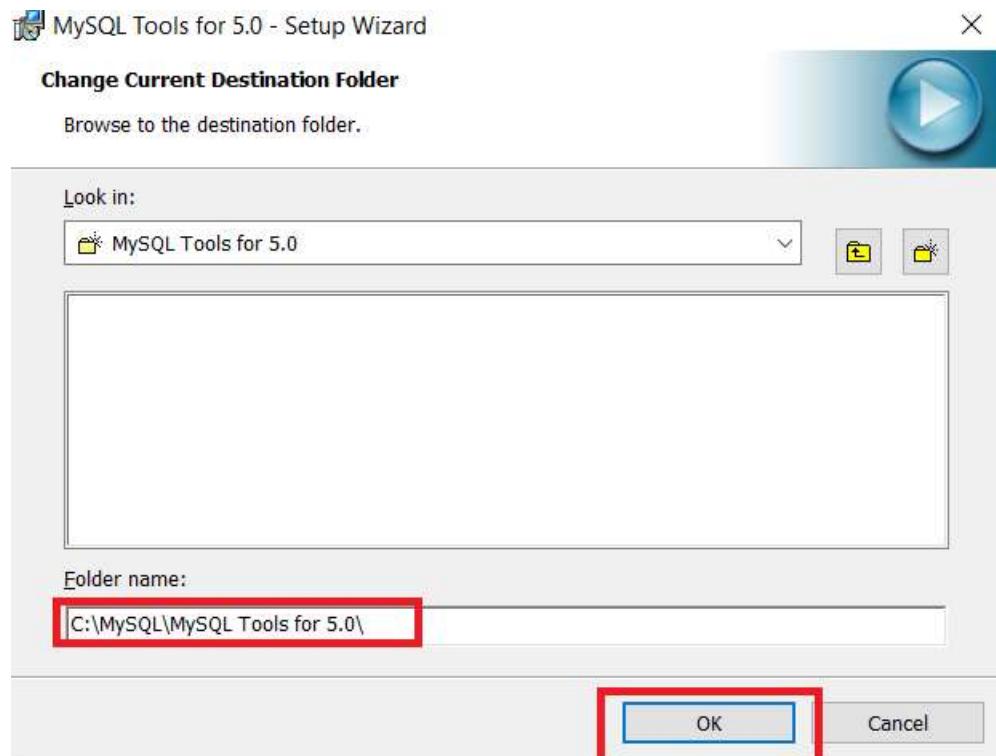
Cliquer sur le bouton **Next>** :



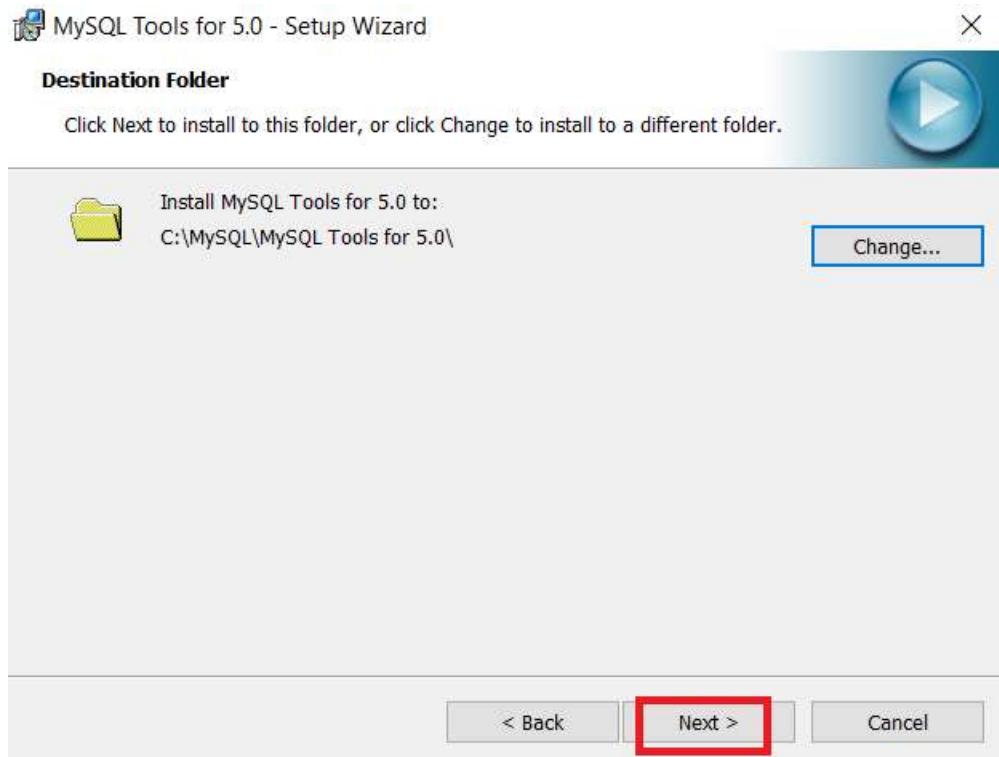
Cocher "**I accept ...**" et cliquer sur le bouton **Next>** :



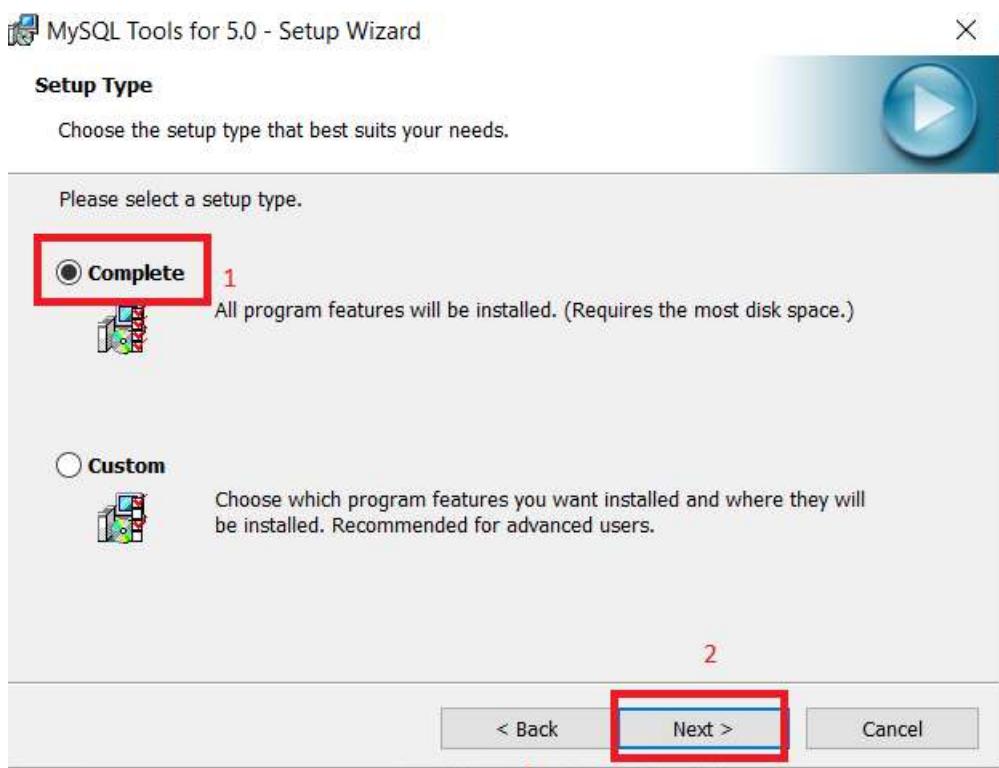
Cliquez sur le bouton **Change...** :



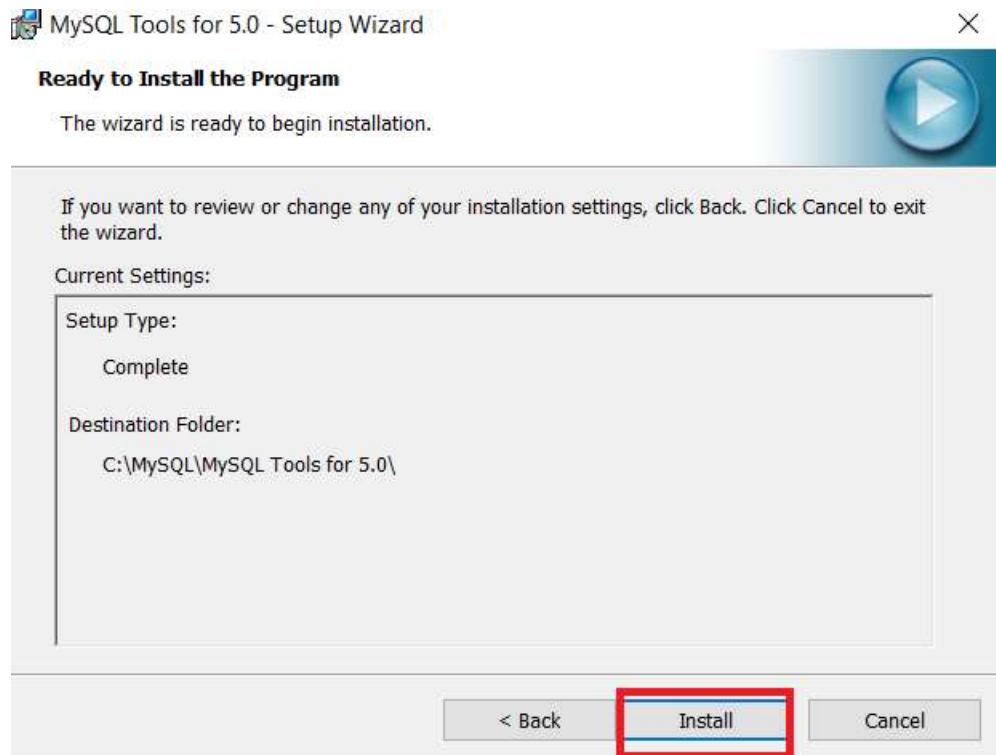
Entrer le chemin d'installation et cliquer sur le bouton **OK** :



Cliquer sur le bouton **Next >** :



Cocher la case à cocher **Complete** et cliquer sur le bouton **Next >** :



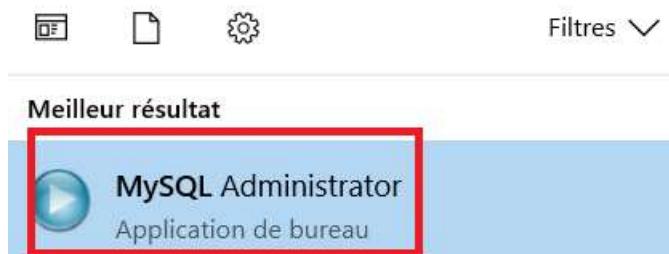
Cliquer sur le bouton **Install** :



Cliquer sur le bouton **Finish** :

Création d'un schéma :

Lancer "MySQL Administrator" :



#### Applications

MySQL Migration Toolkit

MySQL Query Browser

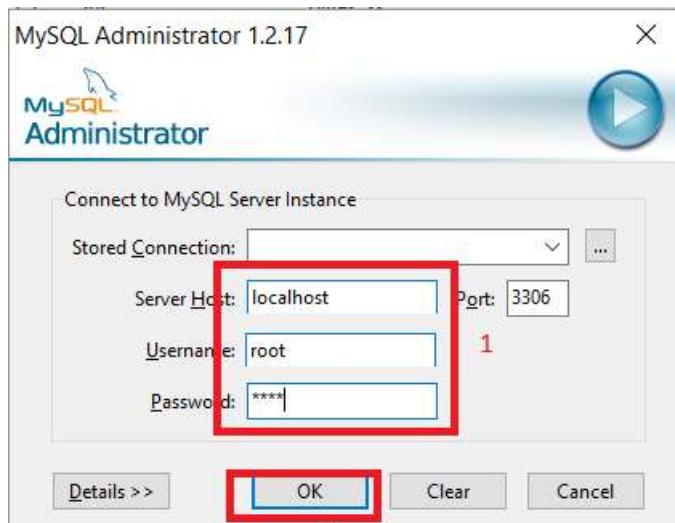
MySQL System Tray Monitor

#### Dossiers

MySQL 5.0 - dans tmp

MySQL 5.0 - dans tools

mysql-connector-java-5.1.17-bin.jar



entrer le login/password et cliquer sur le bouton **OK** :

## MySQL Administrator - Connection: root@localhost:3306

File Edit View Tools Window Help

-  Server Information
-  Service Control
-  Startup Variables
-  User Administration
-  Server Connections
-  Health
-  Server Logs
-  Replication Status
-  Backup
-  Restore
-  Catalogs

### Schemata

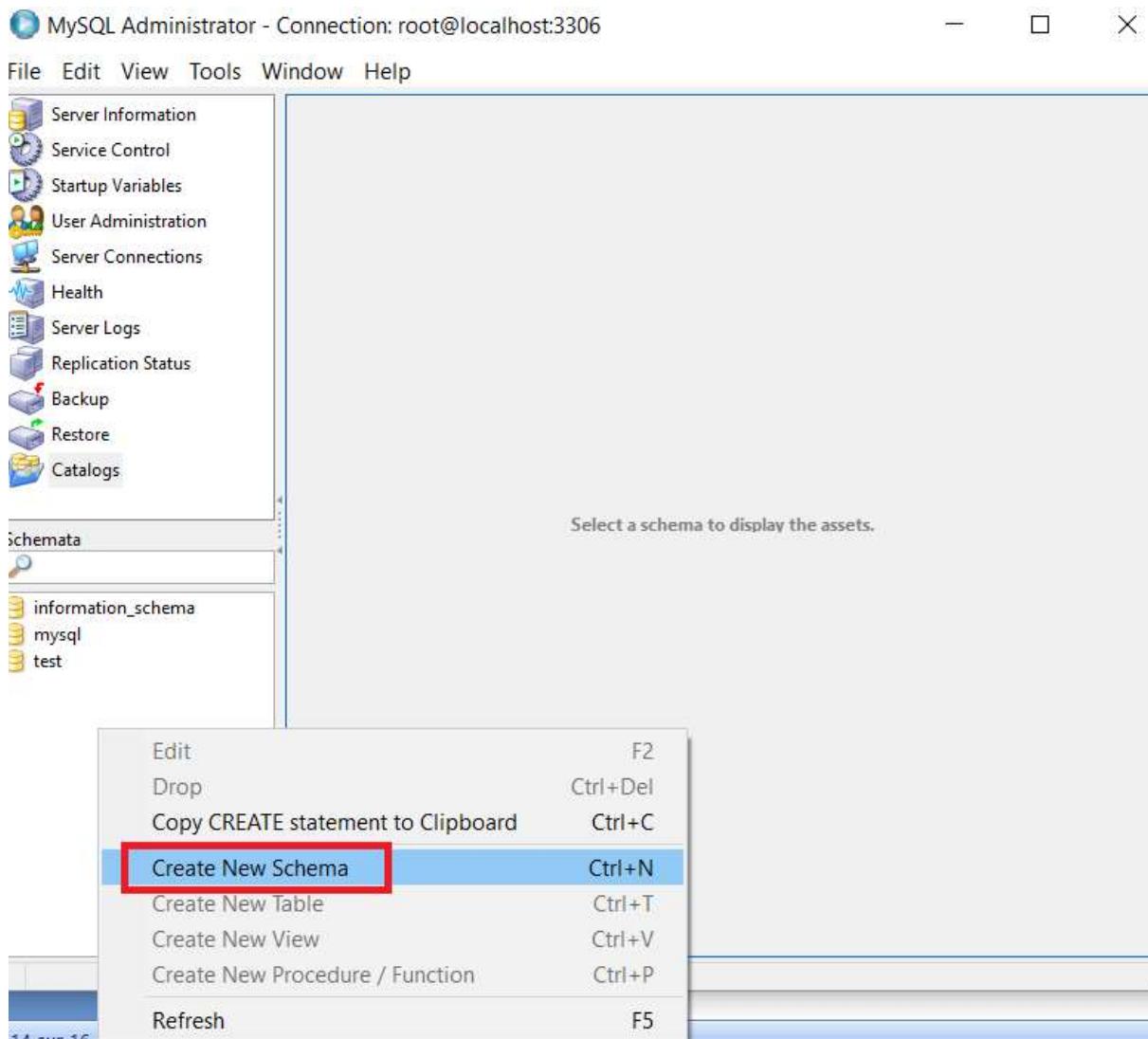
- 
-  information\_schema
-  mysql
-  test

Select a schema

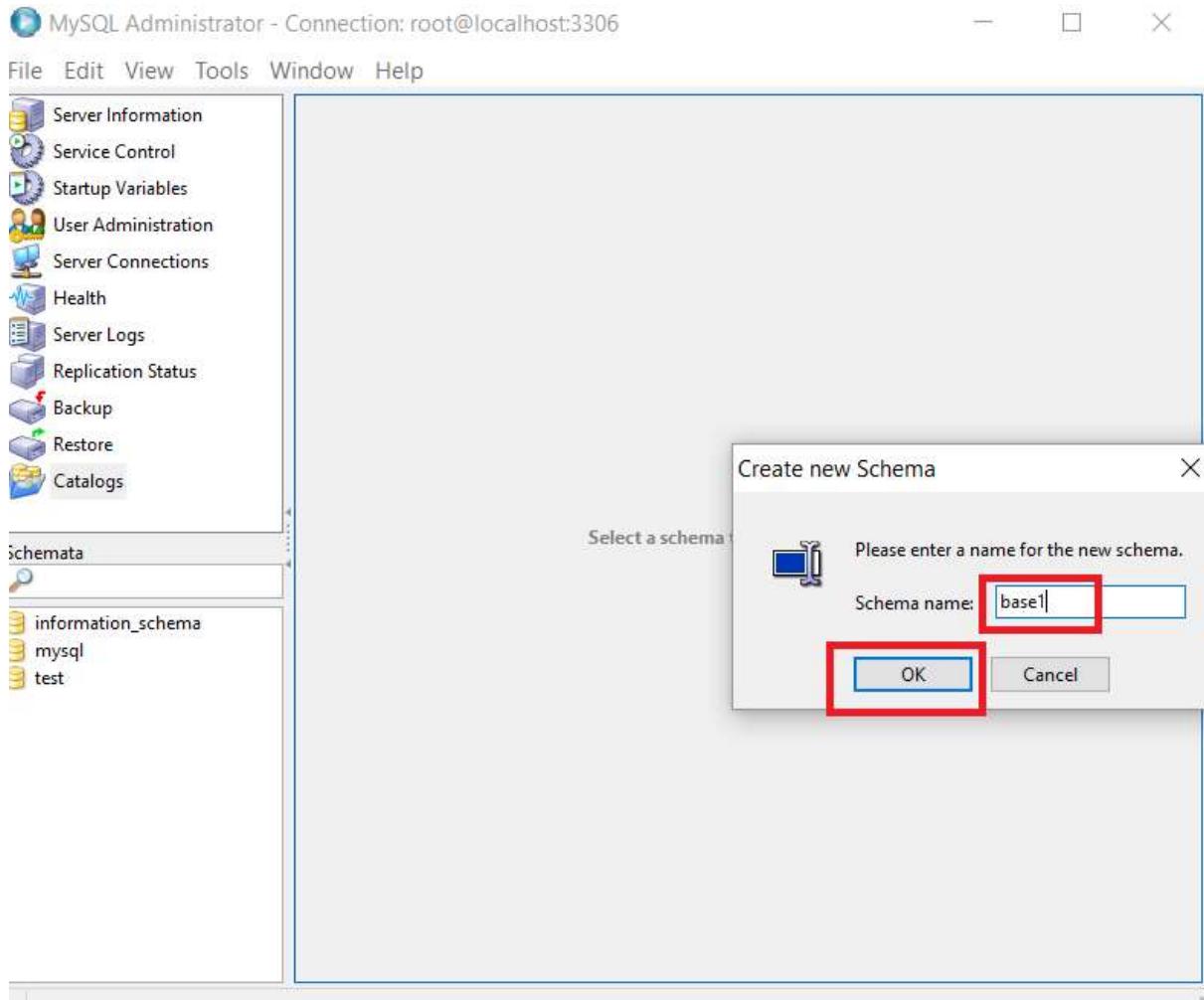


Cliquer à droite de  
la souris pour  
créer un nouveau  
schéma

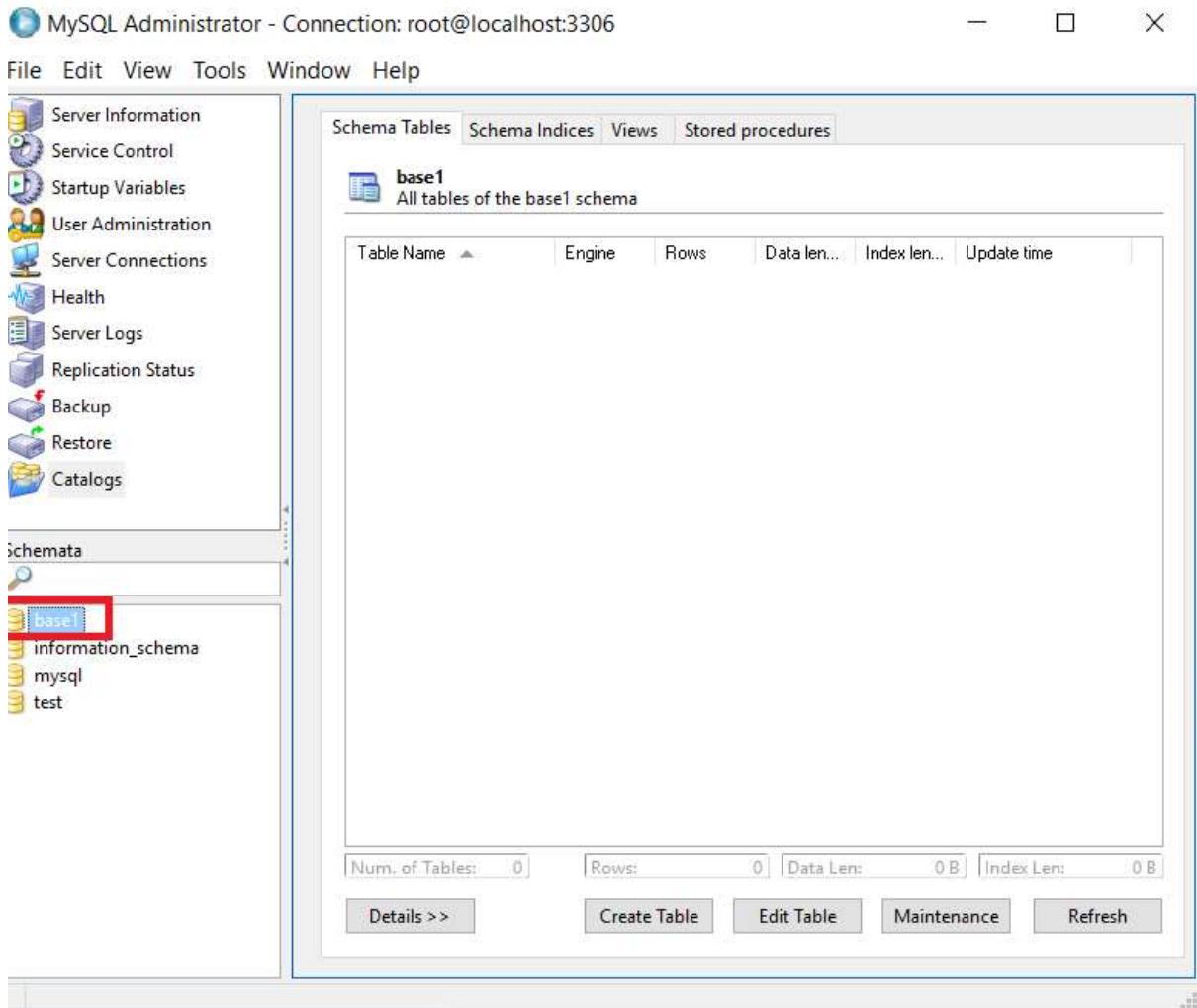
Cliquer sur **Catalogs** et cliquer ensuite à droit de la souris comme illustré ci-dessus :



Cliquez sur **Create New Schema** :



Entrer le nom de votre nouveau schéma et cliquer sur le bouton **OK**. le schéma sera finalement créé comme illustre l'écran suivant :



### **Installation et configuration de WildFly 8.2.0:**

Copier le dossier WildFly dans votre dossier (il faut avoir le droit en lecture et en écriture sur le répertoire choisi) :

Nom	Modifié le	Type	Taille
.installation	20/11/2014 22:43	Dossier de fichiers	
appclient	20/11/2014 22:43	Dossier de fichiers	
bin	20/11/2014 22:43	Dossier de fichiers	
docs	20/11/2014 22:43	Dossier de fichiers	
domain	20/11/2014 22:43	Dossier de fichiers	
modules	20/11/2014 22:43	Dossier de fichiers	
standalone	23/02/2017 20:04	Dossier de fichiers	
welcome-content	20/11/2014 22:43	Dossier de fichiers	
copyright.txt	20/11/2014 22:43	Document texte	3 Ko
jboss-modules.jar	20/11/2014 22:43	Executable Jar File	347 Ko
LICENSE.txt	20/11/2014 22:43	Document texte	26 Ko
README.txt	20/11/2014 22:43	Document texte	3 Ko

### **Ajouter un compte administrateur :**

Avant d'ajouter votre compte administrateur, il faut ajouter la variable d'environnement **JAVA\_HOME** comme illustre l'écran suivant :

## /variables d'environnement

X

Variables utilisateur pour toshiba satellite	
Variable	Valeur
OneDrive	C:\Users\toshiba satellite\OneDrive
Path	%USERPROFILE%\AppData\Local\Microsoft\WindowsApps;
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

Variables système	
Variable	Valeur
ComSpec	C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_CHECK	NO
JAVA_HOME	C:\Program Files\Java\jdk1.7.0_75
JRE_HOME	C:\Program Files\Java\jre/
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files\Java\jdk1.7.0_75\bin;C:\Java\jre 1.8.0_121\bi... COM_EXE_BAT_CMD_VBS_VDF_IS_ICF_WSC_WSH_MSC

Double cliquer le fichier **add-user.bat** (ce fichier se trouve dans le dossier [VOTRE\_DOSSIER\_INSTALLATION]/bin) et suivez les étapes suivantes :

Explorateur

Organiser

Nouveau

Ouvrir

PC > TI31320200A (C:) > serveurs > wildfly-8.2.0.Final > bin >

Nom	Modifié le	Type	Taille
client	20/11/2014 22:43	Dossier de fichiers	
init.d	20/11/2014 22:43	Dossier de fichiers	
service	20/11/2014 22:43	Dossier de fichiers	
jbossclirc	20/11/2014 22:43	Fichier JBOSSCLIRC	1 Ko
<b>add-user.bat</b>	23/02/2017 19:54	Fichier de comma...	2 Ko
add-user.properties	20/11/2014 22:43	Fichier PROPERTIES	3 Ko
add-user.sh	20/11/2014 22:43	Fichier SH	3 Ko
appclient.bat	20/11/2014 22:43	Fichier de comma...	3 Ko
appclient.conf	20/11/2014 22:43	Fichier CONF	2 Ko
appclient.conf.bat	20/11/2014 22:43	Fichier de comma...	3 Ko
appclient.sh	20/11/2014 22:43	Fichier SH	4 Ko
domain.bat	20/11/2014 22:43	Fichier de comma...	5 Ko
domain.conf	20/11/2014 22:43	Fichier CONF	3 Ko
domain.conf.bat	20/11/2014 22:43	Fichier de comma...	4 Ko
domain.sh	20/11/2014 22:43	Fichier SH	9 Ko
jboss-cli.bat	20/11/2014 22:43	Fichier de comma...	2 Ko

C:\WINDOWS\system32\cmd.exe

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

Saisir **a** et cliquer sur **Entrer** :

C:\WINDOWS\system32\cmd.exe

```
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 : administrateur
```

Saisir le nom de votre administrateur et cliquer sur **Entrer** :

```
C:\WINDOWS\system32\cmd.exe
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 : administrateur
Password recommendations are listed below. To modify these restrictions edit the add-user.properties configuration file.
- 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)
- The password should be different from the username
Password : [REDACTED]
```

Saisir votre Password et cliquer sur **Entrer** :

L'outil vous demandera de confirmer le mot de passe. Saisir le même mot de passe :

```
C:\WINDOWS\system32\cmd.exe
Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing property files.
Username : administrateur
Password recommendations are listed below. To modify these restrictions edit the add-user.properties configuration file.
- 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)
- The password should be different from the username
Password :
JBAS015266: Password must have at least 1 digit.
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)[  ]:
About to add user 'administrateur' for realm 'ManagementRealm'
Is this correct yes/no? yes
```

Saisir **yes** et cliquer sur **Entrer** :

```
C:\WINDOWS\system32\cmd.exe
Re-enter Password :
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 'administrateur' for realm 'ManagementRealm'
Is this correct yes/no? yes
Added user 'administrateur' to file 'C:\serveurs\wildfly-8.2.0.Final\standalone\configuration\mgmt-users.properties'
Added user 'administrateur' to file 'C:\serveurs\wildfly-8.2.0.Final\domain\configuration\mgmt-users.properties'
Added user 'administrateur' with groups to file 'C:\serveurs\wildfly-8.2.0.Final\standalone\configuration\mgmt-groups.properties'
Added user 'administrateur' with groups to file 'C:\serveurs\wildfly-8.2.0.Final\domain\configuration\mgmt-groups.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? yes
```

Saisir **yes** et cliquer sur **Entrer** :

```
C:\WINDOWS\system32\cmd.exe
About to add user 'administrateur' for realm 'ManagementRealm'
Is this correct yes/no? yes
Added user 'administrateur' to file 'C:\serveurs\wildfly-8.2.0.Final\standalone\configuration\mgmt-users.properties'
Added user 'administrateur' to file 'C:\serveurs\wildfly-8.2.0.Final\domain\configuration\mgmt-users.properties'
Added user 'administrateur' with groups to file 'C:\serveurs\wildfly-8.2.0.Final\standalone\configuration\mgmt-groups.properties'
Added user 'administrateur' with groups to file 'C:\serveurs\wildfly-8.2.0.Final\domain\configuration\mgmt-groups.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? yes
To represent the user add the following to the server-identities definition <secret value="YWRtaW5pc3RyYXRldXI=" />
Appuyez sur une touche pour continuer... ■
```

Cliquer ensuite sur **Entrer**.

Vérifier que votre administrateur a été bien ajouté au niveau du fichier ***mgmt-users.properties*** (ce fichier se trouve dans le répertoire [VOTRE\_DOSSIER\_INSTALLATION]\domain\configuration) comme illustre l'écran suivant :

```

mgt-users.properties
1# for new installations. Further authentication mechanism can be configured
2# as part of the <management /> in host.xml.
3#
4# Users can be added to this properties file at any time, updates after the server has started
5# will be automatically detected.
6#
7# By default the properties realm expects the entries to be in the format: -
8#
9# A utility script is provided which can be executed from the bin folder to add the users: -
10# - Linux
11#   bin/add-user.sh
12#
13# - Windows
14#   bin\add-user.bat
15#
16# On start-up the server will also automatically add a user $local - this user is specifically
17# for local tools running against this AS installation.
18#
19#
20# The following illustrates how an admin user could be defined, this
21# is for illustration only and does not correspond to a usable password.
22#
23 administrateur=697c6c375dc4a75c8780a1d167ff3439
24#
25#$REALM_NAME=ManagementRealm$ This line is used by the add-user utility to identify the realm name already used in this file.
26#
27

```

Vérifier également que le rôle de votre administrateur a été bien ajouté au niveau du fichier ***mgmt-groups.properties*** (ce fichier se trouve dans le répertoire

[VOTRE\_DOSSIER\_INSTALLATION]\domain\configuration) comme illustre l'écran suivant :

```

mgt-groups.properties
1#
2# Properties declaration of users groups for the realm 'ManagementRealm'.
3#
4# This is used for domain management, users groups membership information is used to assign the user
5# specific management roles.
6#
7# Users can be added to this properties file at any time, updates after the server has started
8# will be automatically detected.
9#
10# The format of this file is as follows: -
11#
12# A utility script is provided which can be executed from the bin folder to add the users: -
13# - Linux
14#   bin/add-user.sh
15#
16# - Windows
17#   bin\add-user.bat
18#
19# The following illustrates how an admin user could be defined.
20#
21 administrateur=ManagementRealm|

```

#### Ajouter un compte utilisateur :

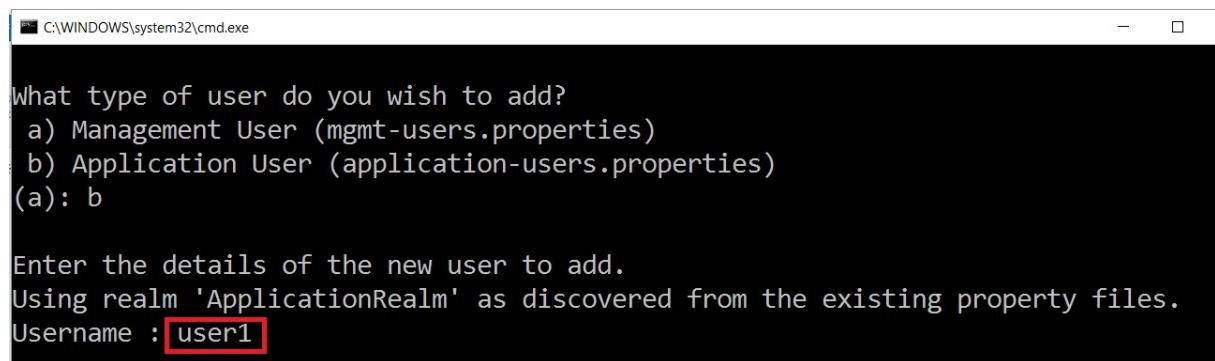
Pour ajouter un compte utilisateur, double cliquer sur le fichier ***add-user.bat*** (ce fichier se trouve dans le dossier [VOTRE\_DOSSIER\_INSTALLATION]/bin) et suivez les étapes suivantes :

What type of user do you wish to add?

- a) Management User (mgt-users.properties)
- b) Application User (application-users.properties)

(a): b-

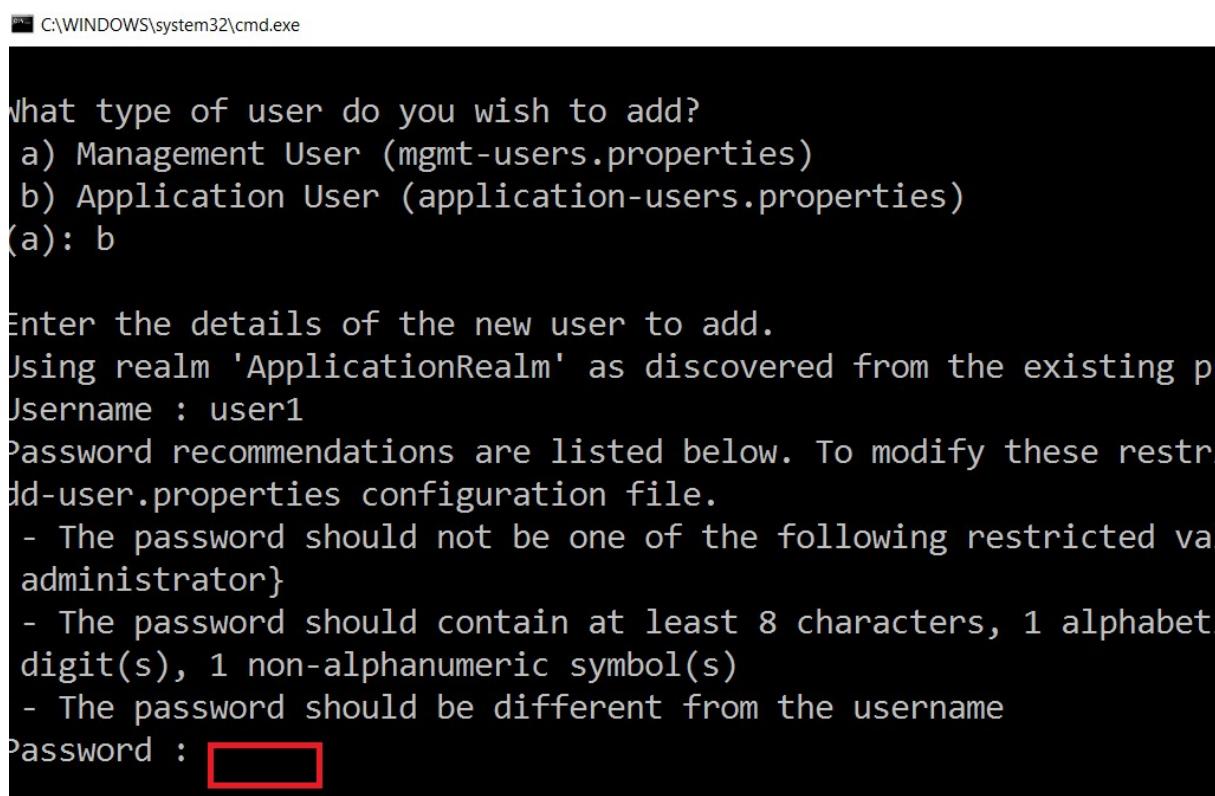
Saisir **b** et cliquer sur **Entrer** :



```
C:\WINDOWS\system32\cmd.exe
What type of user do you wish to add?
a) Management User (mgmt-users.properties)
b) Application User (application-users.properties)
(a): b

Enter the details of the new user to add.
Using realm 'ApplicationRealm' as discovered from the existing property files.
Username : user1
```

Entrer votre utilisateur et cliquer sur **Entrer** :



```
C:\WINDOWS\system32\cmd.exe
What type of user do you wish to add?
a) Management User (mgmt-users.properties)
b) Application User (application-users.properties)
(a): b

Enter the details of the new user to add.
Using realm 'ApplicationRealm' as discovered from the existing property files.
Username : user1
Password recommendations are listed below. To modify these restrictions, edit the application-users.properties configuration file.
- The password should not be one of the following restricted values: {  
    administrator}  
- The password should contain at least 8 characters, 1 alphabet character(s), 1 digit(s), 1 non-alphanumeric symbol(s)  
- The password should be different from the username
Password : [REDACTED]
```

Entrer le mot de passe et cliquer sur **Entrer** :

L'application demandera par la suite de confirmer le mot de passe.

```
C:\WINDOWS\system32\cmd.exe
Enter the details of the new user to add.
Using realm 'ApplicationRealm' as discovered from the existing property files.
Username : user1
Password recommendations are listed below. To modify these restrictions edit the a
dd-user.properties configuration file.
- 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)
- The password should be different from the username
Password :
JBAS015269: Password must have at least 8 characters!
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 li
st, or leave blank for none)[ ]:
About to add user 'user1' for realm 'ApplicationRealm'
Is this correct yes/no? yes
```

Entrer **yes** et cliquer sur **Entrer**.

```
C:\WINDOWS\system32\cmd.exe
About to add user 'user1' for realm 'ApplicationRealm'
Is this correct yes/no? yes
Added user 'user1' to file 'C:\serveurs\wildfly-8.2.0.Final\standalone\configurati
on\application-users.properties'
Added user 'user1' to file 'C:\serveurs\wildfly-8.2.0.Final\domain\configuration\ap
plication-users.properties'
Added user 'user1' with groups to file 'C:\serveurs\wildfly-8.2.0.Final\standalon
e\configuration\application-roles.properties'
Added user 'user1' with groups to file 'C:\serveurs\wildfly-8.2.0.Final\domain\co
nfiguration\application-roles.properties'
Is this new user going to be used for one AS process to connect to another AS pro
cess?
e.g. for a slave host controller connecting to the master or for a Remoting connec
tion for server to server EJB calls.
yes/no? yes
To represent the user add the following to the server-identities definition <secre
t value="dXNlcjE=" />
Appuyez sur une touche pour continuer...
```

Cliquer enfin sur **Entrer**.

Vérifier que votre utilisateur a été bien ajouté au niveau du fichier **application-users.properties** :

```

application-users.properties
1#
2# Properties declaration of users for the realm 'ApplicationRealm' which is the default realm
3# for application services on new installations.
4#
5# This includes the following protocols: remote ejb, remote jndi, web, remote jms
6#
7# Users can be added to this properties file at any time, updates after the server has started
8# will be automatically detected.
9#
10# The format of this realm is as follows: -
11#
12# A utility script is provided which can be executed from the bin folder to add the users: -
13# - Linux
14# bin/add-user.sh
15#
16# - Windows
17# bin\add-user.bat
18# The following illustrates how an admin user could be defined, this
19# is for illustration only and does not correspond to a usable password.
20#
21user1=6e388d2e45352e4133332118f40f3cec
22#
23#$REALM_NAME=ApplicationRealm$ This line is used by the add-user utility to identify the realm name already used in this file.
24#
25

```

Vérifier également que le fichier ***application-roles.properties*** a été mis à jour avec le rôle de votre utilisateur :

```

application-roles.properties
1#
2# Properties declaration of users roles for the realm 'ApplicationRealm' which is the default realm
3# for application services on new installations.
4#
5# This includes the following protocols: remote ejb, remote jndi, web, remote jms
6#
7# Users can be added to this properties file at any time, updates after the server has started
8# will be automatically detected.
9#
10# The format of this file is as follows: -
11#
12# A utility script is provided which can be executed from the bin folder to add the users: -
13# - Linux
14# bin/add-user.sh
15#
16# - Windows
17# bin\add-user.bat
18#
19# The following illustrates how an admin user could be defined.
20#
21user1=ApplicationRealm|

```

**Configurer la datasource pour accéder à la base de données MySQL (le schéma base1 créé ci-dessus) :**

#### **Etape 1 : Configuration du driver**

Créer l'arborescence suivante :[VOTRE\_DOSSIER\_INSTALLATION]\modules\com\mysql\main comme illustre l'écran suivant :

Ce PC > TI31320200A (C:) > serveurs > wildfly-8.2.0.Final > modules > com > mysql > main				
Nom	Modifié le	Type	Taille	
module.xml	19/02/2017 21:24	Document XML	1 Ko	
mysql-connector-java-5.1.17-bin.jar	28/09/2011 21:59	Executable Jar File	770 Ko	
mysql-connector-java-5.1.17-bin.jar.index	19/02/2017 21:25	Fichier INDEX	1 Ko	

Copier dans ce dossier le driver de MySQL (mysql-connector-java-5.1.17-bin.jar).

Créer le fichier **module.xml**. Le contenu de ce fichier est :

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.0" name="com.mysql">

    <resources>
        <resource-root path="mysql-connector-java-5.1.17-bin.jar"/>
        <!-- Insert resources here -->
    </resources>
    <dependencies>
        <module name="javax.api"/>
    </dependencies>
</module>
```

## Etape 2 : Configuration de la DataSource :

Modifier le fichier **[VOTRE\_Dossier\_INSTALLATION]\standalone\configuration\standalone.xml** comme suit (voir les modifications en surbrillance):

```
<datasources>
    <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS"
enabled="true" use-java-context="true">
        <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-1;DB_CLOSE_ON_EXIT=FALSE</connection-url>
        <driver>h2</driver>
        <security>
            <user-name>sa</user-name>
            <password>sa</password>
        </security>
    </datasource>
    <datasource jta="true" jndi-name="java:/MySQLDS" pool-name="MySQLDS" enabled="true"
use-java-context="true" use-ccm="true">
        <connection-url>jdbc:mysql://localhost:3306/base1</connection-url>
        <driver>com.mysql</driver>
        <security>
            <user-name>root</user-name>
            <password>root</password>
        </security>
        <timeout>
            <idle-timeout-minutes>0</idle-timeout-minutes>
            <query-timeout>600</query-timeout>
        </timeout>
        <statement>
            <prepared-statement-cache-size>100</prepared-statement-cache-size>
            <share-prepared-statements>true</share-prepared-statements>
        </statement>
    </datasource>
    <drivers>
        <driver name="h2" module="com.h2database.h2">
            <xa-datasource-class>org.h2.jdbc.JdbcDataSource</xa-datasource-class>
        </driver>
    </drivers>
</datasources>
```

```

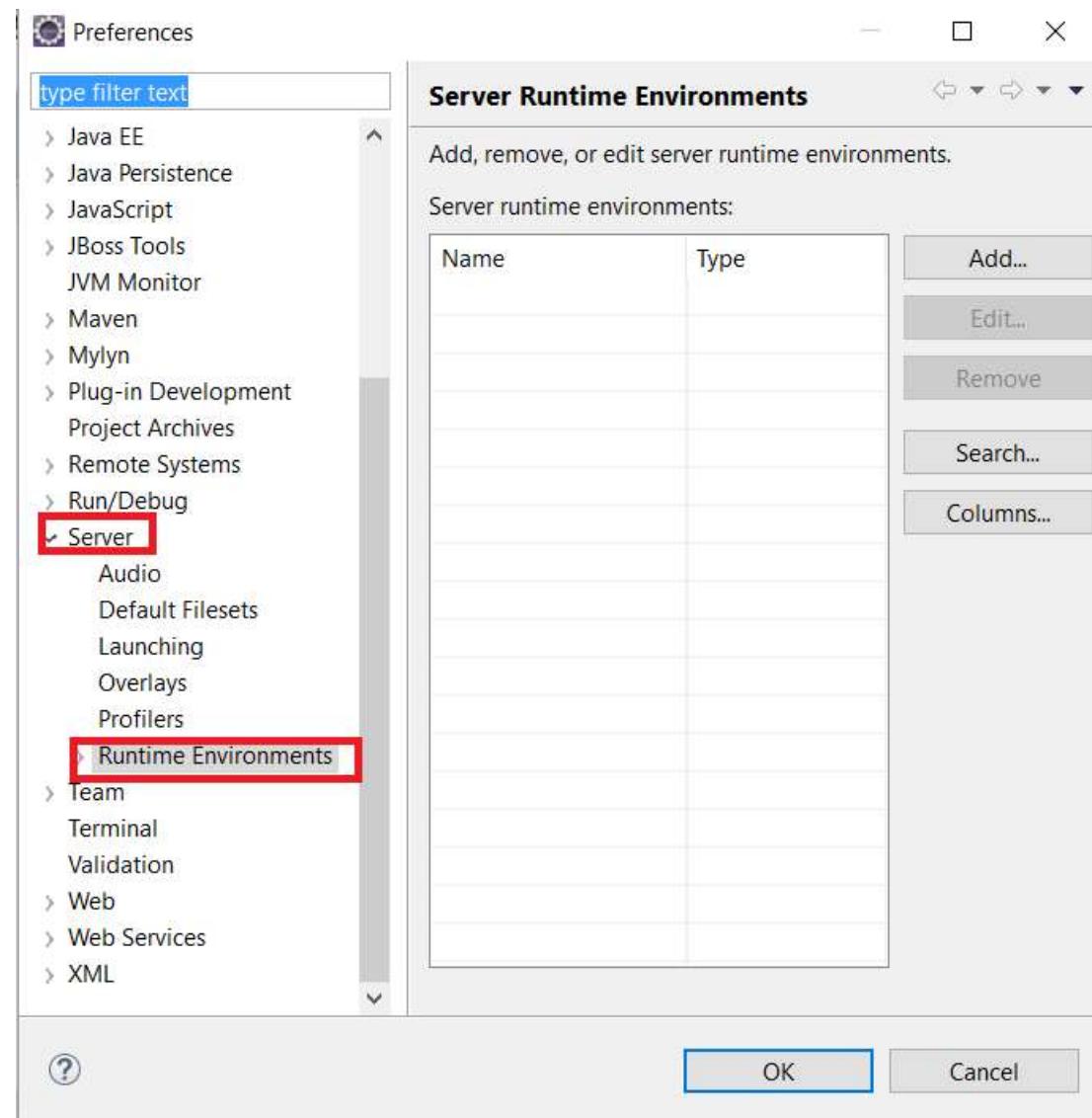
<driver name="com.mysql" module="com.mysql">
    <xa-datasource-class>com.mysql.jdbc.Driver</xa-datasource-class>
</driver>
</drivers>
</datasources>

```

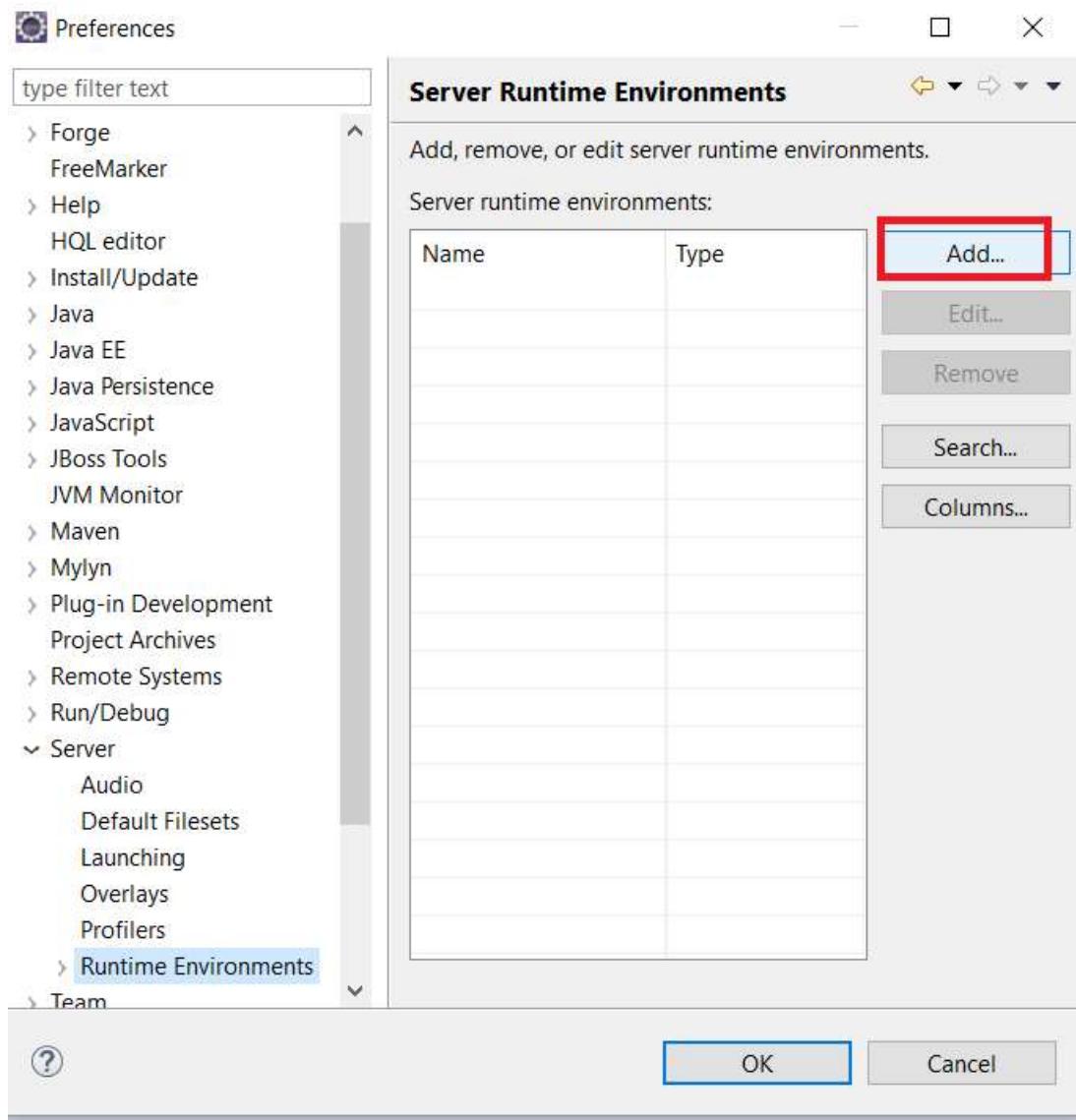
N'oubliez pas de modifier le nom et le mot de passe de votre utilisateur.

#### Configuration de WildFly au niveau d'Eclipse:

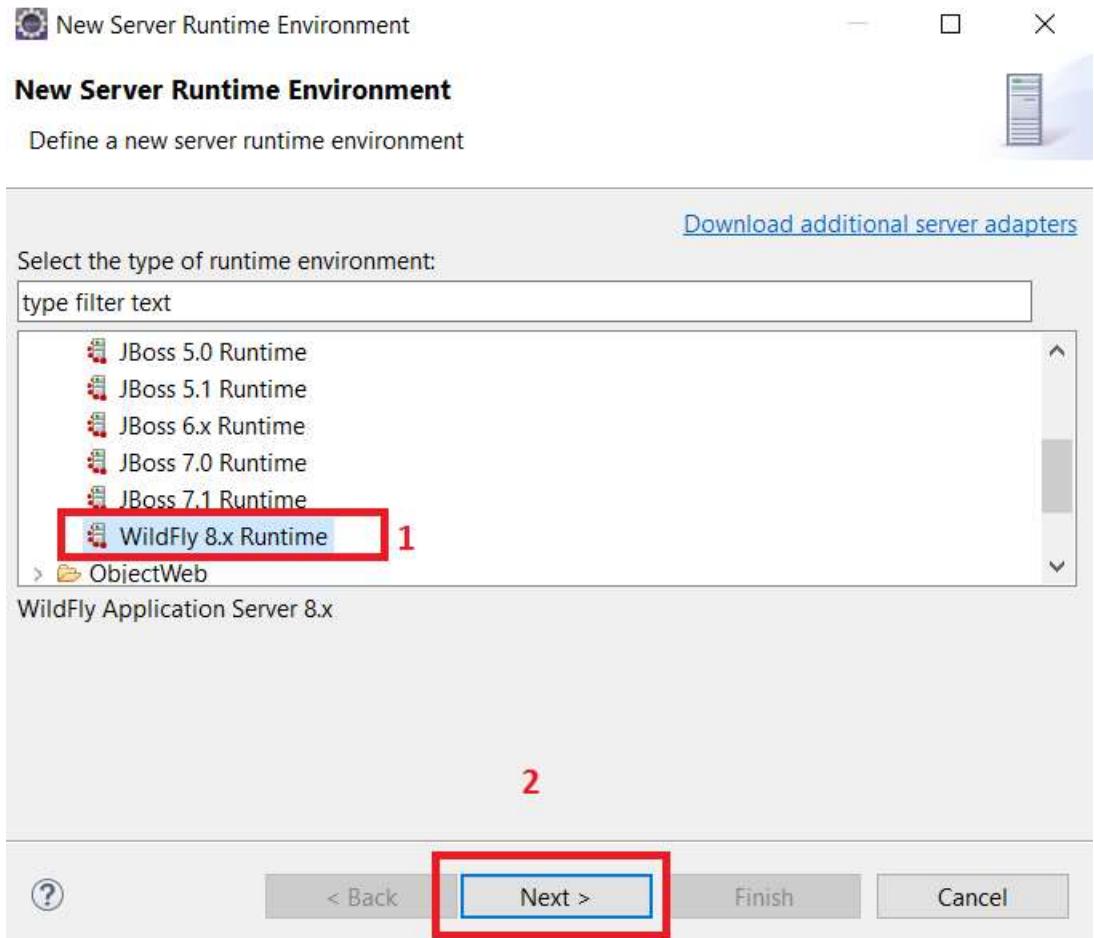
Dans Eclipse Luna, cliquer sur **Window ==>Preferences :**



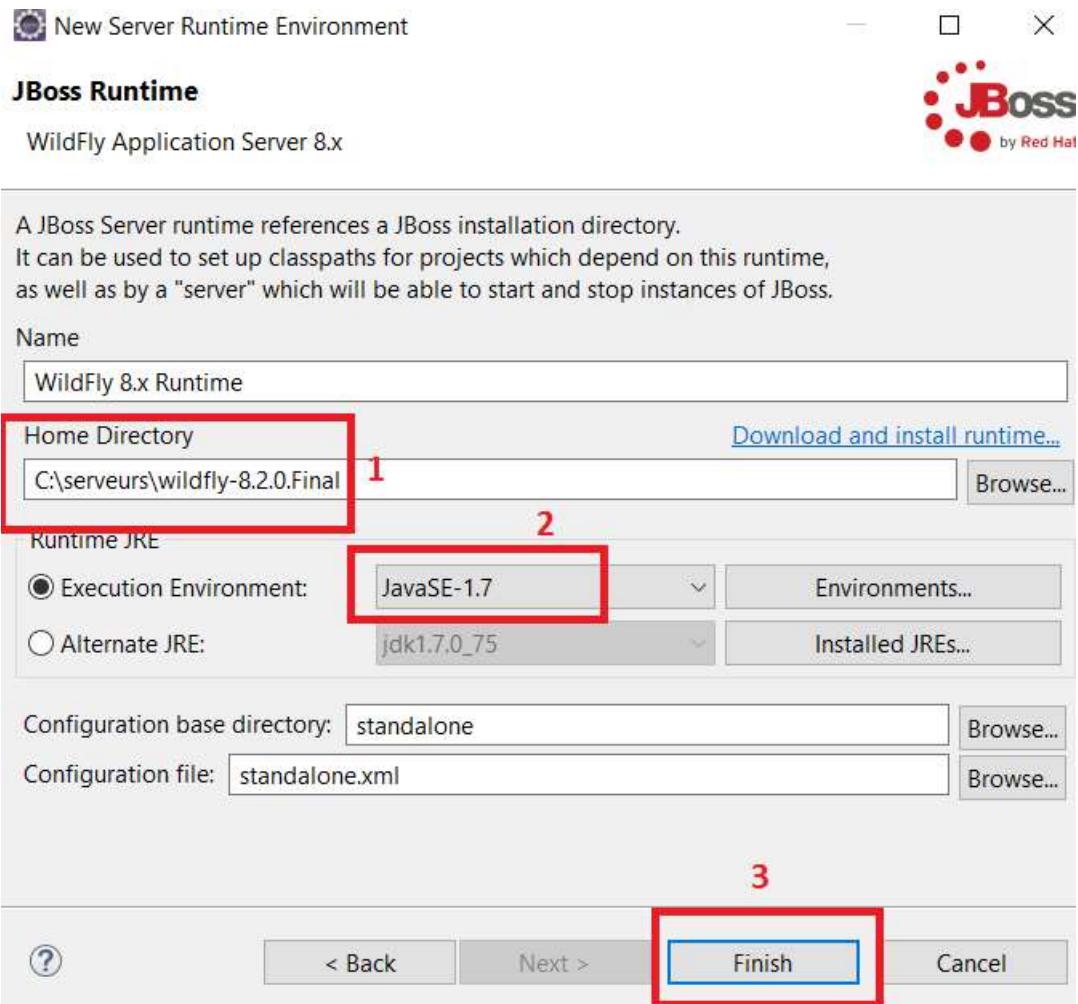
Cliquez sur **Server=> Runtime Environments :**



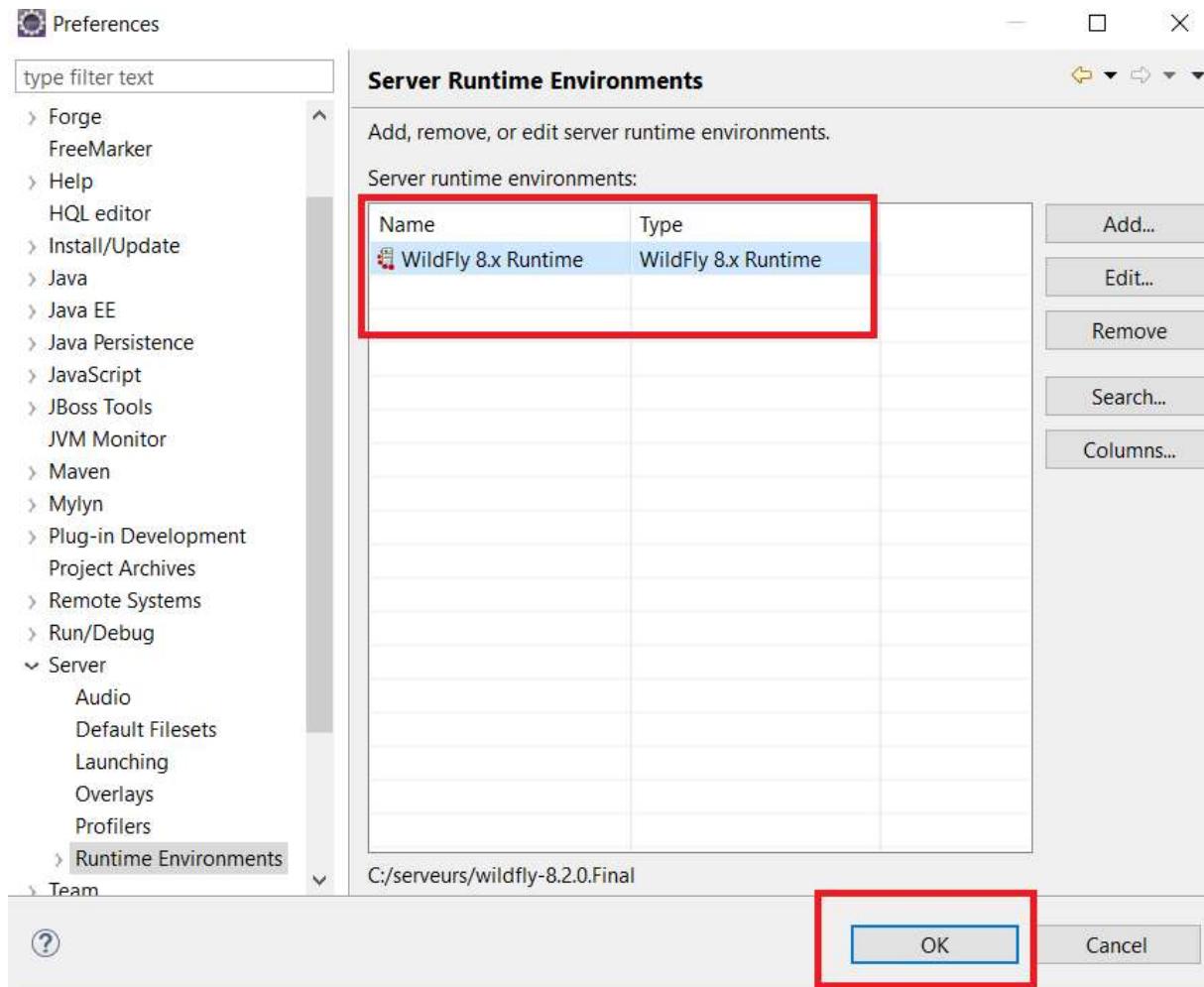
Cliquer sur le bouton **Add...** :



Sélectionner **WildFly 8.x Runtime** et cliquer sur **Next >** :



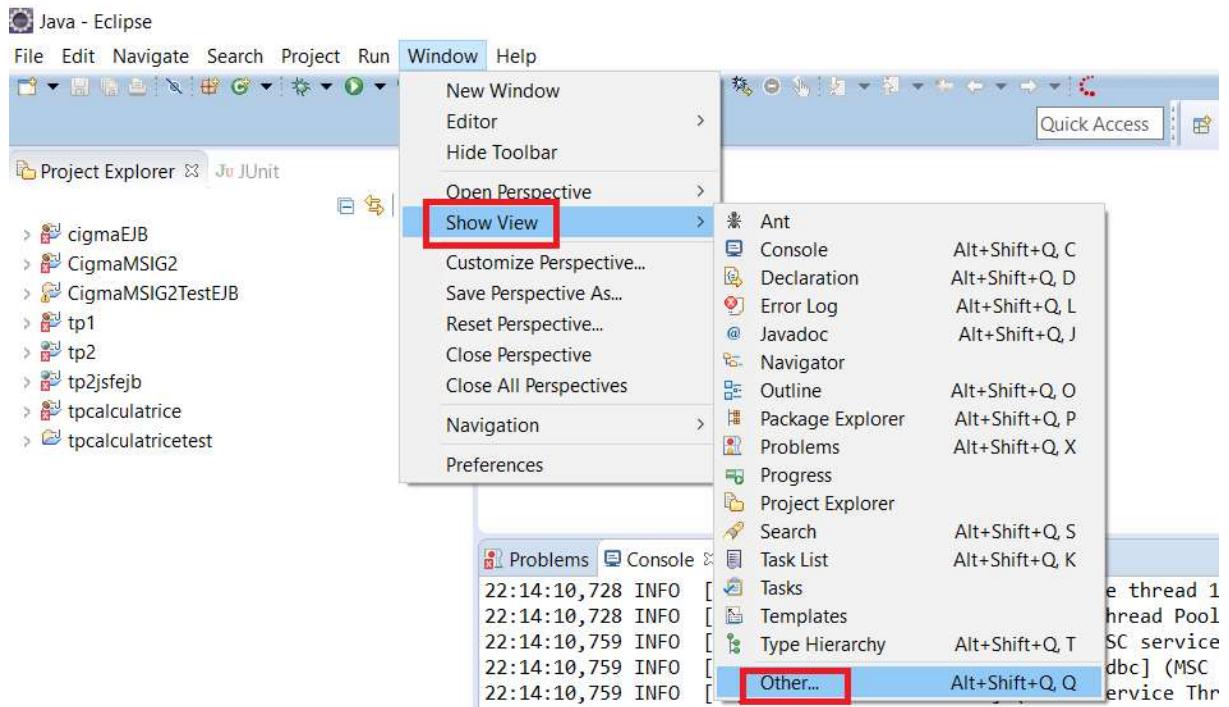
Entrer le chemin d'installation de WildFly, la JVM et cliquer ensuite sur **Finish** :



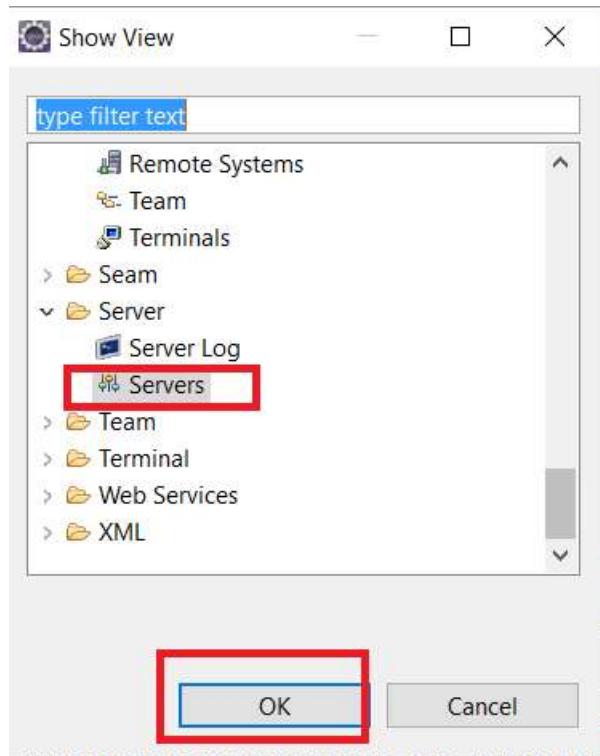
Cliquer ensuite sur **OK**.

Ajouter le serveur WildFly au niveau de la vue Servers :

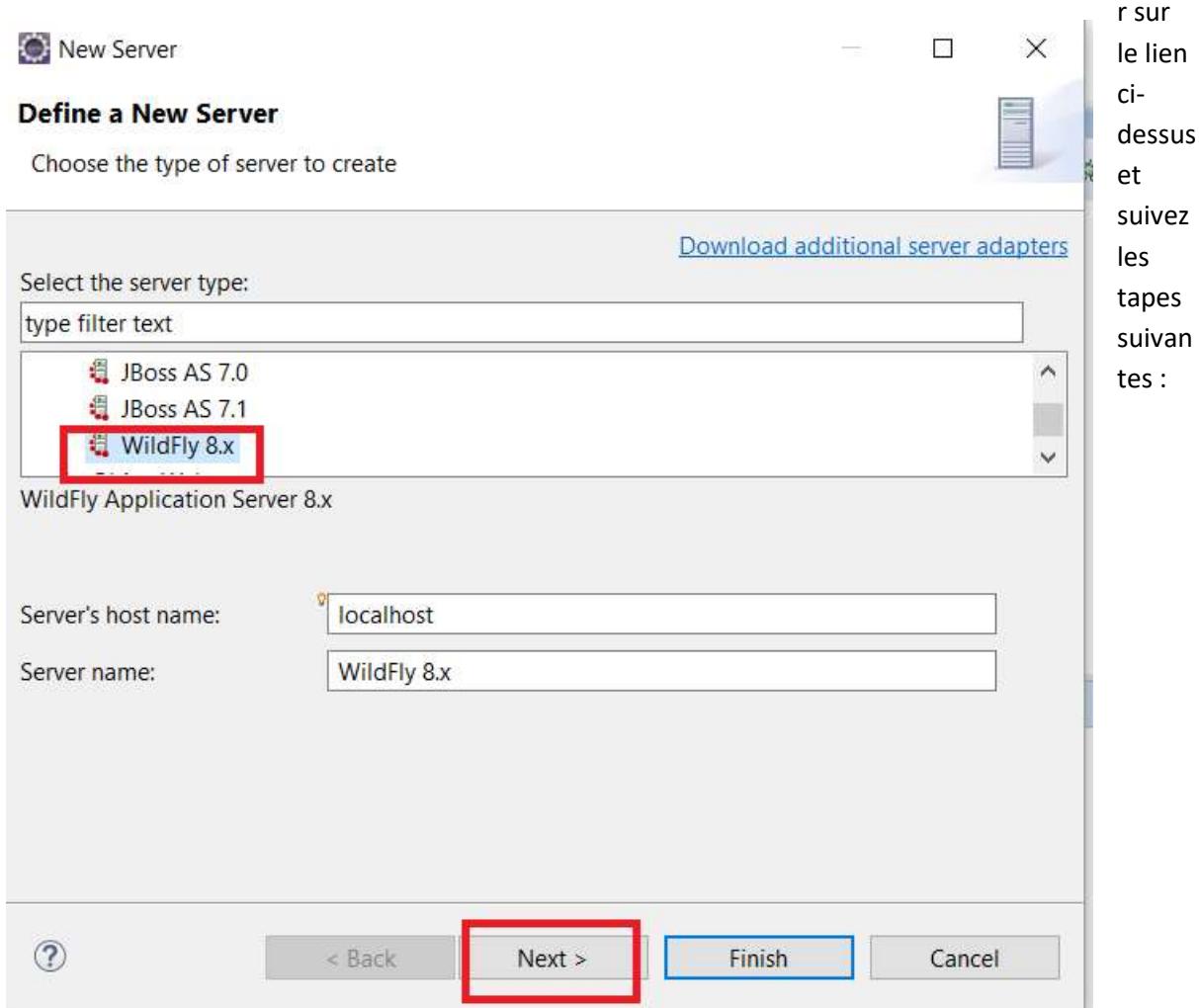
Cliquer sur **Window==>Show View==>Other...** comme illustre l'écran suivant :



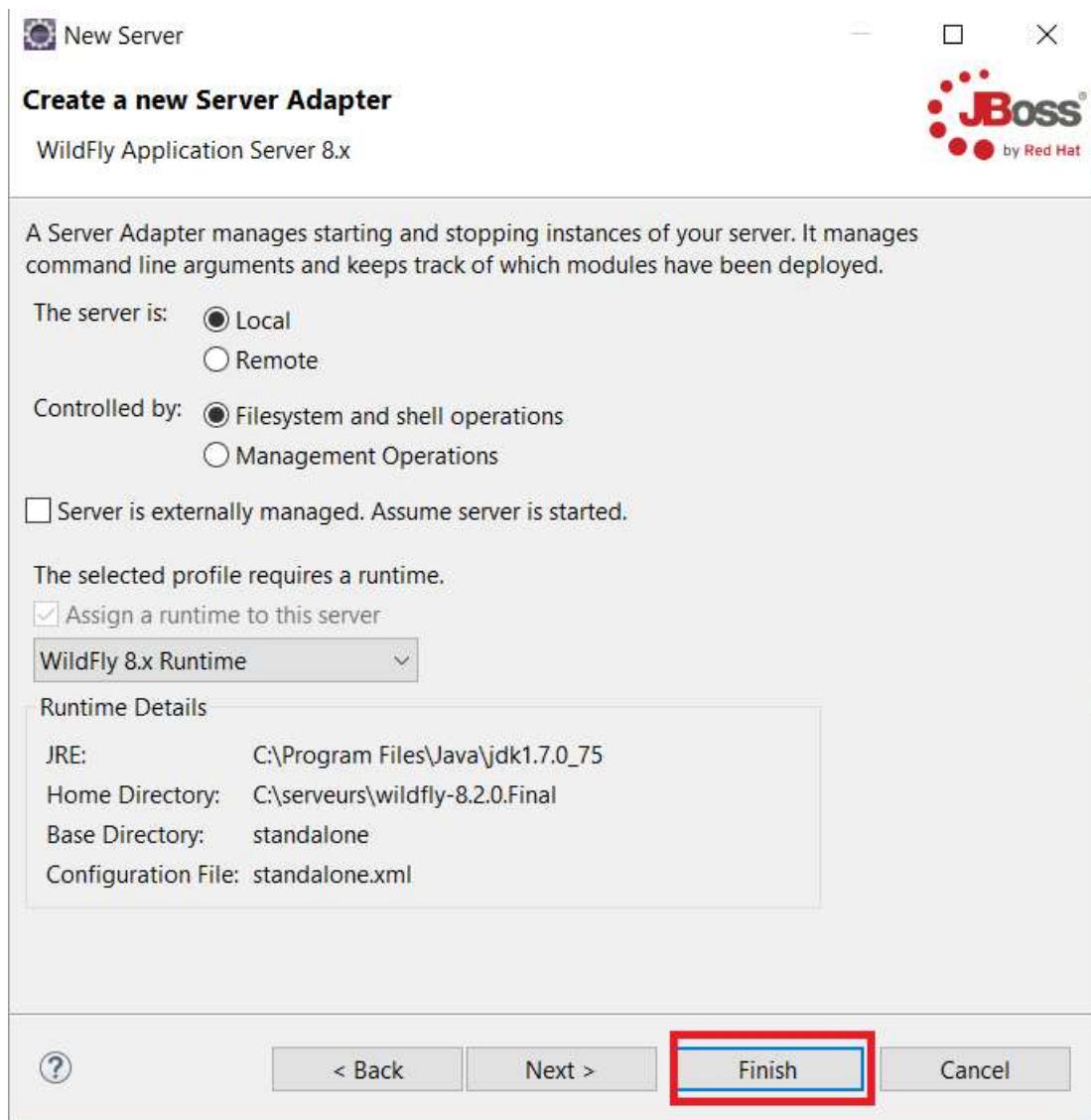
Cliquer sur **Other...**



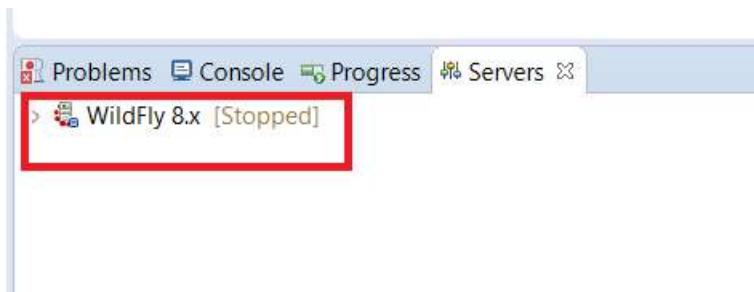
Cliquer sur **Servers** et cliquer ensuite sur le bouton **OK**:



Sélectionner **WildFly 8.x** et cliquer sur **Next >** :



Cliquez sur le bouton **Finish** :



Le serveur est ajouté comme illustré dans l'écran ci-dessus.

Pour démarrer le serveur, cliquer sur l'icône comme illustré dans l'écran ci-dessous :

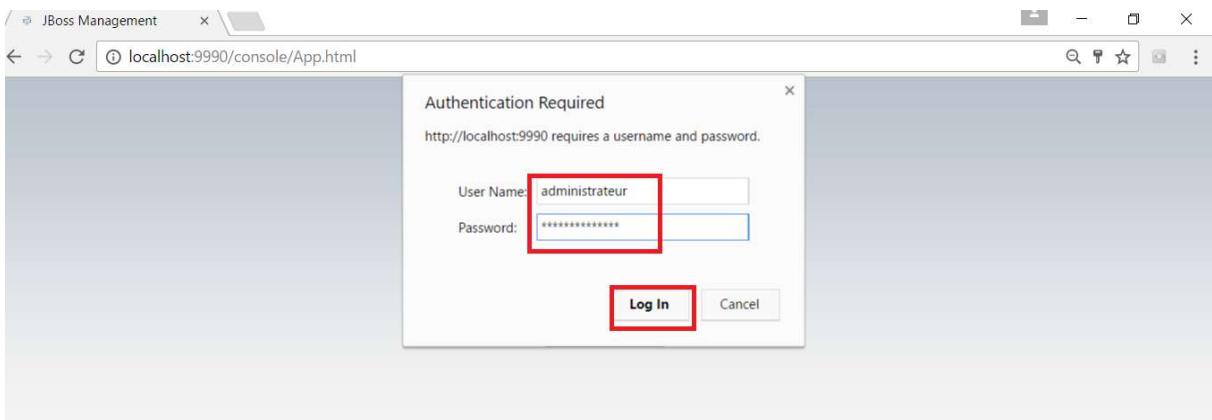


Vérifier que le serveur est bien démarré et qu'il n'y a aucune erreur au niveau de la vue Console) :

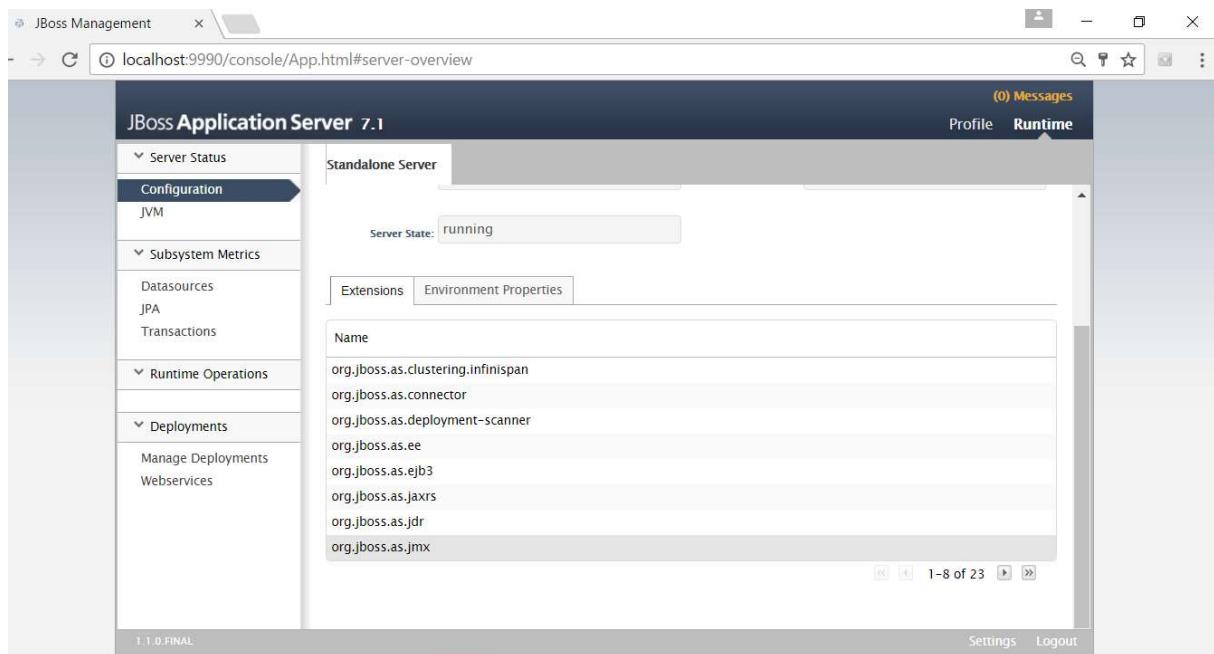
```
WILDFLY 8.X JBoss Application Server Startup Configuration C:\Program Files\Java\jdk1.8.0_101\jre\bin\javaw.exe (14 mars 2017 22:40:14)
22:40:16,202 INFO [org.jboss.as.mail.extension] (MSC service thread 1-8) JBAS015400: Bound mail session [java:jboss/mail/Default]
22:40:16,206 INFO [org.jboss.as.security] (ServerService Thread Pool -- 45) JBAS013171: Activating Security Subsystem
22:40:16,206 WARN [org.jboss.as.txn] (ServerService Thread Pool -- 45) JBAS010153: Node identifier property is set to the default value. Please
22:40:16,209 INFO [org.jboss.as.security] (MSC service thread 1-8) JBAS013170: Current PicketBox version:4.0.21.Final
22:40:16,226 INFO [org.wildfly.extension.undertow] (MSC service thread 1-2) JBAS017502: Undertow 1.1.0.Final starting
22:40:16,227 INFO [org.wildfly.extension.undertow] (ServerService Thread Pool -- 47) JBAS017502: Undertow 1.1.0.Final starting
22:40:16,306 INFO [org.jboss.remoting] (MSC service thread 1-5) JBoss Remoting version 4.0.6.Final
22:40:16,319 INFO [org.jboss.as.connector.subsystems.datasources] (ServerService Thread Pool -- 27) JBAS010404: Deploying non-JDBC-compliant dr
22:40:16,324 INFO [org.jboss.as.clustering.infinispan] (ServerService Thread Pool -- 32) JBAS010280: Activating Infinispan subsystem.
22:40:16,334 INFO [org.jboss.as.webservices] (ServerService Thread Pool -- 48) JBAS015537: Activating WebServices Extension
22:40:16,374 INFO [org.jboss.as.connector.logging] (MSC service thread 1-2) JBAS010408: Starting JCA Subsystem (IronJacamar 1.1.9.Final)
22:40:16,379 INFO [org.jboss.as.connector.deployers.jdbc] (MSC service thread 1-1) JBAS010417: Started Driver service with driver-name = h2
22:40:16,380 INFO [org.jboss.as.connector.deployers.jdbc] (MSC service thread 1-1) JBAS010417: Started Driver service with driver-name = com.my
22:40:16,675 INFO [org.wildfly.extension.undertow] (ServerService Thread Pool -- 47) JBAS017527: Creating file handler for path C:\serveurs\wil
22:40:16,690 INFO [org.wildfly.extension.undertow] (MSC service thread 1-4) JBAS017525: Started server default-server.
22:40:16,711 INFO [org.wildfly.extension.undertow] (MSC service thread 1-8) JBAS017531: Host default-host starting
22:40:16,973 INFO [org.jboss.as.server.deployment.scanner] (MSC service thread 1-5) JBAS015012: Started FileSystemDeploymentService for directo
22:40:17,056 INFO [org.wildfly.extension.undertow] (MSC service thread 1-1) JBAS017519: Undertow HTTP listener default listening on localhost/1
22:40:17,197 INFO [org.jboss.as.connector.subsystems.datasources] (MSC service thread 1-7) JBAS010400: Bound data source [java:jboss/datasource
22:40:17,197 INFO [org.jboss.as.connector.subsystems.datasources] (MSC service thread 1-2) JBAS010400: Bound data source [java:/MySqlDS]
22:40:17,375 INFO [org.jboss.ws.common.management] (MSC service thread 1-8) JBWS02052: Starting JBoss Web Services - Stack CXF Server 4.3.2.Fi
22:40:17,624 INFO [org.jboss.as] (Controller Boot Thread) JBAS015961: Http management interface listening on http://127.0.0.1:9990/management
22:40:17,625 INFO [org.jboss.as] (Controller Boot Thread) JBAS015951: Admin console listening on http://127.0.0.1:9990
22:40:17,625 INFO [org.jboss.as] (Controller Boot Thread) JBAS015874: WildFly 8.2.0.Final "Tweek" started in 3501ms - Started 190 of 240 servic
```

### Accès à la console d'administration de WildFly :

Lancer le lien <http://localhost:8080> :

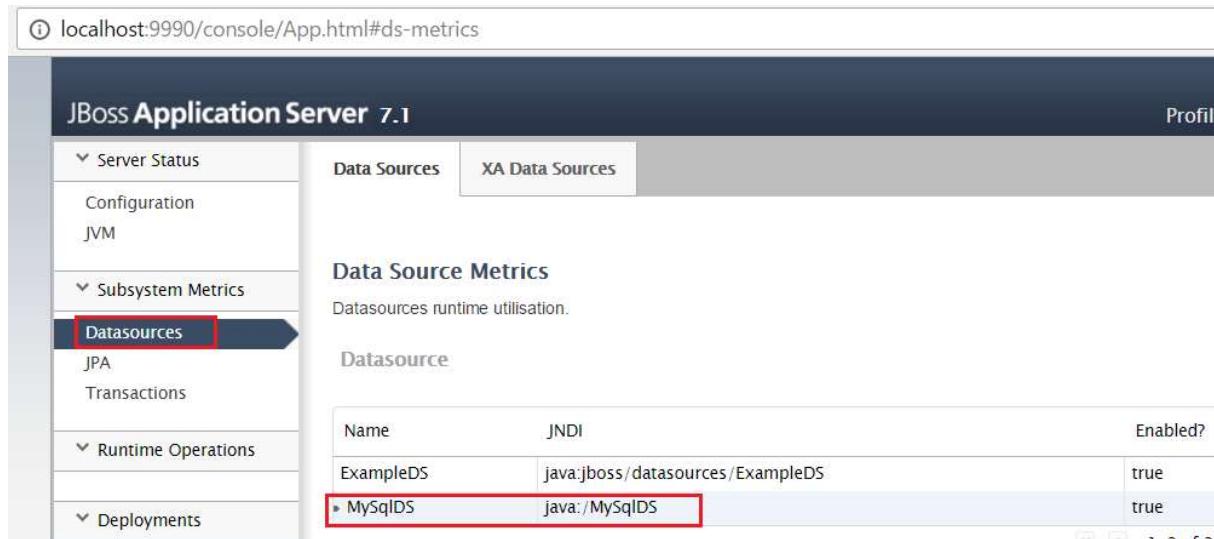


Saisir votre compte administrateur et cliquer sur le bouton Log In :



The screenshot shows the JBoss Application Server 7.1 Management Console. The left sidebar has a 'Configuration' section highlighted. The main area shows a 'Standalone Server' tab with a 'Server State: running' status. Below it is a table of subsystems, with the 'Datasources' row expanded to show various JNDI names: org.jboss.as.clustering.infinispan, org.jboss.as.connector, org.jboss.as.deployment-scanner, org.jboss.as.ee, org.jboss.as.ejb3, org.jboss.as.jaxrs, org.jboss.as.jdr, and org.jboss.as.jmx. A navigation bar at the bottom includes 'Settings' and 'Logout'.

Vous pouvez consulter la source de données MySqlDS comme illustré au niveau de l'écran ci-après :

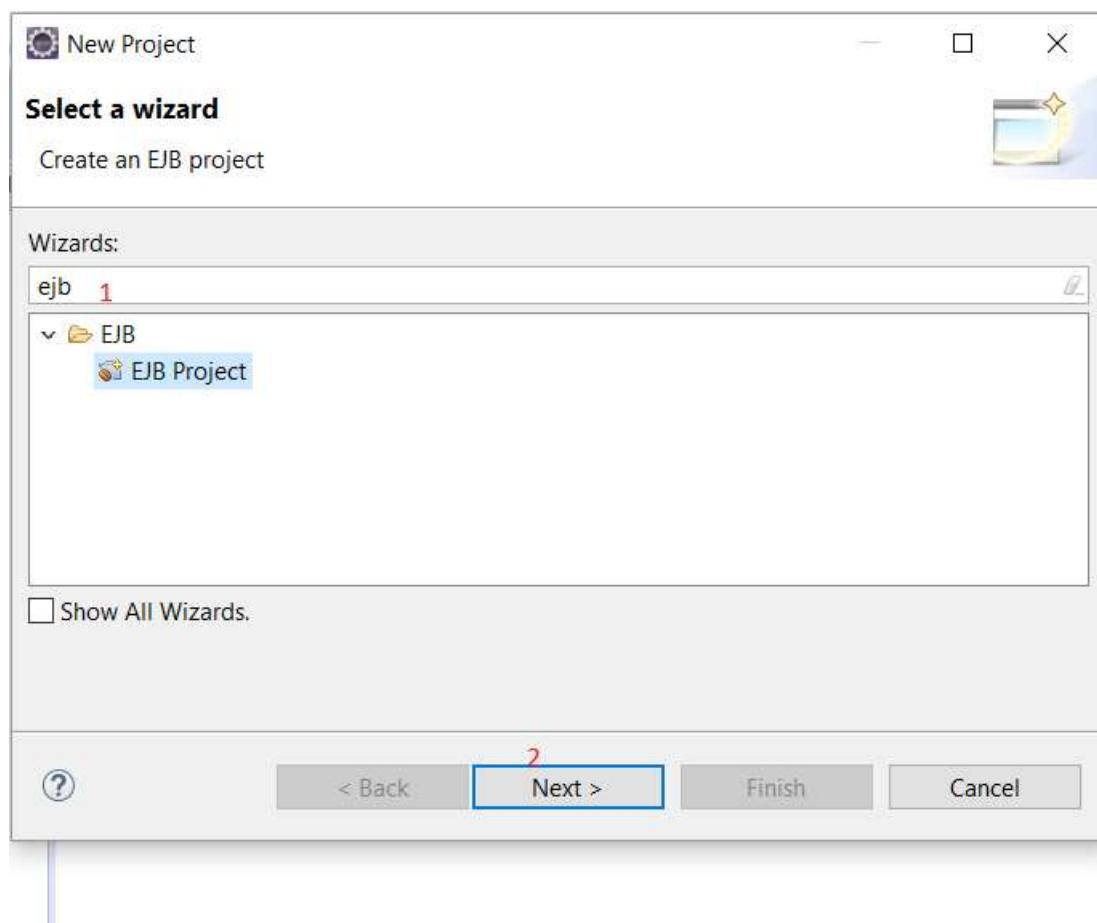
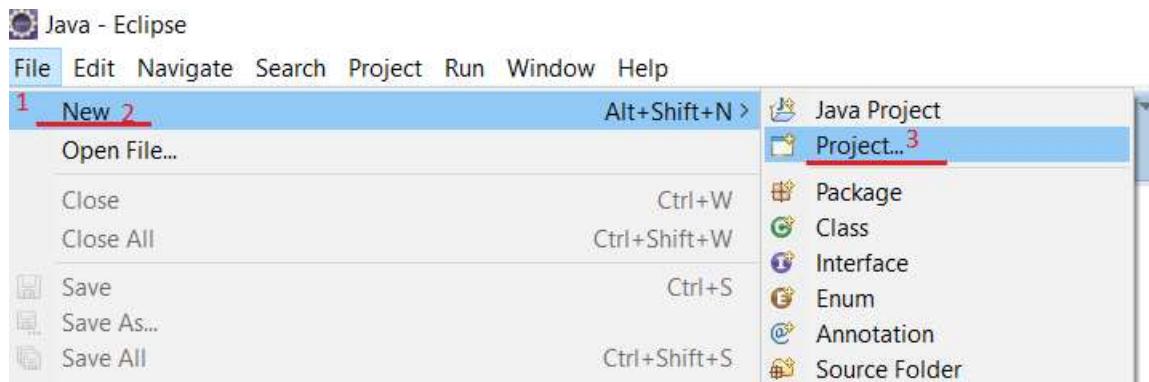


The screenshot shows the 'Data Sources Metrics' page. The left sidebar has a 'Datasources' section highlighted. The main area shows a table of data sources. The 'MySqlDS' row is selected and highlighted with a red box. The table columns are Name, JNDI, and Enabled?. The 'MySqlDS' row has values: Name 'MySqlDS', JNDI 'java:/MySqlDS', and Enabled? 'true'.

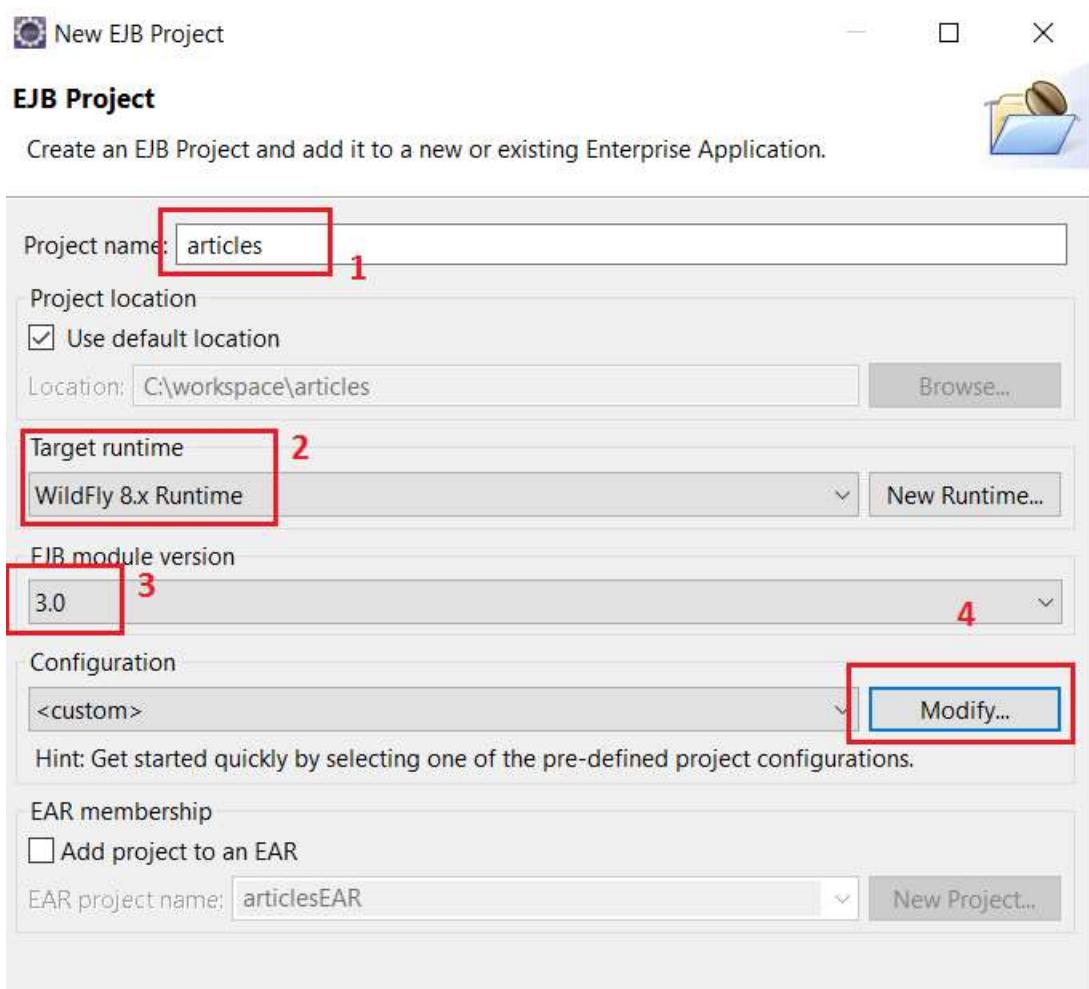
Name	JNDI	Enabled?
ExampleDS	java:jboss/datasources/ExampleDS	true
MySqlDS	java:/MySqlDS	true

## 4. Développement de l'application EJB 3.0

Créer un nouveau projet EJB. Pour ceci suivre les étapes suivantes :

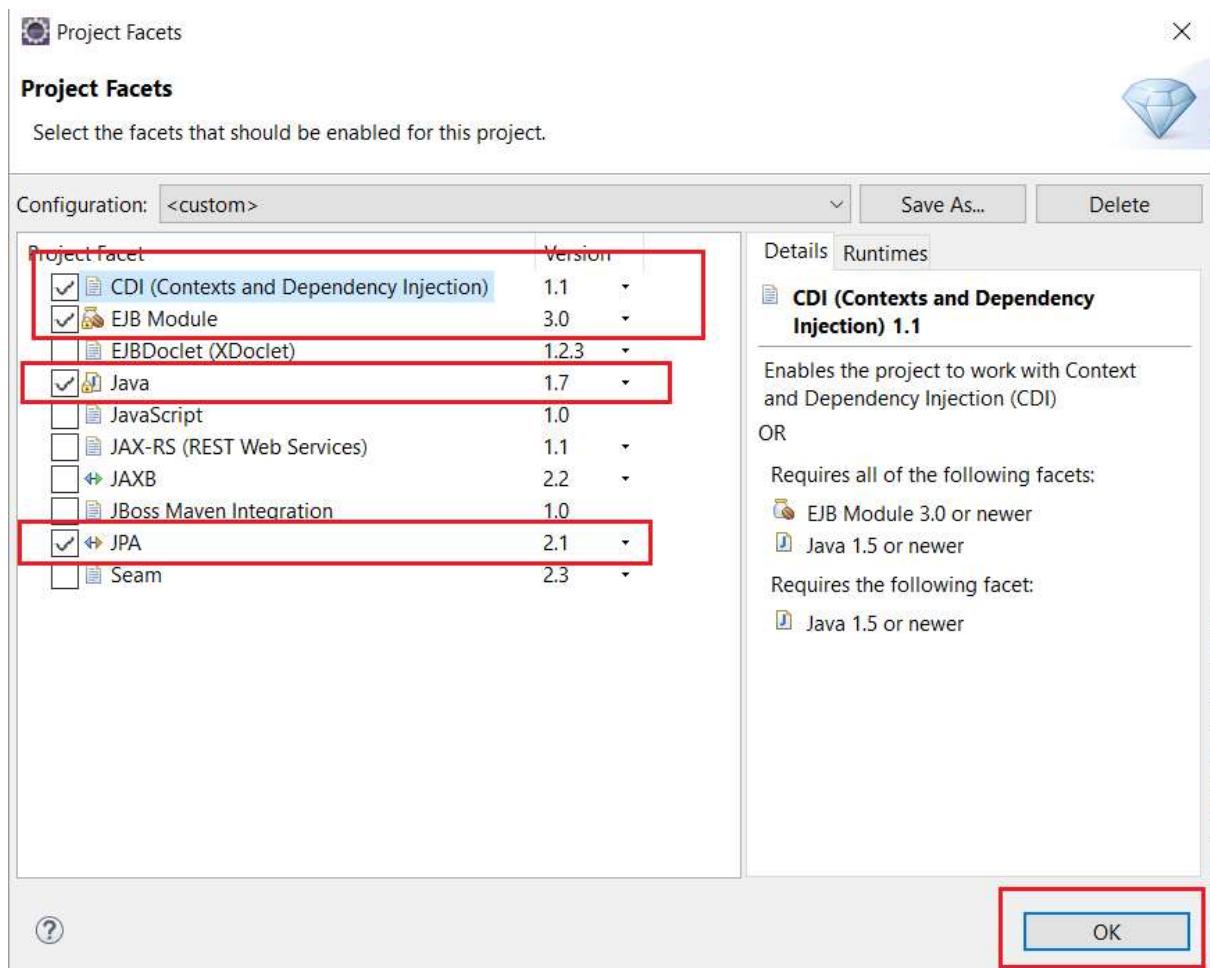


Sélectionner le type de projet "**EJB project**" et actionner ensuite le bouton **Next >** :



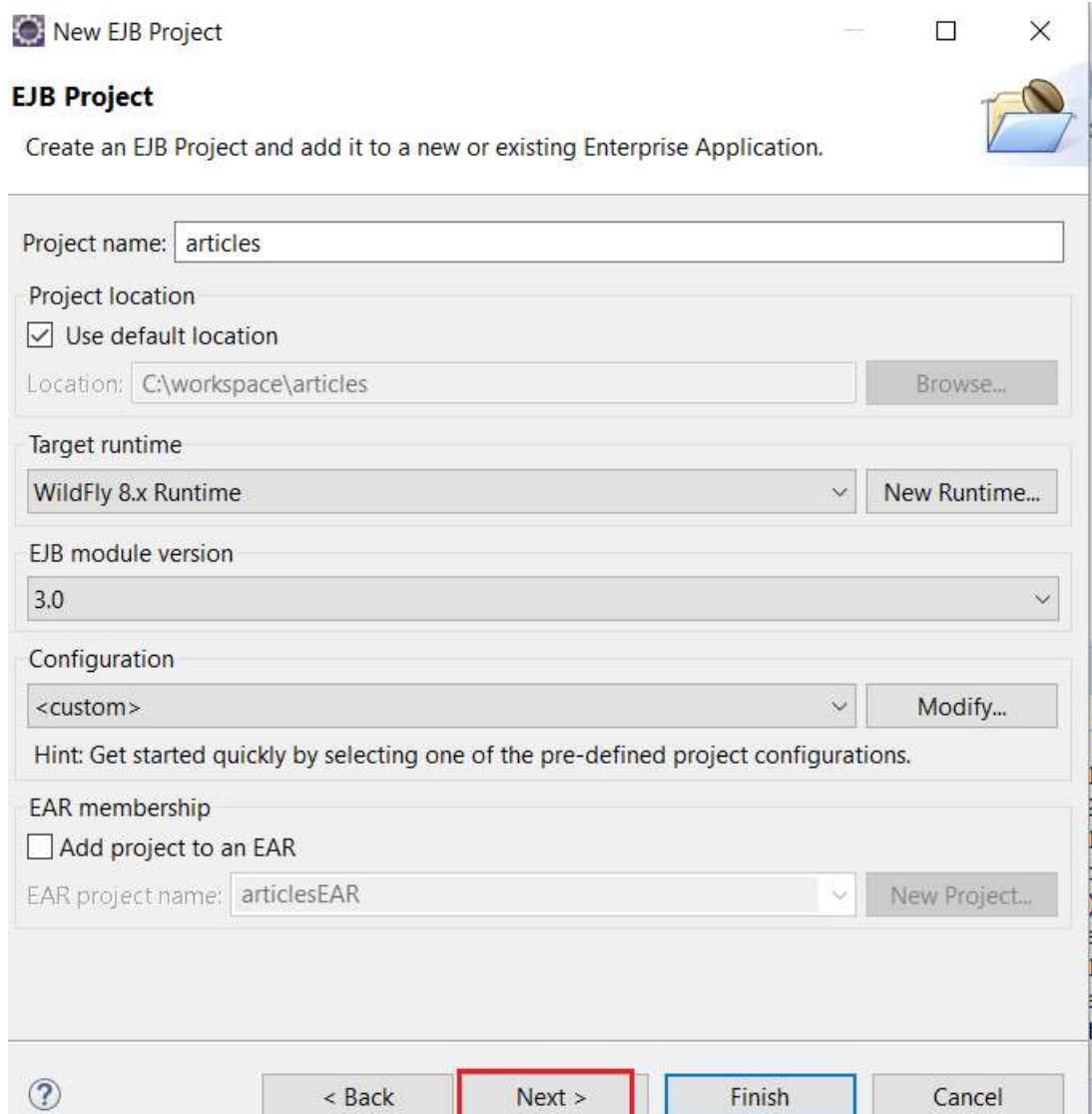
- 1- Entrer le nom de votre projet (ici : articles).
- 2- Sélectionner "Target Runtime".
- 3- Sélectionner EJB 3.0.

Cliquez ensuite sur le bouton **Modify...** :

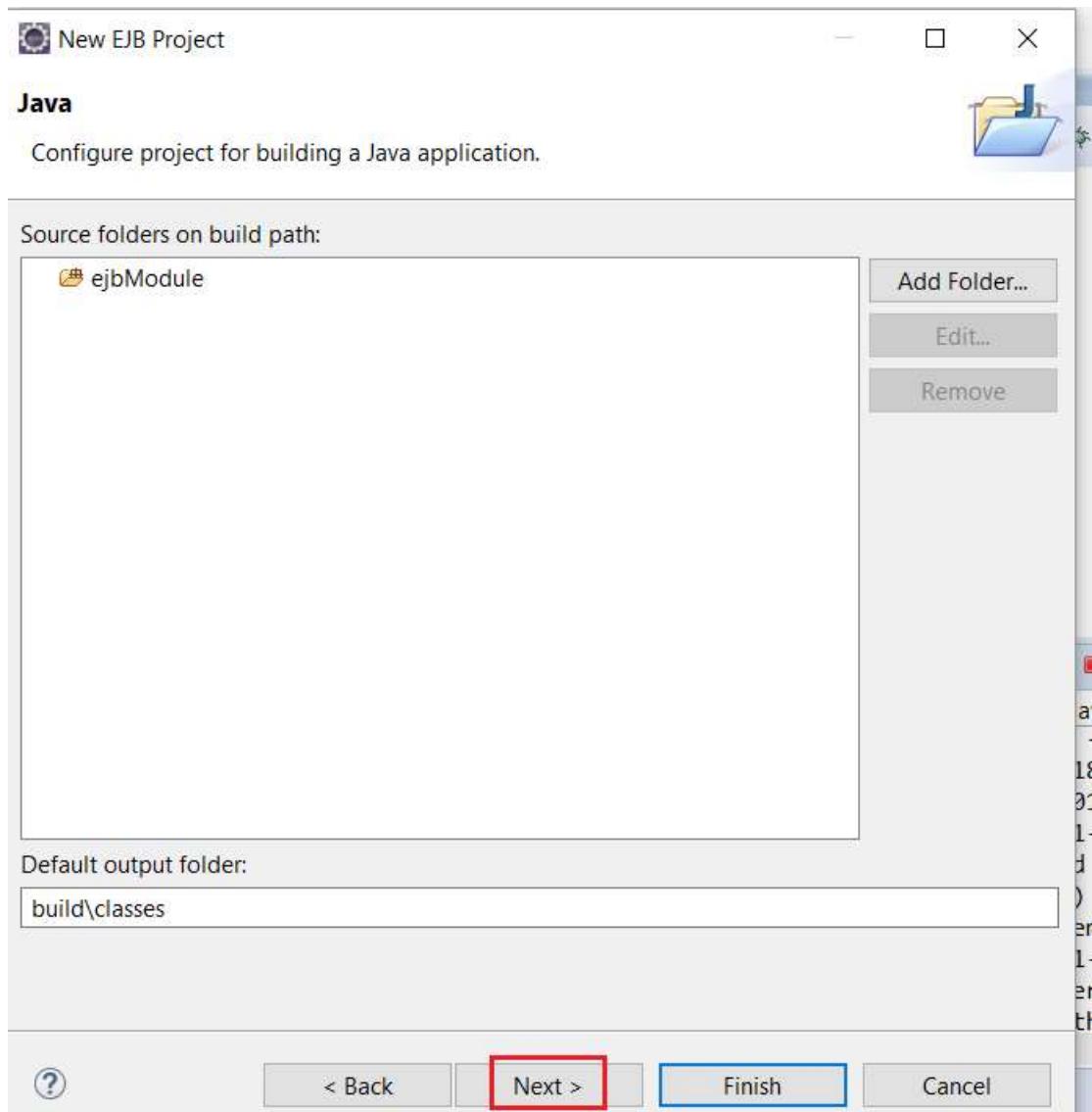


- 1- Cocher CDI (version 1.1).
- 2- Cocher EJB Module version 3.0.
- 3- Cocher Java version 1.7.
- 4- Cocher JPA version 2.1.

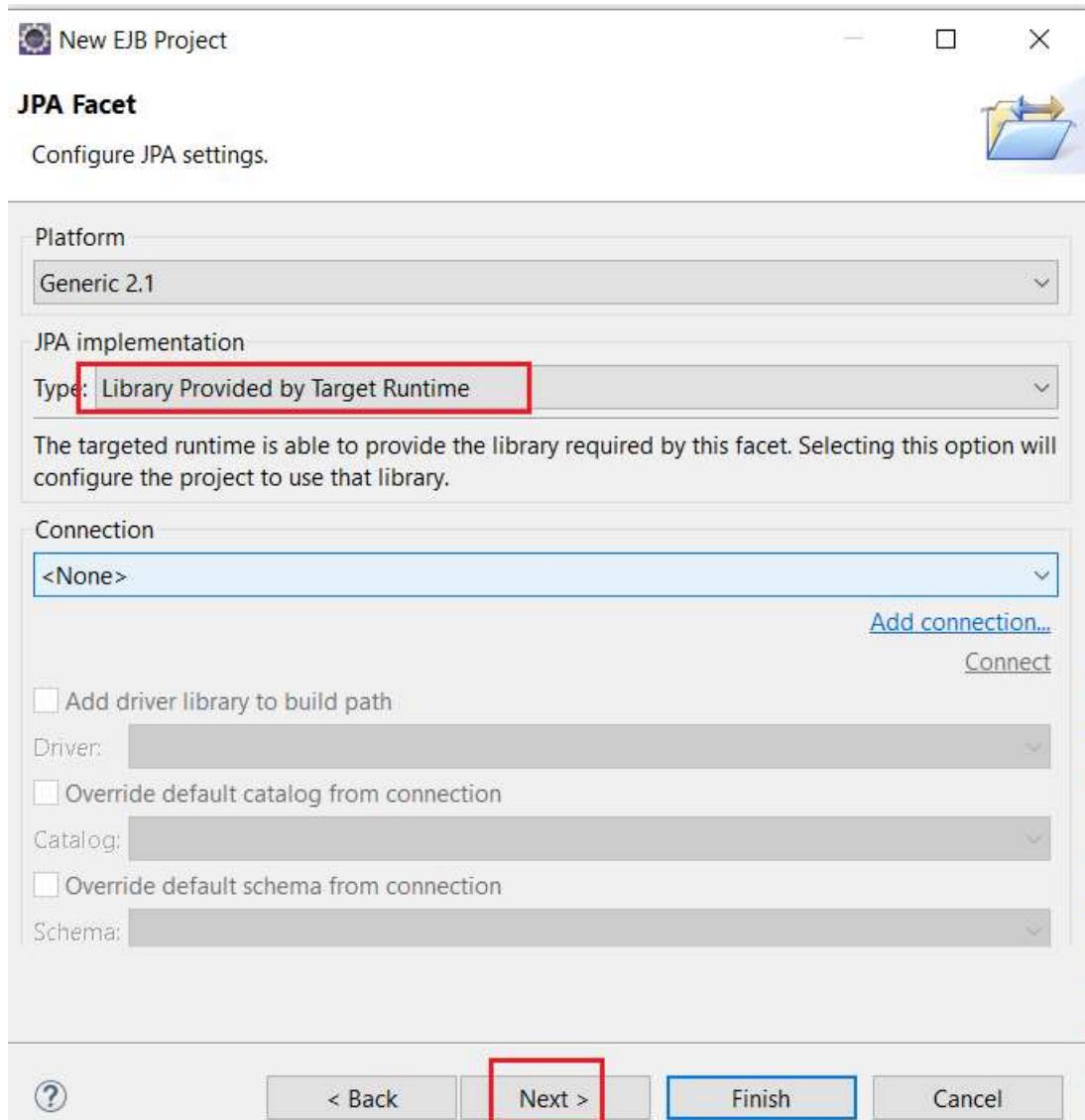
Cliquer ensuite sur le bouton **OK** :



Cliquez sur **Next >** :



Cliquer sur **Next >** :



Cliquer sur **Next >** :

## New EJB Project

### EJB Module

Configure EJB module settings.



#### EJB Client JAR

Create an EJB Client JAR module to hold the client interfaces and classes

Name: articlesClient

Client JAR URI: articlesClient.jar

Generate ejb-jar.xml deployment descriptor



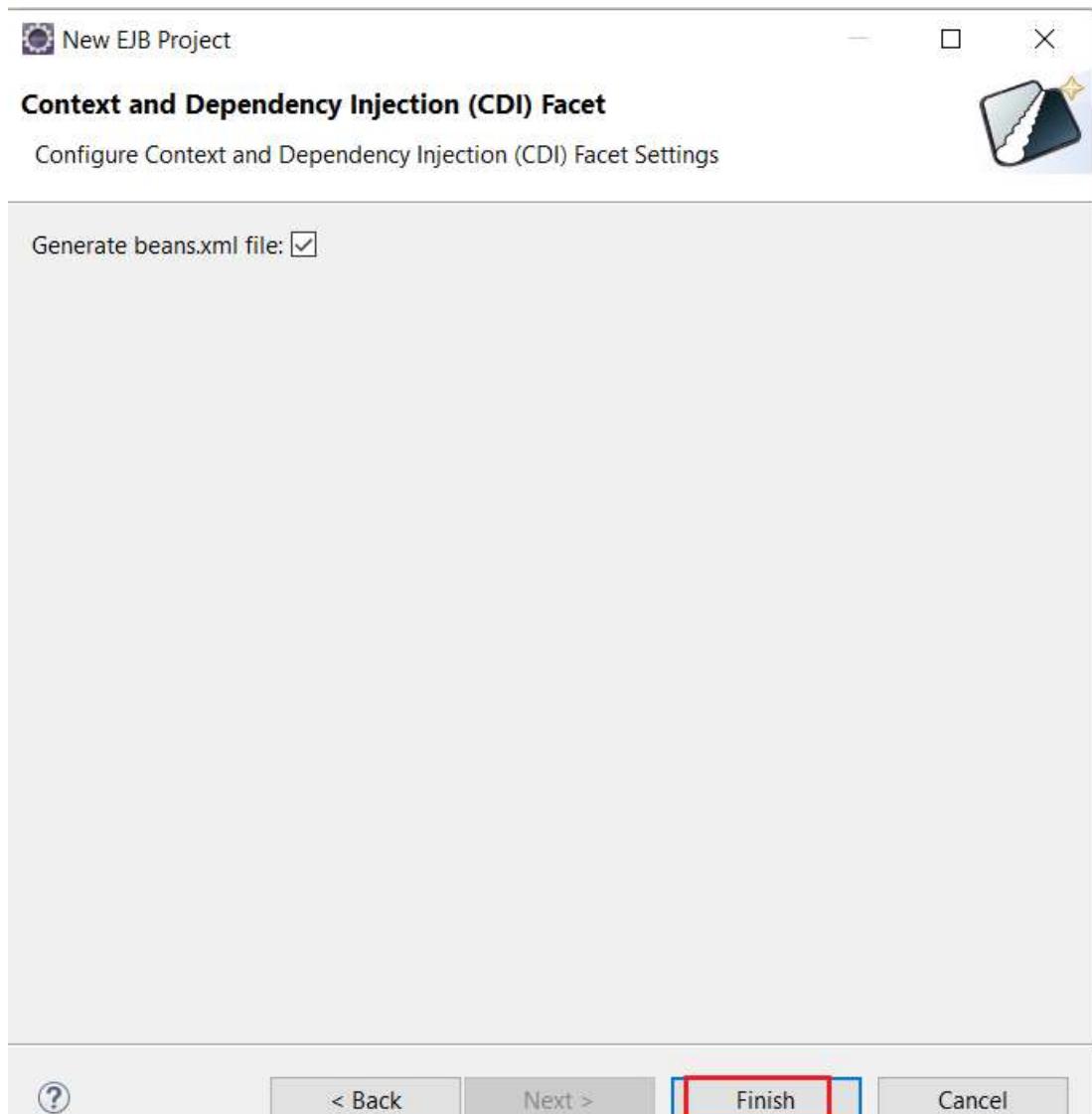
< Back

Next >

Finish

Cancel

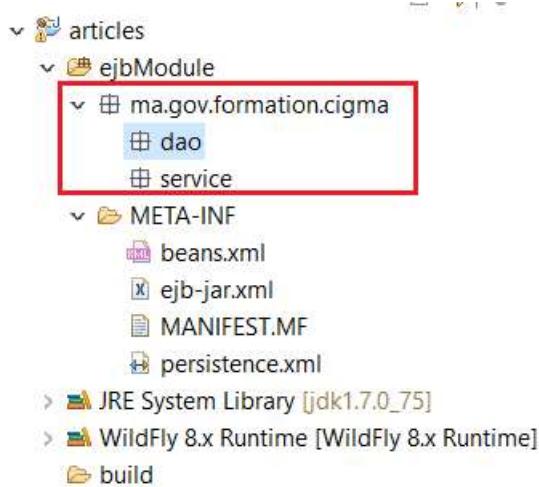
Cliquez sur **Next >** :



Cliquer sur **Finish**. L'arborescence du projet suivante est créée :

```
articles
└── ejbModule
    ├── META-INF
    │   ├── beans.xml
    │   ├── ejb-jar.xml
    │   ├── MANIFEST.MF
    │   └── persistence.xml
    ├── JRE System Library [jdk1.7.0_75]
    └── WildFly 8.x Runtime [WildFly 8.x Runtime]
    └── build
```

- Créer les deux packages **ma.gov.formation.cigma.service** et **ma.gov.formation.cigma.dao** :



- Créer la classe **Article** suivante :

```
package ma.gov.formation.cigma.service;

import java.io.Serializable;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;

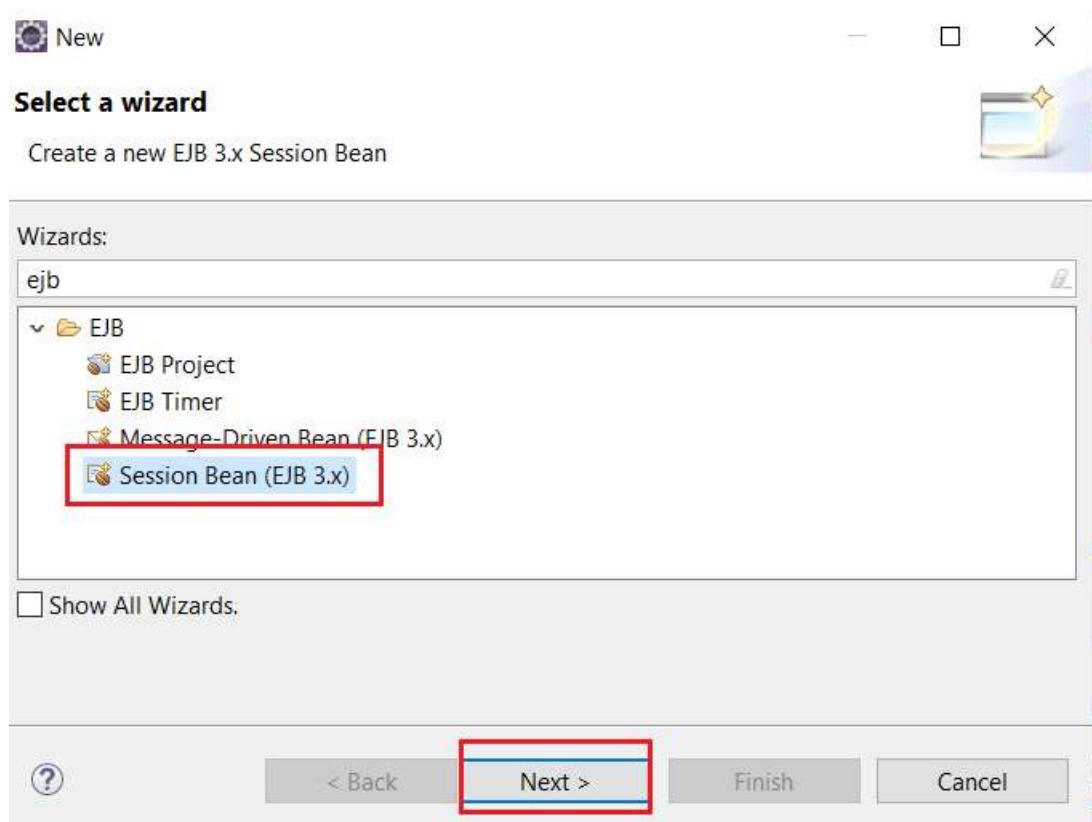
@Entity
public class Article implements Serializable {
    private static final long serialVersionUID = 6161046340508532663L;
    @Id
    @GeneratedValue
    private Integer id;
    private String designation;
    private Double prix;
    private Integer quantiteEnStock;
    public Article() {
    }
    public Article(String designation, Double prix, Integer quantiteEnStock) {
        super();
        this.designation = designation;
        this.prix = prix;
        this.quantiteEnStock = quantiteEnStock;
    }
    @Override
    public String toString() {
        return "Article [id=" + id + ", designation=" + designation + ", "
            + prix + ", quantiteEnStock=" + quantiteEnStock + "]";
    }
    public Integer getId() {
        return id;
    }
    public void setId(Integer id) {
        this.id = id;
    }
    public String getDesignation() {
        return designation;
    }
    public void setDesignation(String designation) {
```

```

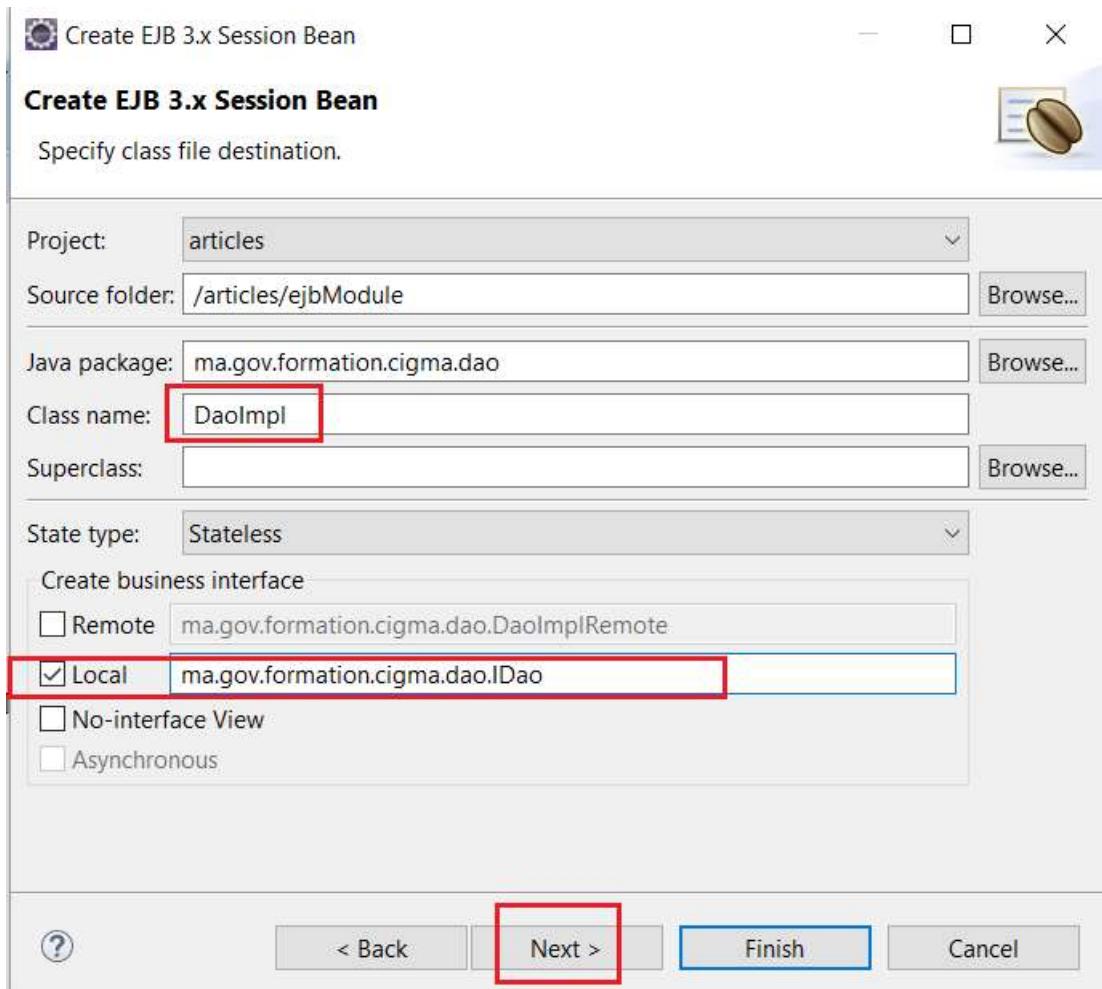
        this.designation = designation;
    }
    public Double getPrix() {
        return prix;
    }
    public void setPrix(Double prix) {
        this.prix = prix;
    }
    public Integer getQuantiteEnStock() {
        return quantiteEnStock;
    }
    public void setQuantiteEnStock(Integer quantiteEnStock) {
        this.quantiteEnStock = quantiteEnStock;
    }
}

```

- Créer l'EJB Session **Daolmpl** dans le package dao :

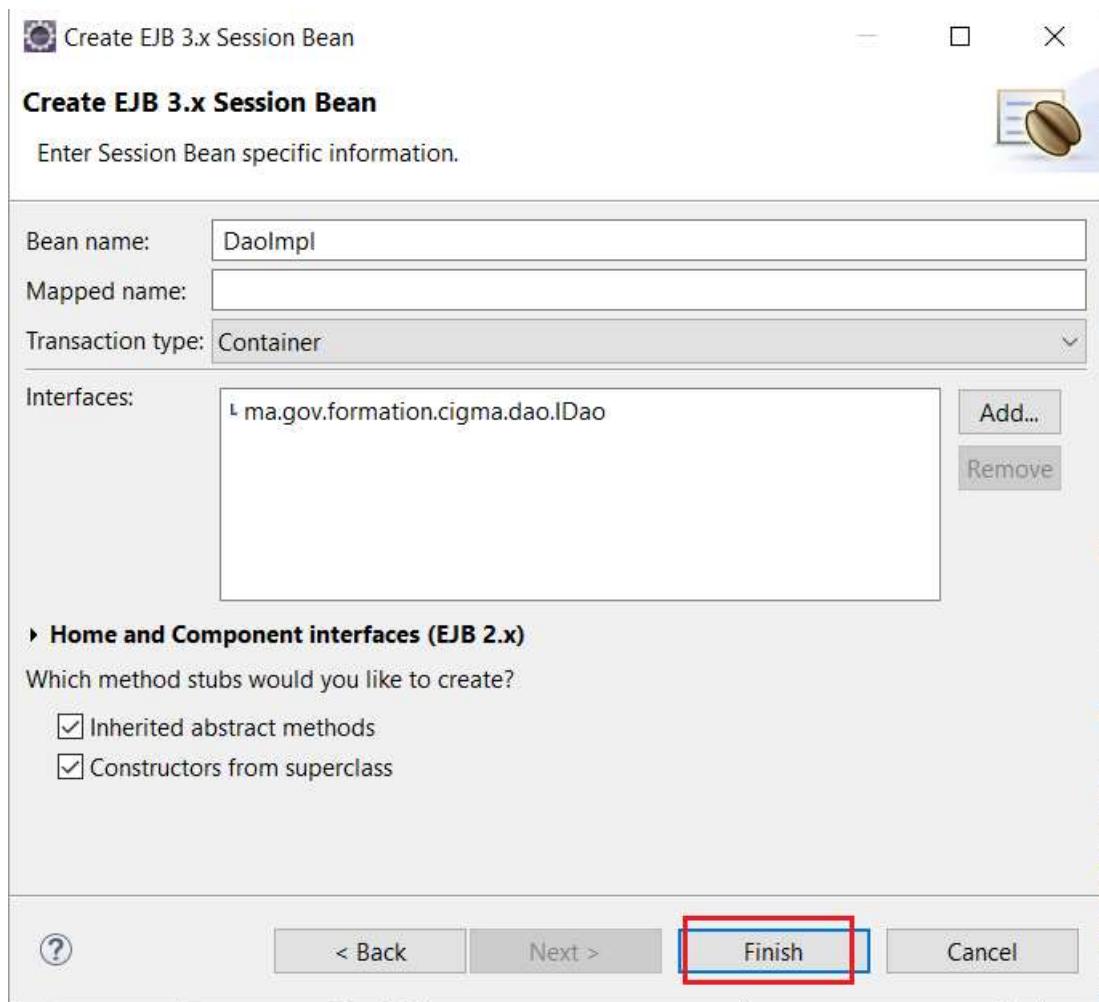


Cliquer sur **Next >** :



1. Entrer le nom de la classe (Daolmpl ici).
2. Cocher Local et saisir le nom de l'interface.

Cliquer ensuite sur **Next >** :



Cliquez sur **Finish**. L'interface IDao et la classe Daolmpl suivantes seront créées :

```
package ma.gov.formation.cigma.dao;

import javax.ejb.Local;

@Local
public interface IDao {
```

```
package ma.gov.formation.cigma.dao;

import javax.ejb.Stateless;

@Stateless
public class DaoImpl implements IDao {

    public DaoImpl() {
    }

}
```

- Créer les méthodes DAO suivantes :

```
package ma.gov.formation.cigma.dao;

import java.util.List;

import javax.ejb.Local;

import ma.gov.formation.cigma.service.Article;

@Local
public interface IDao {
    public void save(Article a);
    public Article getById(Integer id);
    public List<Article> getAll();
    public void remove(Integer id);
}
```

```
package ma.gov.formation.cigma.dao;

import java.util.List;

import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;

import ma.gov.formation.cigma.service.Article;

@Stateless
public class DaoImpl implements IDao {
    @PersistenceContext
    private EntityManager em;

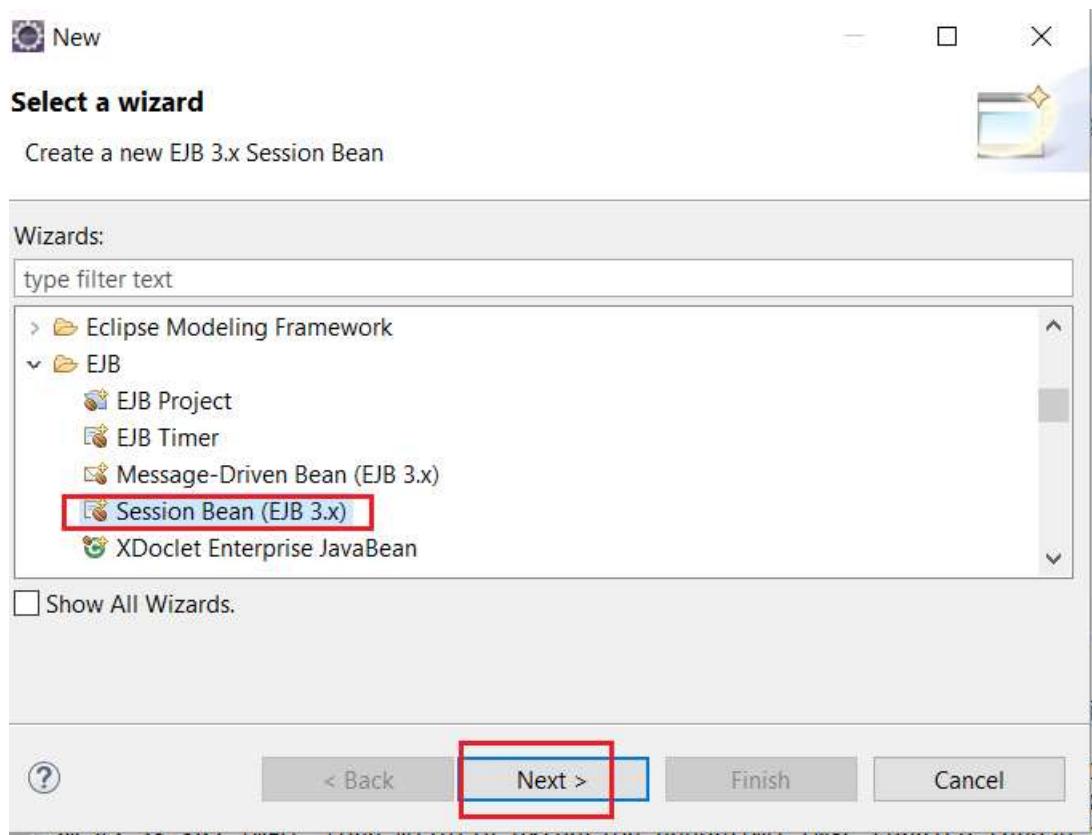
    @Override
    public void save(Article a) {
        em.merge(a);
    }

    @Override
    public Article getById(Integer id) {
        return em.find(Article.class, id);
    }

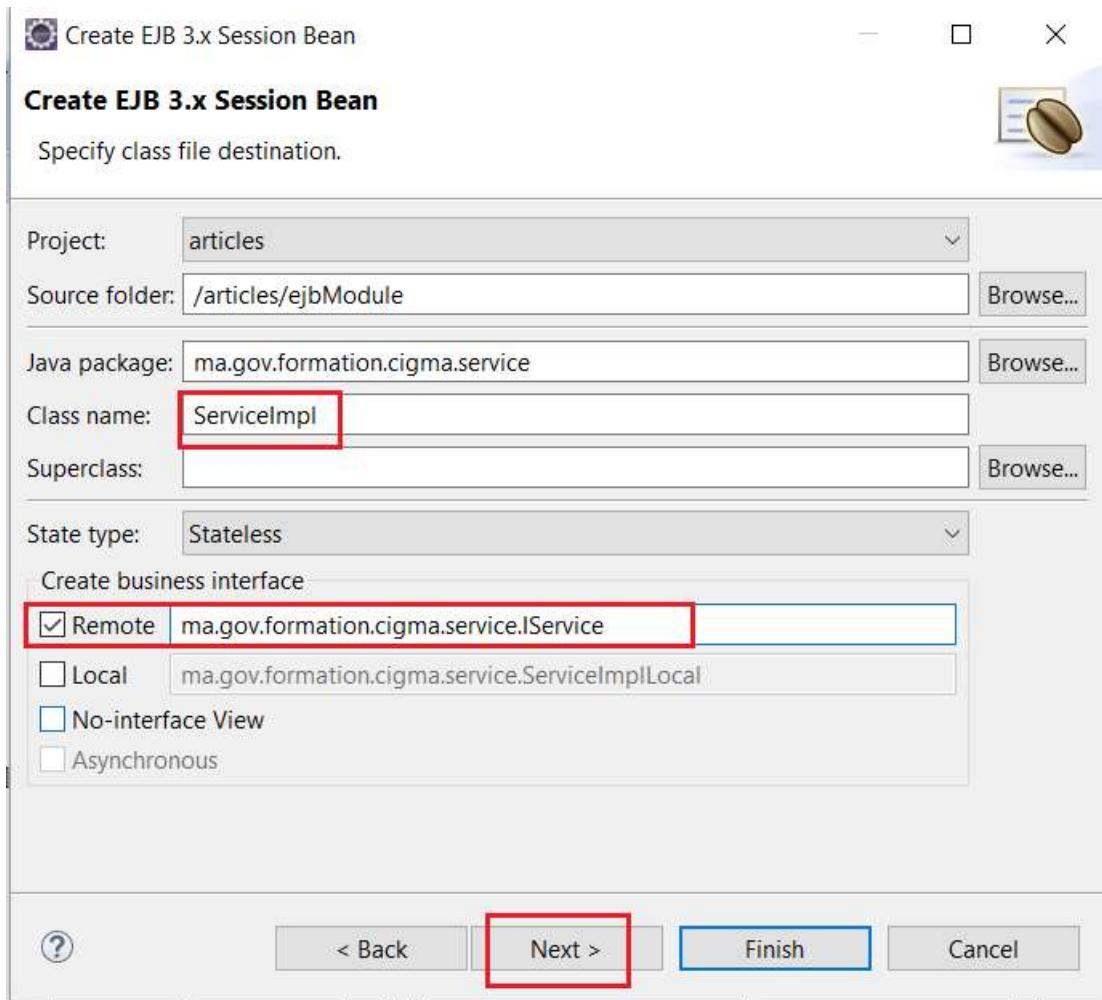
    @Override
    public List<Article> getAll() {
        return em.createQuery("from Article").getResultList();
    }

    @Override
    public void remove(Integer id) {
        Article a=getById(id);
        if (a != null)
            em.remove(a);
    }
}
```

- Créer l'EJB Session ServiceImpl :

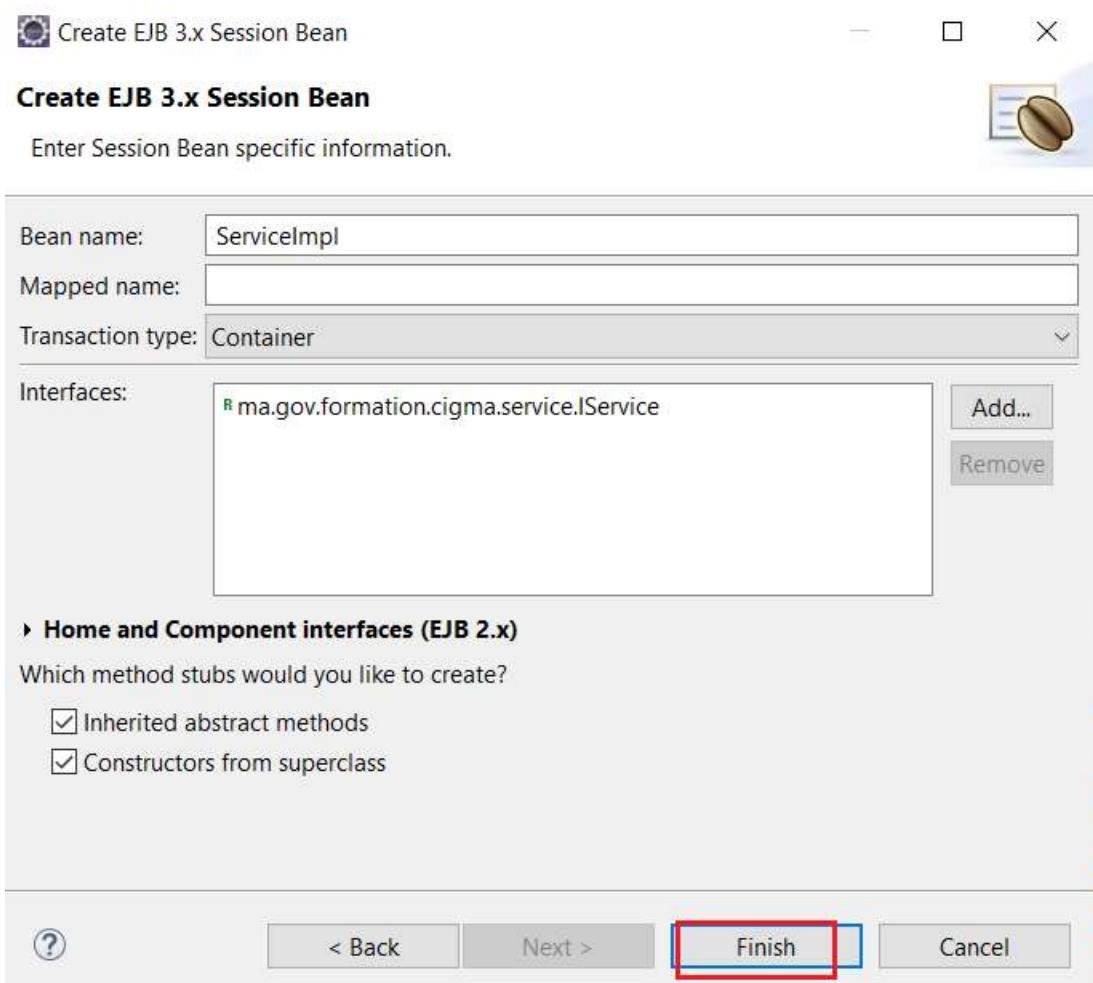


Cliquer sur **Next >** :



1. Entrer le nom de la classe EJB Session.
2. Cocher Remote et saisir le nom de l'interface.

Cliquer sur **Next >** :



Cliquer sur **Finish**. L'interface **IService** et la classe **ServiceImpl** seront créées:

```
package ma.gov.formation.cigma.service;

import javax.ejb.Remote;

@Remote
public interface IService {
}
```

```
package ma.gov.formation.cigma.service;

import javax.ejb.Stateless;

@Stateless
public class ServiceImpl implements IService {

    public ServiceImpl() {
        // TODO Auto-generated constructor stub
    }

}
```

- Implémenter les services suivants :

```
package ma.gov.formation.cigma.service;

import java.util.List;

import javax.ejb.Remote;

public interface IService {
    public void save(Article a);
    public Article getById(Integer id);
    public List<Article> getAll();
    public void remove(Integer id);
}
```

```
package ma.gov.formation.cigma.service;

import java.util.List;

import javax.ejb.EJB;
import javax.ejb.Stateless;

import ma.gov.formation.cigma.dao.IDao;

@Stateless
public class ServiceImpl implements IService {

    @EJB
    private IDao dao;

    @Override
    public void save(Article a) {
        dao.save(a);
    }

    @Override
    public Article getById(Integer id) {
        return dao.getById(id);
    }

    @Override
    public List<Article> getAll() {
        return dao.getAll();
    }

    @Override
    public void remove(Integer id) {
        dao.remove(id);
    }
}
```

- Modifier le fichier **persistence.xml** comme suit :

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1"
```

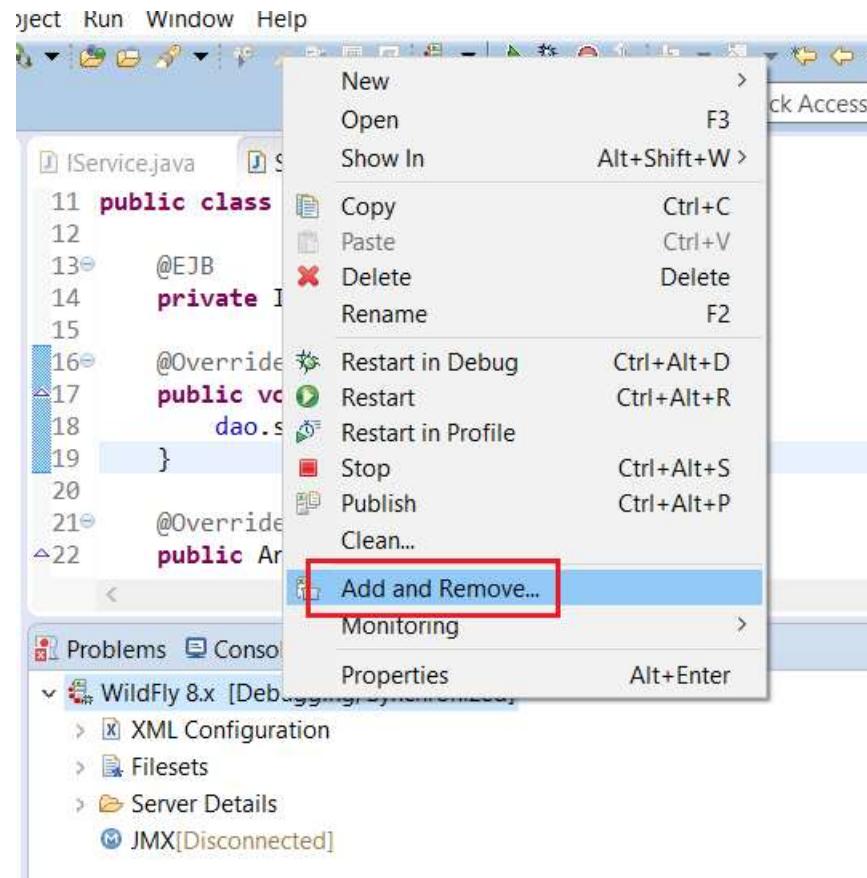
```

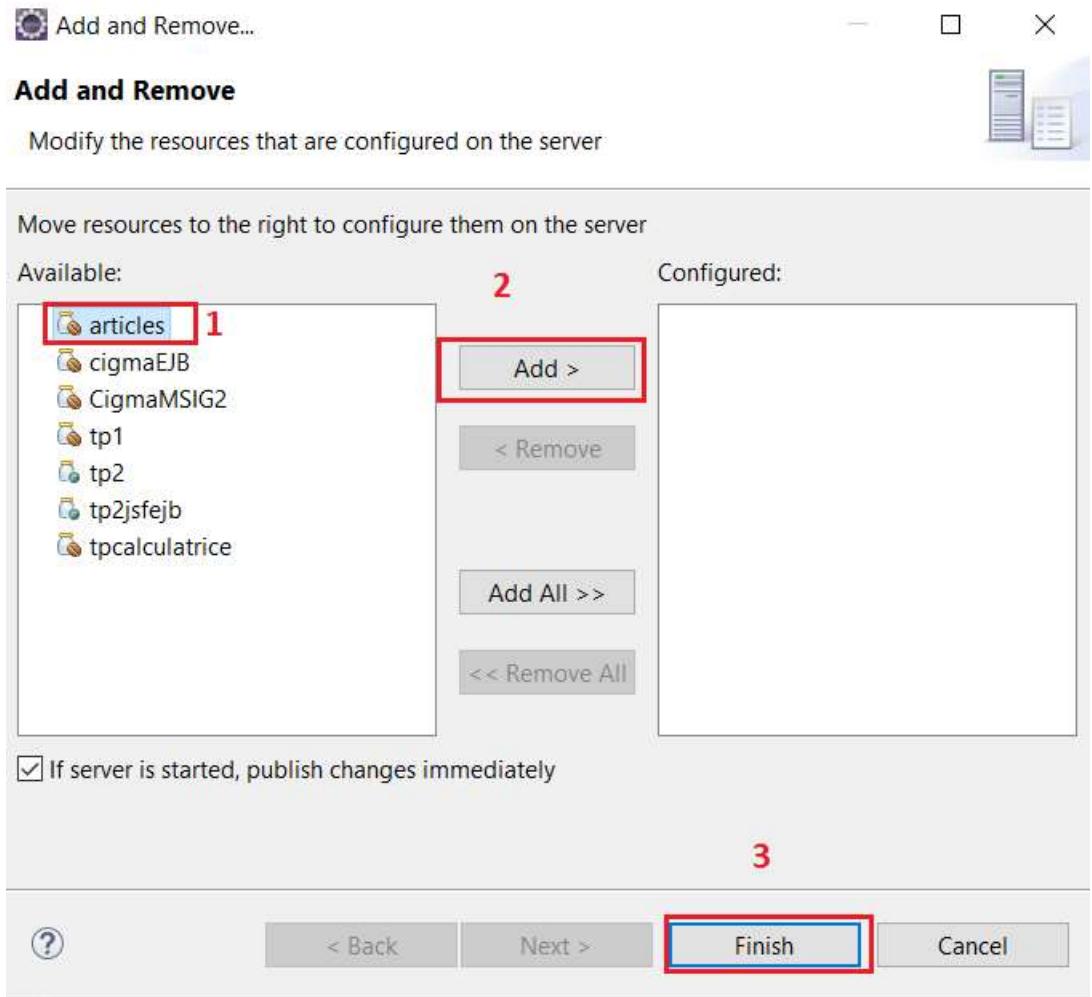
<?xml version="1.0" encoding="UTF-8" ?>
<persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
        http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
    <persistence-unit name="unite1" transaction-type="JTA">
        <jta-data-source>java:/ MySqlDS </jta-data-source>
        <properties>
            <property name="showSql" value="true" />
            <property name="hibernate.dialect"
                value="org.hibernate.dialect.MySQL5InnoDBDialect" />
            <property name="hibernate.hbm2ddl.auto" value="update" />
        </properties>
    </persistence-unit>
</persistence>

```

- Pour déployer le projet EJB "articles" dans WildFly, suivez les étapes suivantes :

Cliquez à droite sur le serveur et cliquez sur "**Add and Remove**" comme illustré au niveau de l'écran ci-après :





1. Sélectionner votre projet EJB.
2. Cliquer sur le bouton **Add>**.

Cliquez ensuite sur le bouton **Finish** :

Vérifier que les deux EJB (ServiceImpl et DaoImpl) sont bien déployés :

```
21:50:21,267 INFO [org.jboss.as.ejb3.deployment.processors.EjbJndiBindingsDeploymentUnitProcessor] (MSC service thread 1-7) JNDI bindings
java:global/articles/DaoImpl!ma.gov.formation.cigma.dao.IDao
java:app/articles/DaoImpl!ma.gov.formation.cigma.dao.IDao
java:module/DaoImpl!ma.gov.formation.cigma.dao.IDao
java:global/articles/DaoImpl
java:app/articles/DaoImpl
java:module/DaoImpl

21:50:21,267 INFO [org.jboss.as.ejb3.deployment.processors.EjbJndiBindingsDeploymentUnitProcessor] (MSC service thread 1-7) JNDI bindings
java:global/articles/ServiceImpl!ma.gov.formation.cigma.service.IService
java:app/articles/ServiceImpl!ma.gov.formation.cigma.service.IService
java:module/ServiceImpl!ma.gov.formation.cigma.service.IService
java:jboss/exported/articles/ServiceImpl!ma.gov.formation.cigma.service.IService
java:global/articles/ServiceImpl
java:app/articles/ServiceImpl
java:module/ServiceImpl

21:50:21,436 INFO [org.jboss.weld.deployer] (MSC service thread 1-1) JBAS016005: Starting Services for CDI deployment: articles.jar
```

- Vérifier également que le projet EJB (articles.jar) est bien déployé au niveau de la console d'administration de WildFly :

The screenshot shows the JBoss Application Server 7.1 deployment manager interface. The left sidebar has sections for Server Status, Configuration, JVM, Subsystem Metrics (Datasources, JPA, Transactions), Runtime Operations, Deployments (Manage Deployments, Webservices), and a navigation bar at the top with (0) Messages, Profile, and Runtime.

The main content area is titled "Deployments" and contains a table with the following data:

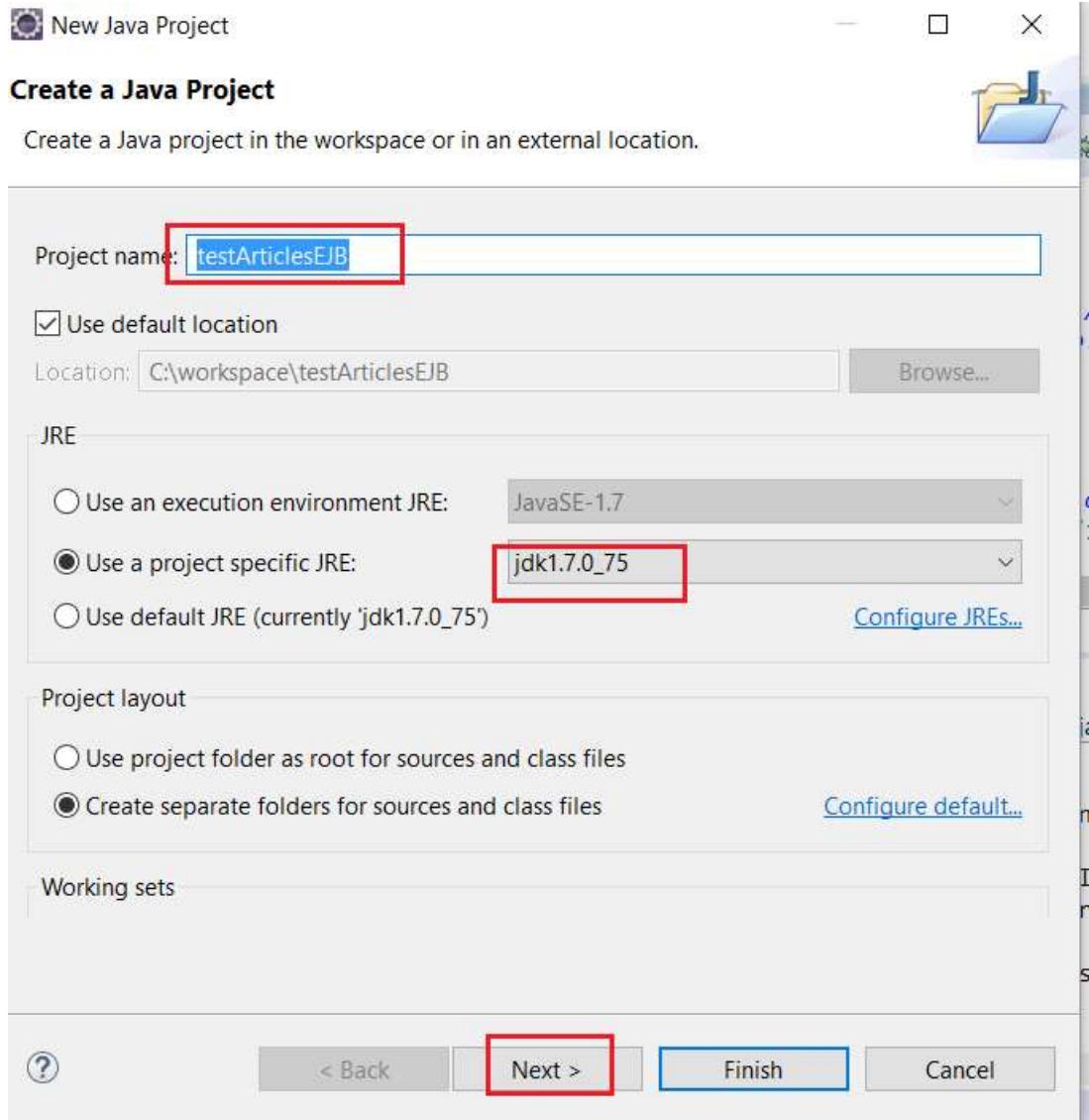
Name	Runtime Name	Enabled	En/Disable	Remove
articles.jar	articles.jar	✓	Disable	Remove

Below the table, it says "1-1 of 1".

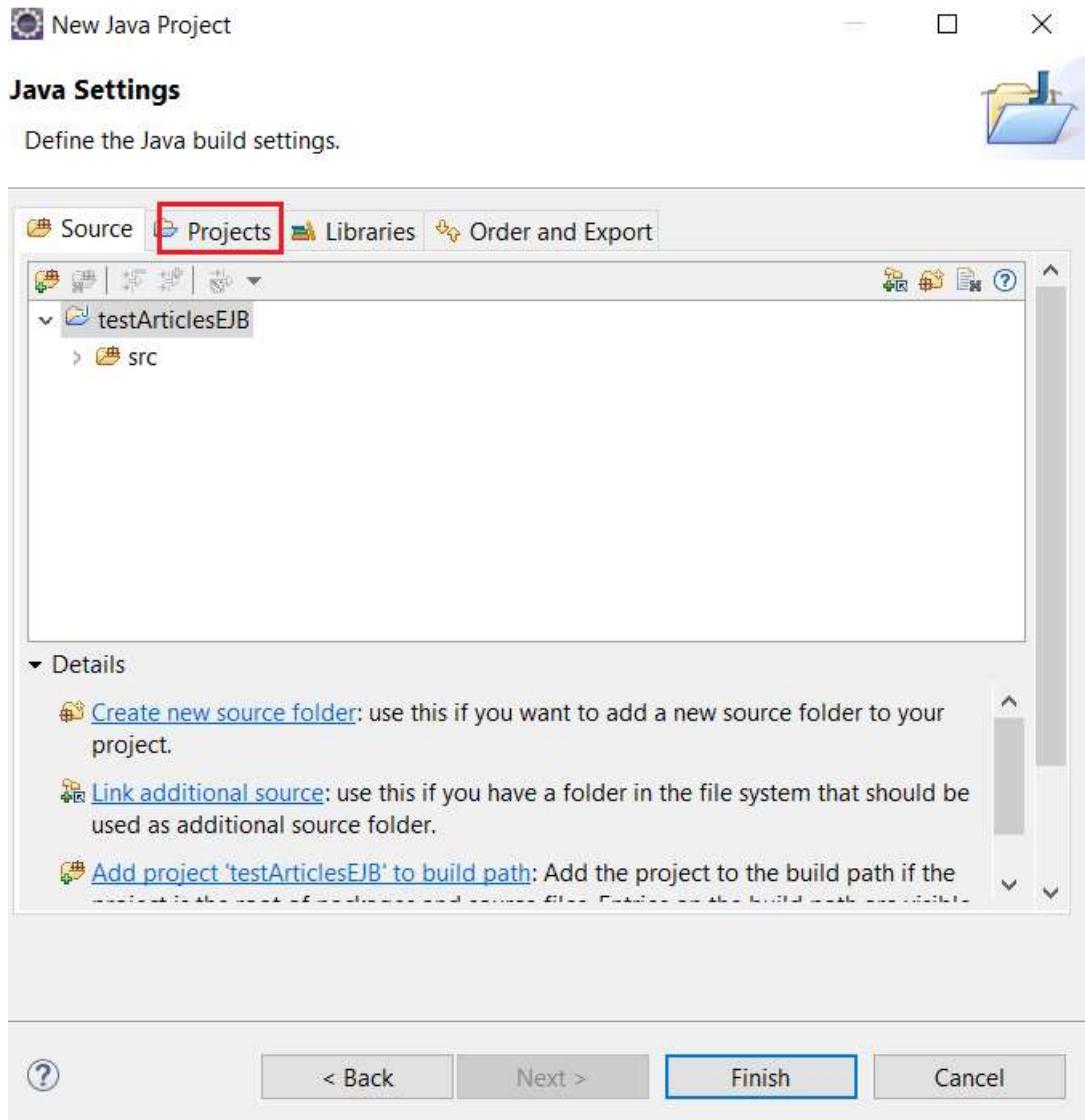
## 5. Tester l'application

Pour tester les services CRUD de l'EJB ServiceImpl, suivre les étapes suivantes :

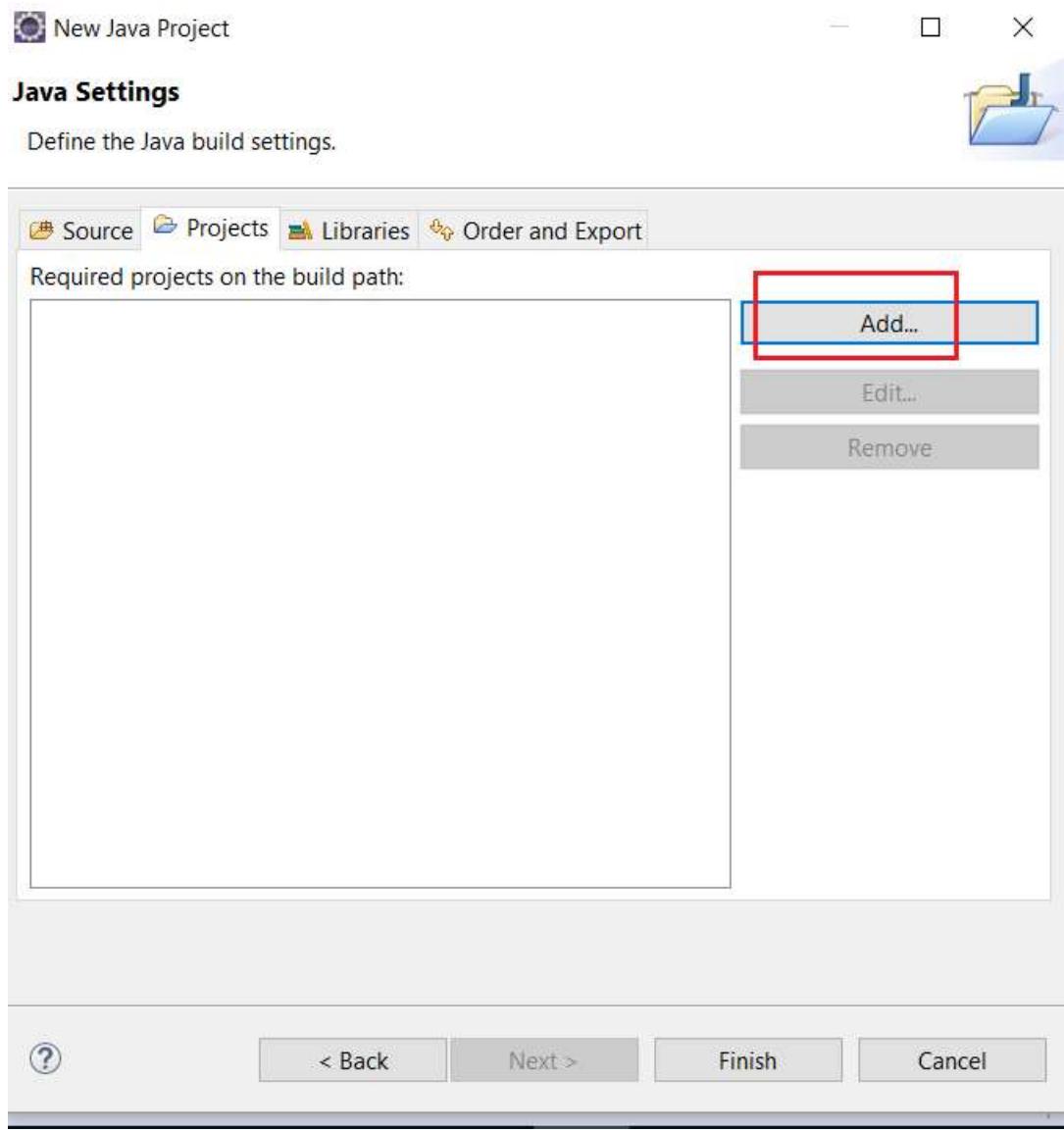
- Créer le projet java testArticlesEJB :



- Cliquer sur **Next >** :



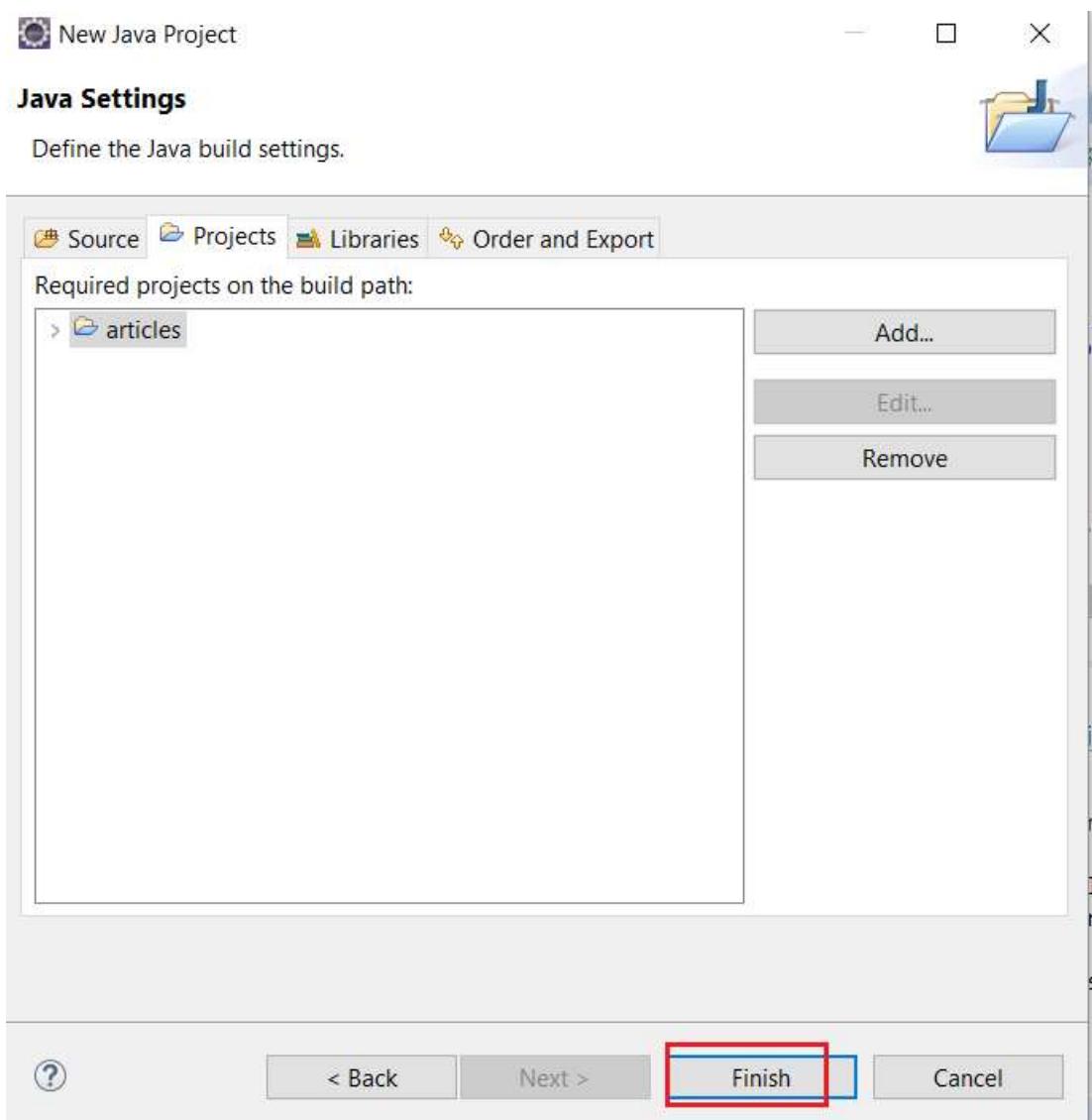
- Cliquer sur l'ongle **Projects** :



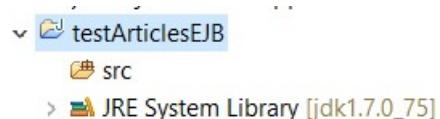
- Cliquer sur le bouton **Add...** :



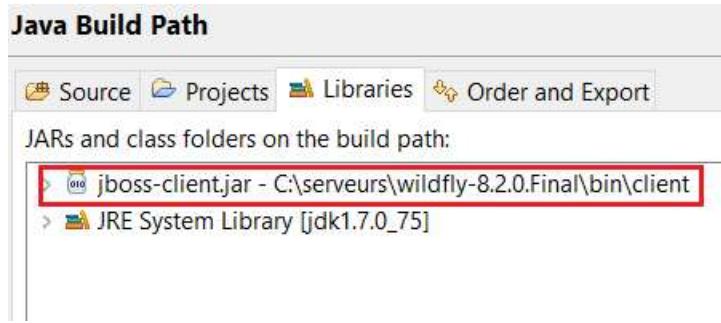
Cliquer sur le bouton **OK** :



Cliquer sur le bouton **Finish**. L'arborescence du projet suivante sera créée :



- Ajouter le jar jboss-client.jar (ce fichier se trouve dans le chemin [DOSSIER\_INSTALLATION\_JBOSS]\bin\client) comme illustré dans l'écran suivant :



- Créer le fichier ***jboss-ejb-client.properties*** :

```
remote.connections=default
remote.connection.default.host=localhost
remote.connection.default.port=8080
remote.connection.default.connect.options.org.xnio.Options.SASL_POLICY_NOANONYMOUS=false
remote.connection.default.connect.options.org.xnio.Options.SASL_POLICY_NOPLAINTEXT=false
remote.connection.default.connect.options.org.xnio.Options.SASL_DISALLOWED_MECHANISMS=${host.auth:jboss-LOCAL-USER}
remote.connection.default.username=user1
remote.connection.default.password=user1
```

- Créer la classe Test suivante :

```
package test;

import java.util.List;
import java.util.Properties;

import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;

import ma.gov.formation.cigma.service.Article;
import ma.gov.formation.cigma.service.IService;

public class Test {

    public static void main(String[] args) {
        InitialContext context = null;
        try {

            Properties props = new Properties();
            props.put(Context.URL_PKG_PREFIXES,
"org.jboss.ejb.client.naming");
            context = new InitialContext(props);
            String lookupName =
"ejb:/articles//ServiceImpl!ma.gov.formation.cigma.service.IService";
            System.out.println(lookupName);
            IService bean = (IService) context.lookup(lookupName);
            bean.save(new Article("TV LG", 7000.0, 20));
            bean.save(new Article("TV Sony", 10000.0, 15));
            bean.save(new Article("PC DELL", 1500.0, 70));
            bean.save(new Article("Switch", 1000.0, 100));
            List<Article> articles = bean.getAll();
        }
    }
}
```

```

        for (Article article : articles) {
            System.out.println(article);
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            context.close();
        } catch (NamingException e) {
            e.printStackTrace();
        }
    }
}

```

- Exécuter le résultat et vérifier que les articles ont été créés au niveau de la table Article :

Resultset 1

The screenshot shows a MySQL Workbench interface with a result set titled "Resultset 1". The SQL query area contains the command:

```
1 select *| FROM `basel`.`article`
```

The result set displays the following data:

	id	designation	prix	quantiteEnStock
1	23	TV LG	7000	20
	24	TV Sony	10000	15
	25	PC DELL	1500	70
	26	Switch	1000	100