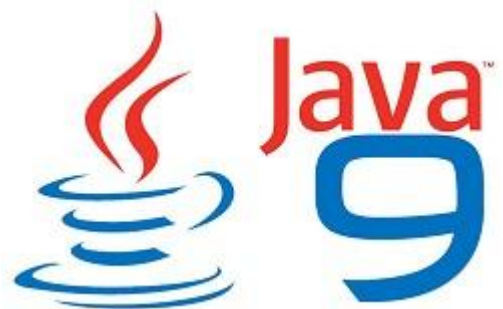


[Nous Contacter](#)

Tutoriel Java N°12 : accès base de données MySQL

[Accueil](#) / [200 tutos](#) / [Nos tutoriaux](#) / [java](#) / Tutoriel Java N°12 : accès base de données MySQL

Avec **Objis**, spécialiste Java depuis 2005, maîtrisez l'insertion et la récupération de données dans une base de données MySQL. Installez le Driver MySQL Connector, qui possède les implémentations pour la base MySQL des interfaces Connexion, Statement, ResultSet de l'API java.sql. Apprenez ensuite à Packager votre livraison pour la mise en production. Identifiez enfin les opportunités d'amélioration du code



ASTUCE -> [Soyez informé](#) de la mise à jour de nos tutos ou de la création de nouveaux tutos.





[+ Requêtes SQL](#) [200 Tutos](#) [Catalogue](#) [#10000Codeurs](#)

Connaissance de JAVA.

[Nous Contacter](#) 
Objectifs

Mettre en oeuvre l'API JDBC java.sql

Comprendre les 5 étapes clés pour l'insertion de données

Comprendre les 6 étapes clés pour la récupération de données

Liens utiles

[200 tutos java Objis](#)

[Objis, spécialiste formation java depuis 2005](#)

[Déploiement Jar sur Tutoriel Java officiel Oracle](#)

Durée

30min

Programme

Partie 1 : [création du projet](#)

Partie 2 : [ajout du driver JDBC MySql](#)





Partie 4 : [Insertion données en base](#)

+ [Accueil](#) [Clients](#) [200 Tutos](#) [Catalogue](#) [#10000Codeurs](#)

Partie 5 : [récupération de données](#)

Partie 6 : [création livraison en production \(sans manip MANIFEST.MF\)](#)

[Nous Contacter](#)

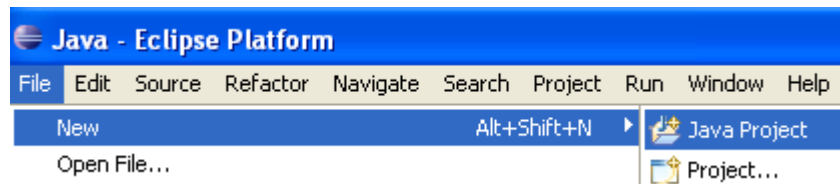


Partie 7 : [création livraison en production \(avec manip MANIFECST.MF\)](#)

Partie 8 : [optimisations](#)

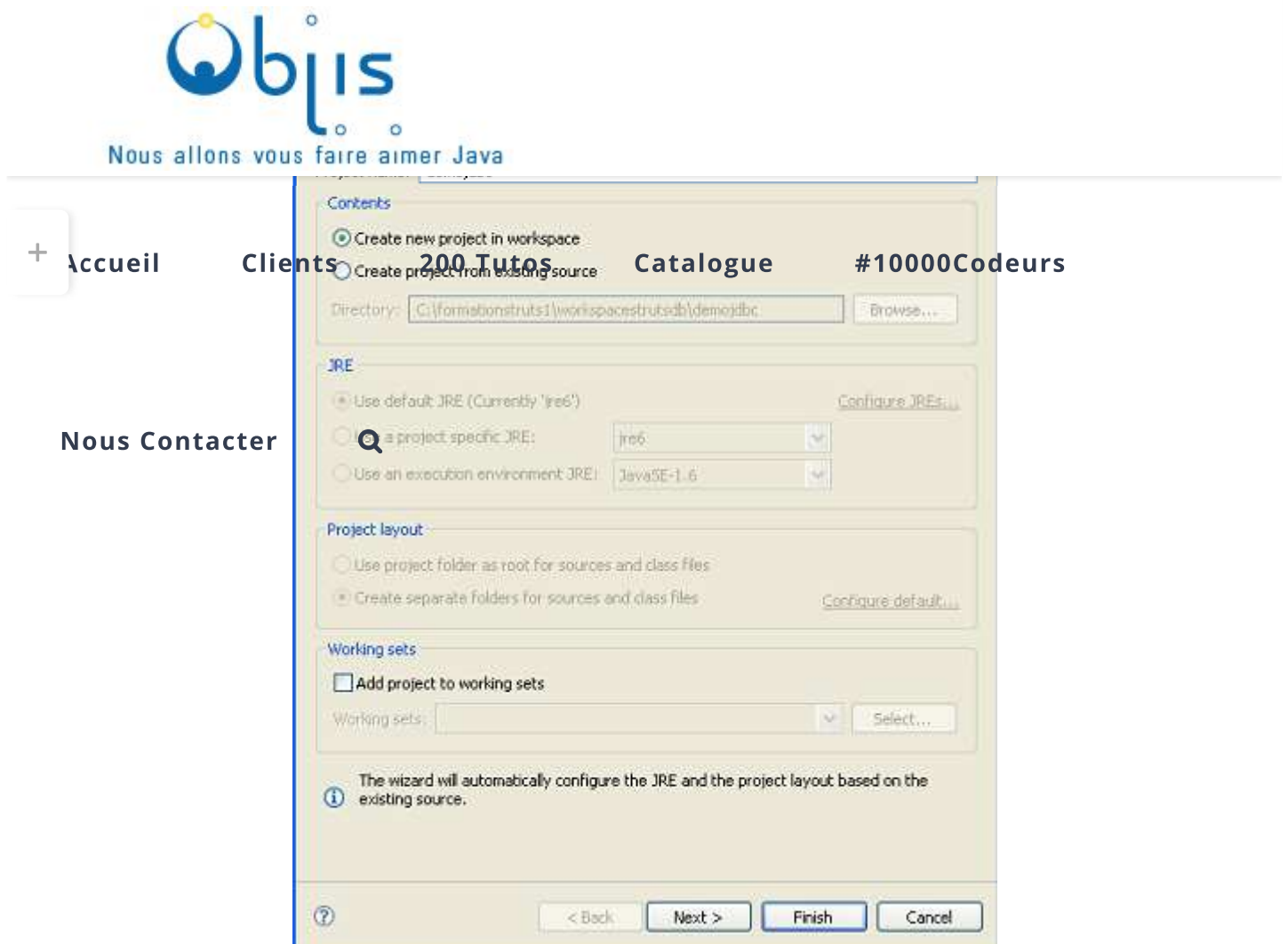
Partie 1 : projet eclipse demojdbc

A partir d'Eclipse (Standard ou Enterprise), Menu File->New



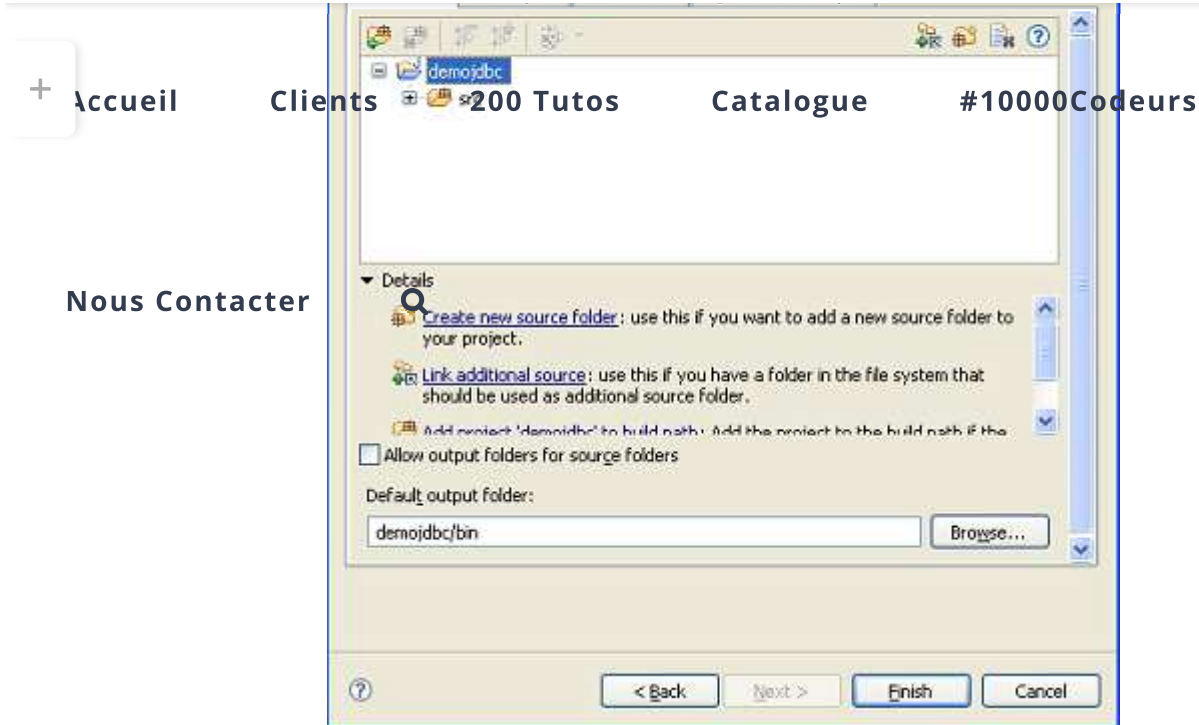
L'écran suivant apparaît.





Ajoutez le nom du projet (ici demojdbc) dans le champ 'Project Name', puis appuyez sur bouton 'Next'.

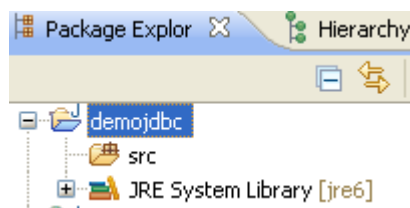
L'écran suivant apparaît.



Eclipse déposera les fichiers compilés (.class) dans le répertoire bin.

Cliquez sur 'Finish'

Le projet apparaît dans la vue 'Explorateur de package'.



Création package et classe

Cliquez droit sur src->new->Class



Nous allons vous faire aimer Java

+ Accueil	Open in New Window		Project...
	Open Type Hierarchy	F4	Package
	Show In	Alt+Shift+I	Class
	Copy	Ctrl+C	Interface
	Copy Qualified Name		Enum

L'écran suivant apparaît

New Java Class

Java Class
Create a new Java class.

Source folder: demojdbc/src Browse...

Package: com.objis.demojdbc Browse...

☐ Enclosing type: Browse...

Name: DemoJdbc

Modifiers: ☒ public ☐ default ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add...

Which method stubs would you like to create?

☒ public static void main(String[] args)

☐ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

? Finish Cancel

Entrez dans le champ Name le nom de classe : **DemoJdbc**

Entrez dans le champ Package le nom du package dans lequel la classe sera :
com.objis.demojdbc

Cochez la case 'public static void main()' afin de lancer l'application à partir de cette classe.



Nous allons vous faire aimer Java

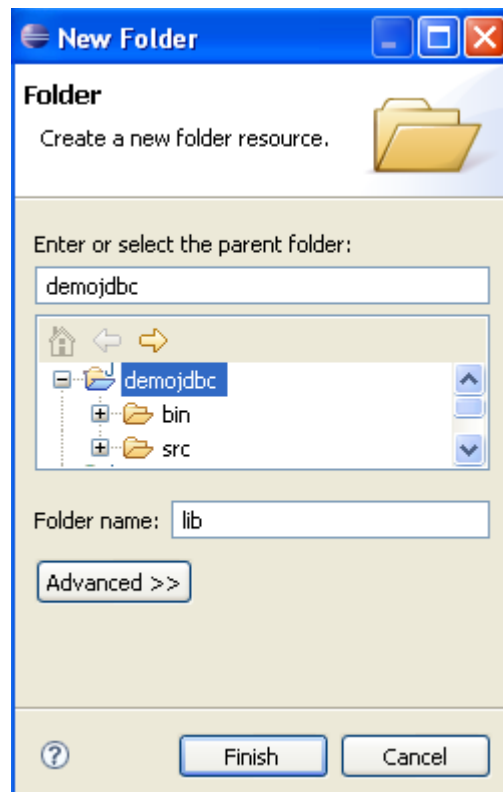
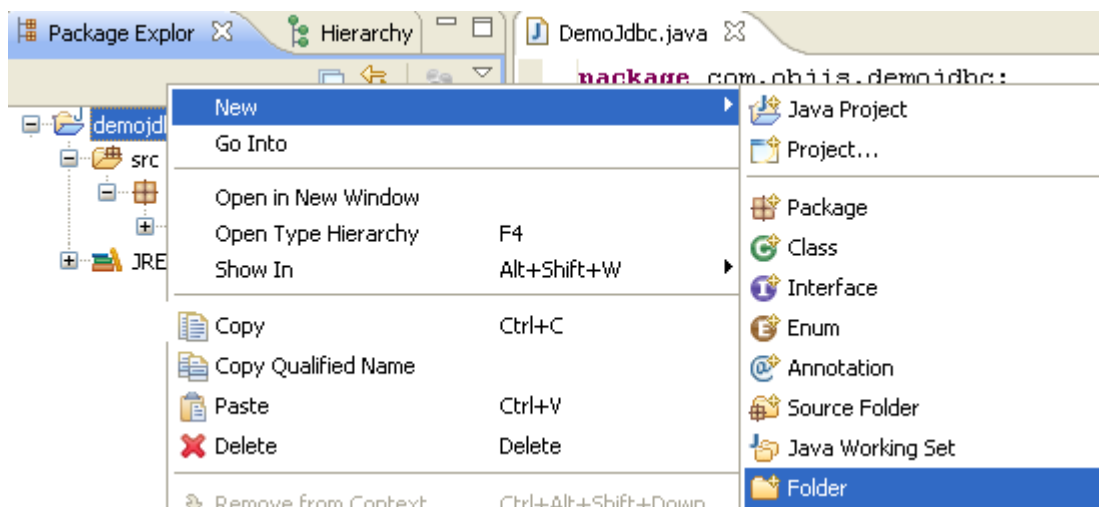
com.objis.demodb
DemoJdbc.java

+ Accueil Clients 200 Tutos Catalogue #10000Codeurs

Partie 2 : Ajout du driver JDBC

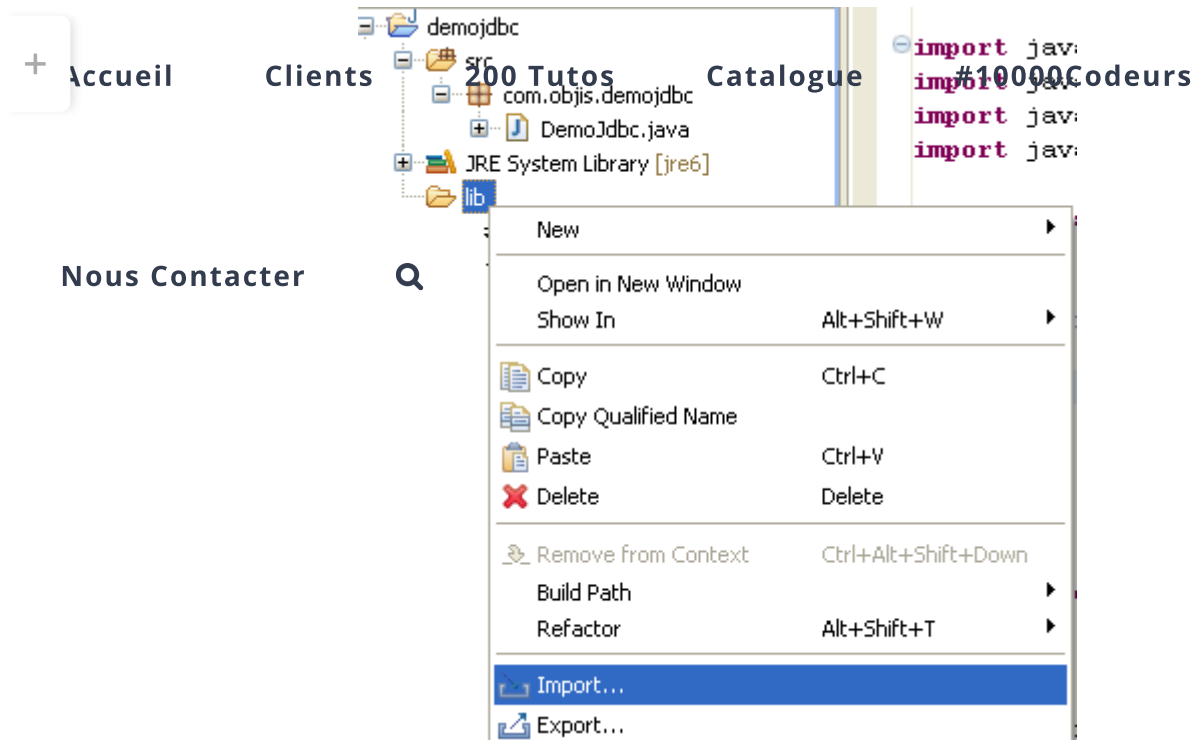
Librairie du projet (Driver JDBC) et CLASSPATH eclipse
Nous Contacter

Créez un répertoire lib à la racine du projet

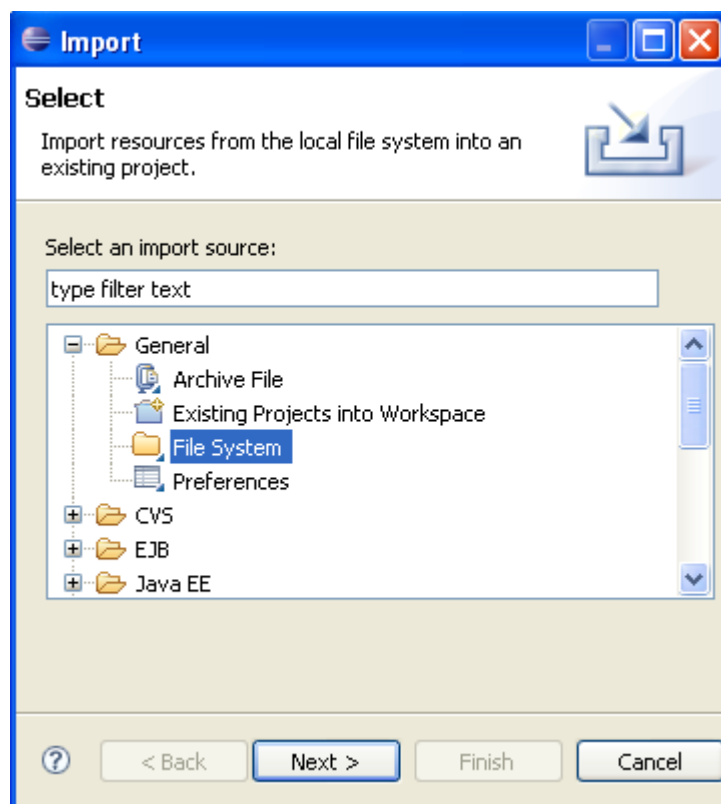




Nous allons vous faire aimer Java

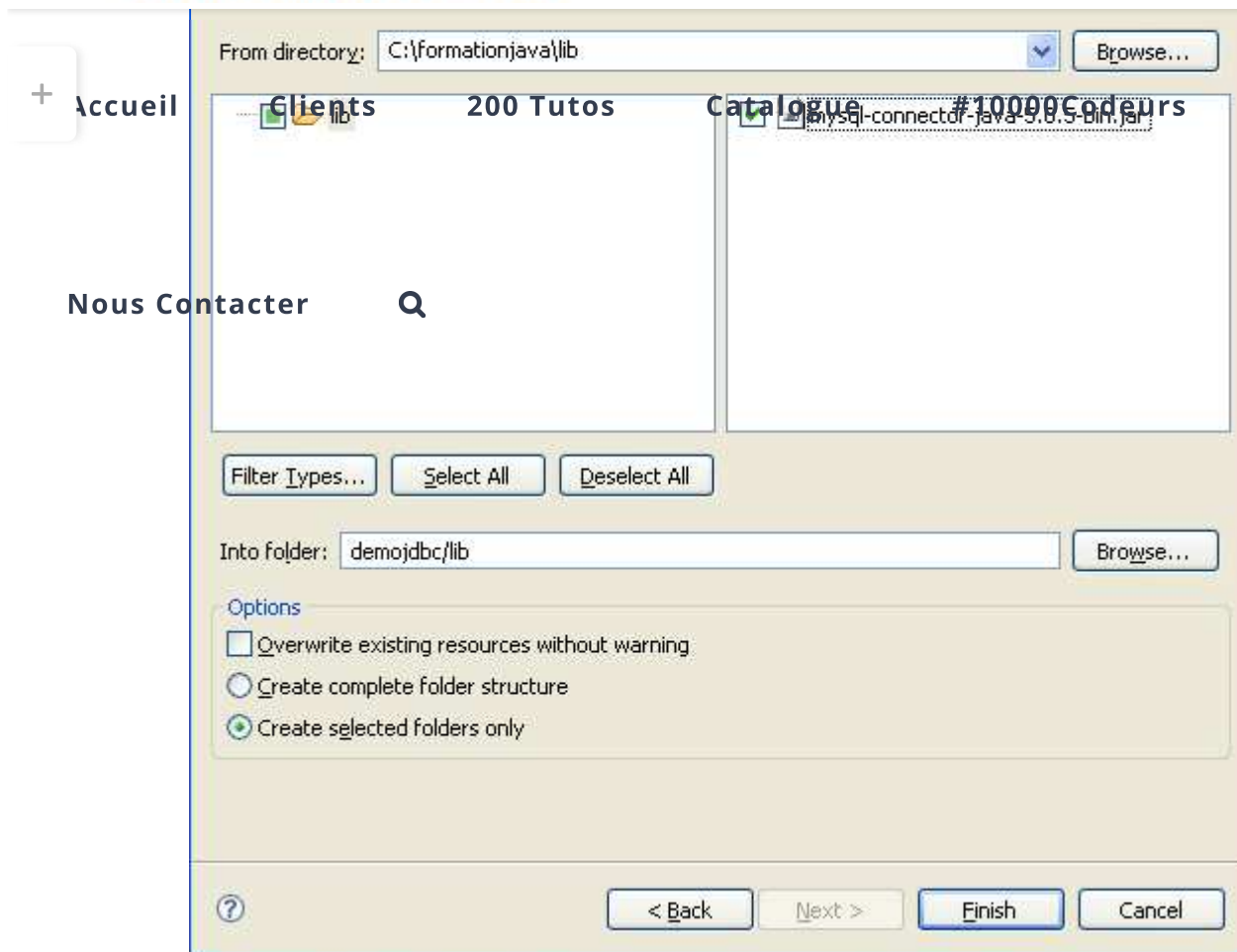


General/File System



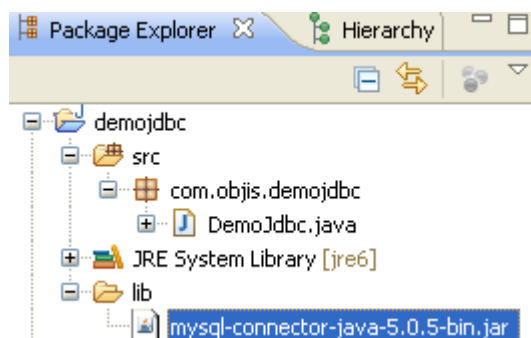
navigation dans votre système jusqu'à trouver le driver (ici mysql-connector-java-5.0.5.jar, récupéré suite au [dézipage de ZIP Driver](#))



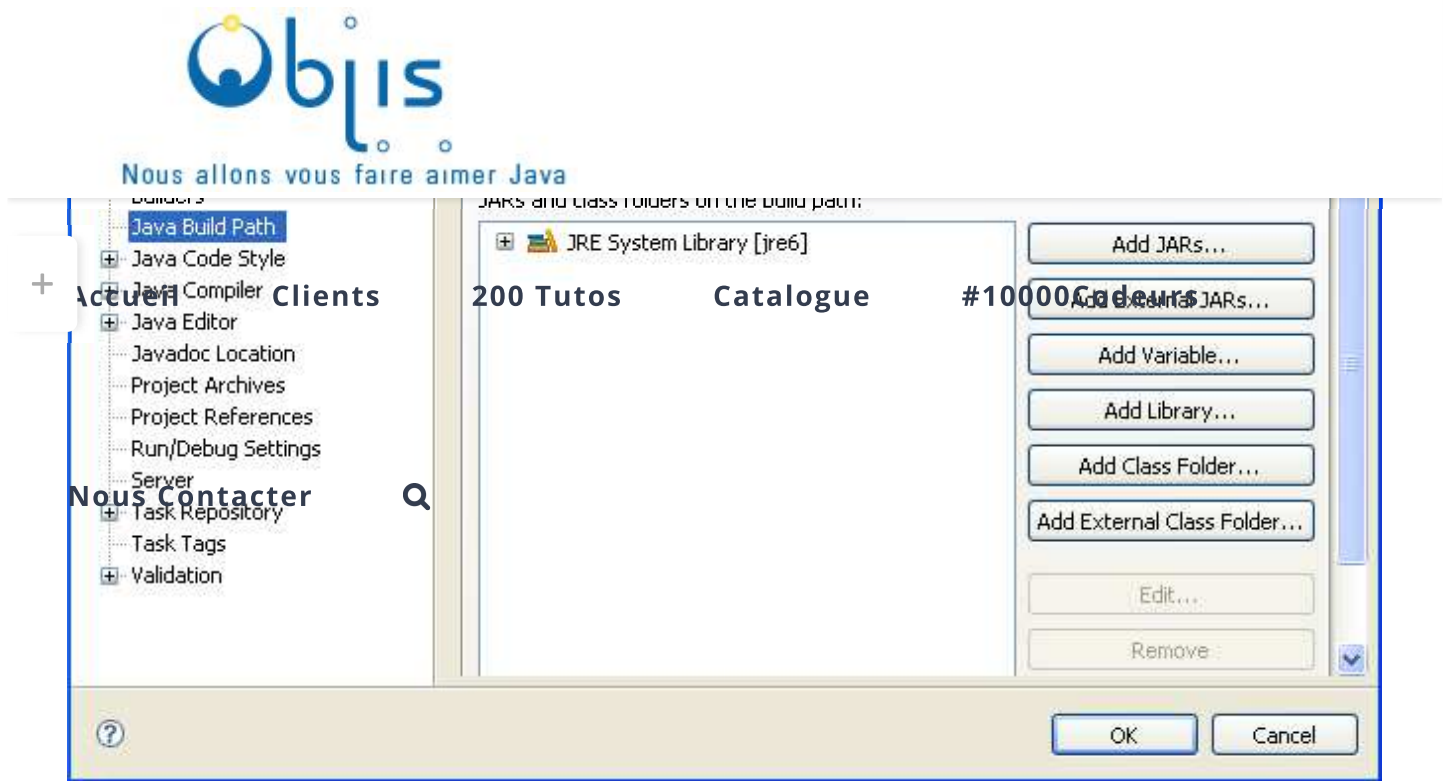


Cliquez sur Finish

Le driver apparaît dans la vue.

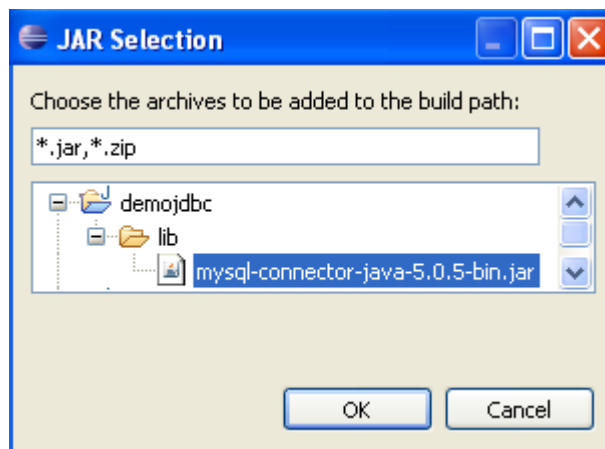


Informez Eclipse de l'existence de ce driver

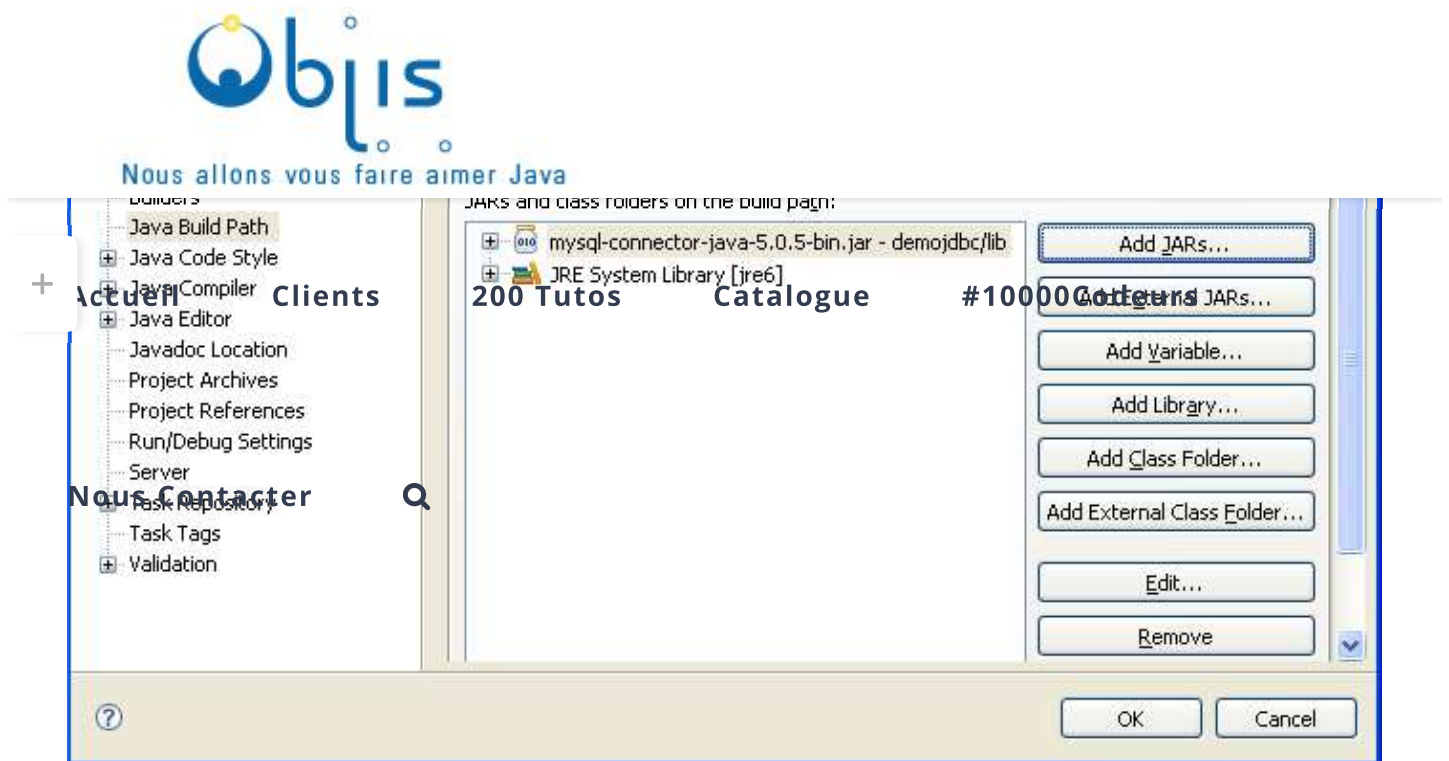


Cliquez-droit sur le projet->properties.

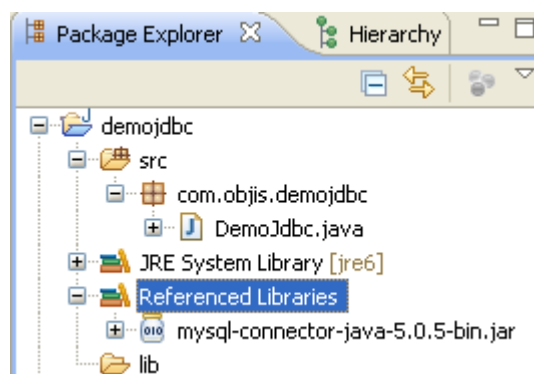
L'écran suivant apparaît.



Dans l'onglet 'Librairies' cliquez sur bouton 'Add Jars' car le driver est déjà dans le projet (Sinon il faudrait cliquer sur Add External Jars)



sélectionnez le driver puis OK



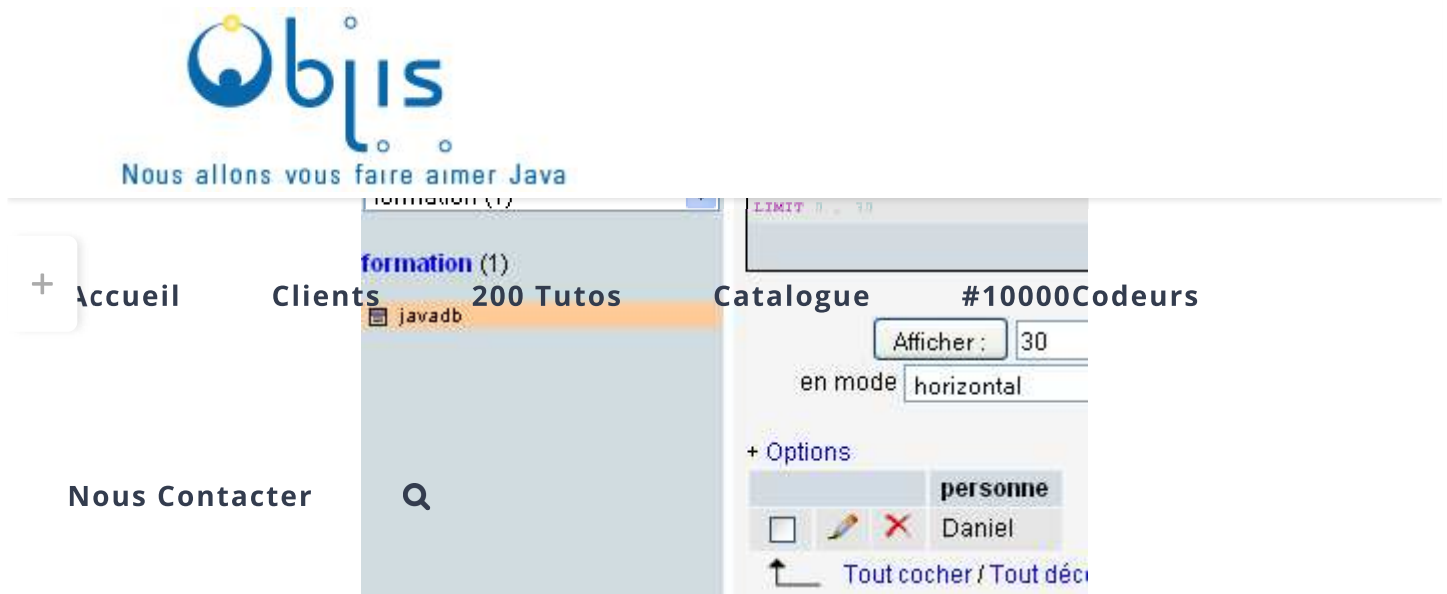
Le driver apparaît désormais dans 'Referenced librairies'.

Partie 3 : création table

En utilisant le client [PhpMyAdmin](#), créer :

- Une base de données 'formation'
- une table 'javadb' possédant une seule colonne 'personne'

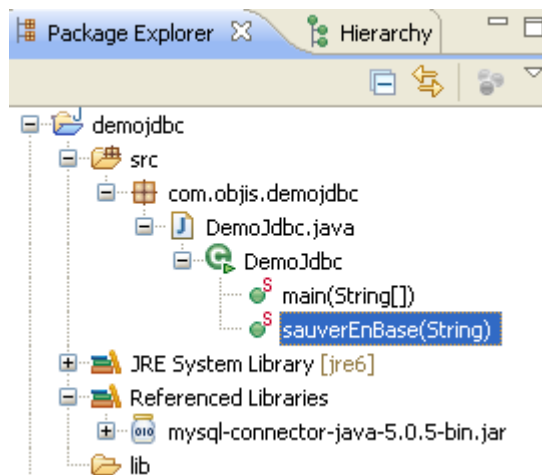




Expliquez

Partie 4 : insertion de données

Code d'accès aux données



Créez une méthode sauverEnBase() suivante, qui prend en paramètre une chaîne de caractère à insérer en base de données.





Nous allons vous faire aimer Java

Statement st = null;

```
try {
    // Etape 1 : Chargement du driver
    Class.forName("com.mysql.jdbc.Driver");
    // Etape 2 : récupération de la connexion
    cn = DriverManager.getConnection(url, login, passwd);
    // Etape 3 : Création d'un statement
    st = cn.createStatement();
    String sql = "INSERT INTO `javadb` (`personne`) VALUES ('"+ personne + "')";
```

Nous Contacter

```
// Etape 4 : exécution requête
st.executeUpdate(sql);
```

```
} catch (SQLException e) {
    e.printStackTrace();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
} finally {
    try {
        // Etape 5 : libérer ressources de la mémoire.
        cn.close();
        st.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

Cette méthode statique sera appelée à partir de la méthode main() et donc au lancement de l'application.

QUESTION 1 : Dans quel package se trouve Connection ? Est-ce une Classe ou une interface ?

QUESTION 2 : à quoi sert le code : Class.forName ?

QUESTION 3 : à quoi sert le 'try / catch' ?

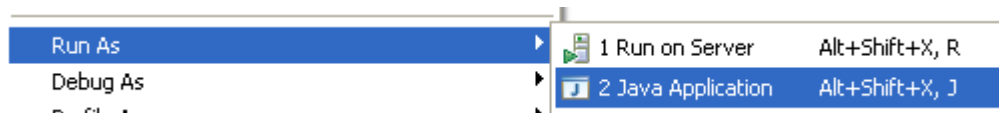
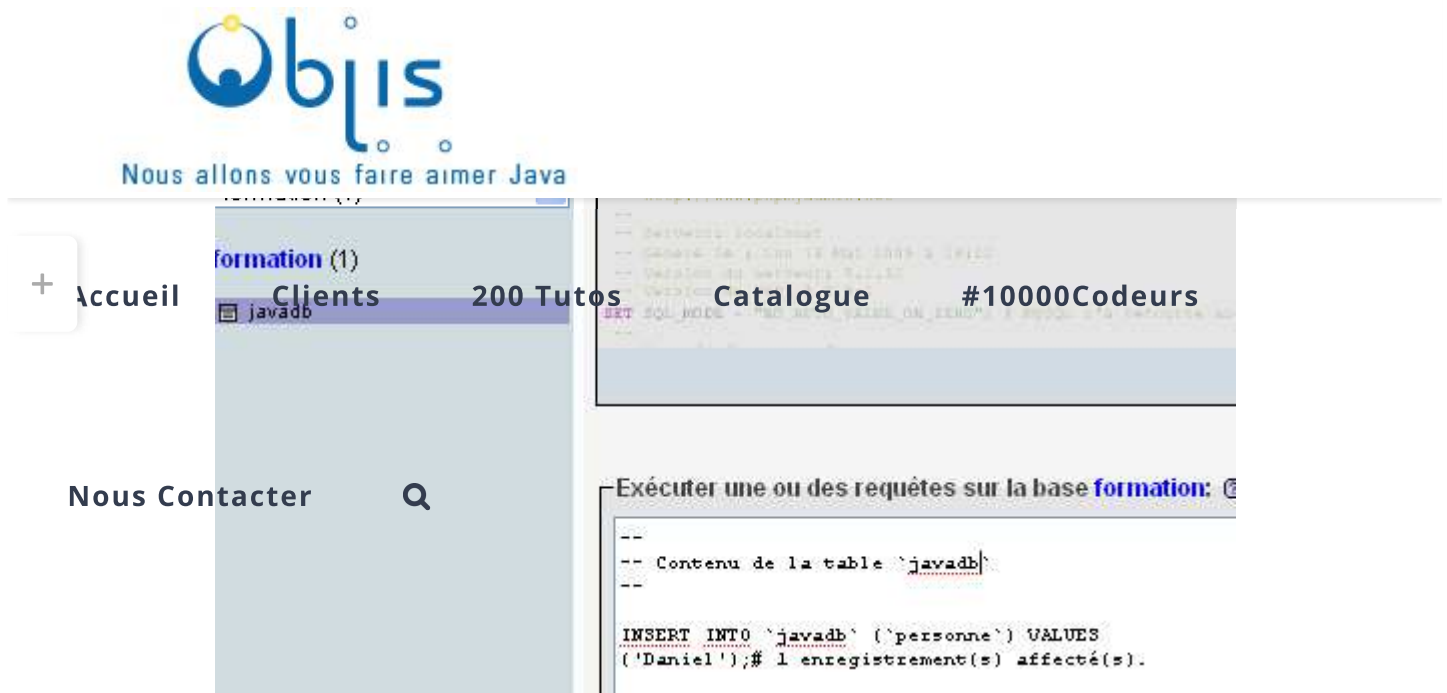
QUESTION 4 : y a t'il une autre façon de gérer les erreurs potentielles ?

QUESTION 5 : à quoi sert le bloc 'finally' ?

```
public static void main(String[] args) {
    sauverEnBase("Jean DUPONT");
}
```

cliquez-droit sur la classe DemoJdbc->Run As->Java Application.





La données a bien été insérée en base de données.

Partie 5 : lecture de données

Ci-dessous le code de récupération de données





Nous allons vous faire aimer Java

+ Accueil Clients 200 Tutos Catalogue #10000Codeurs

Nous Contacter

```

Connection cn = null; Statement st = null; ResultSet rs = null;
try {
    // Etape 1 : Chargement du driver
    Class.forName("com.mysql.jdbc.Driver");
    // Etape 2 : récupération de la connexion
    cn = DriverManager.getConnection(url, login, passwd);
    // Etape 3 : Création d'un statement
    st = cn.createStatement();
    String sql = "SELECT * FROM javadb";
    // Etape 4 : exécution requête
    rs = st.executeQuery(sql);
    // Etapes 5 (parcours ResultSet)
    while (rs.next()) {
        System.out.println(rs.getString("personne"));
    }
} catch (SQLException e) {
    e.printStackTrace();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
} finally {
    try { // Etape 6 : libérer ressources de la mémoire.
        cn.close();
        st.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}

```

Résultat

```

<terminated> DemoJdbc [Java Application] C:\Program Files\Ja
Daniel
Jean DUPONT

```

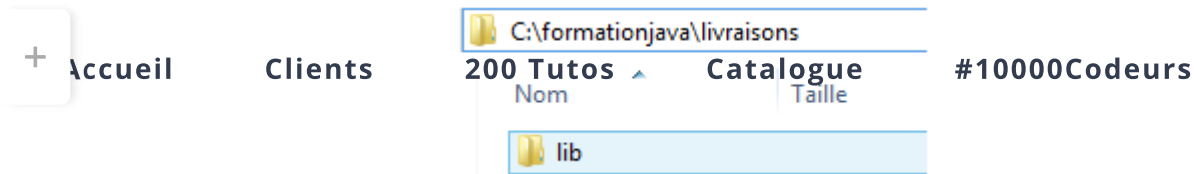
Nous avons récupérés les informations en base de données.

Code du projet

[demo java jdbc](#)

Partie 6 : livraison en production (sans manipulation du Manifest)

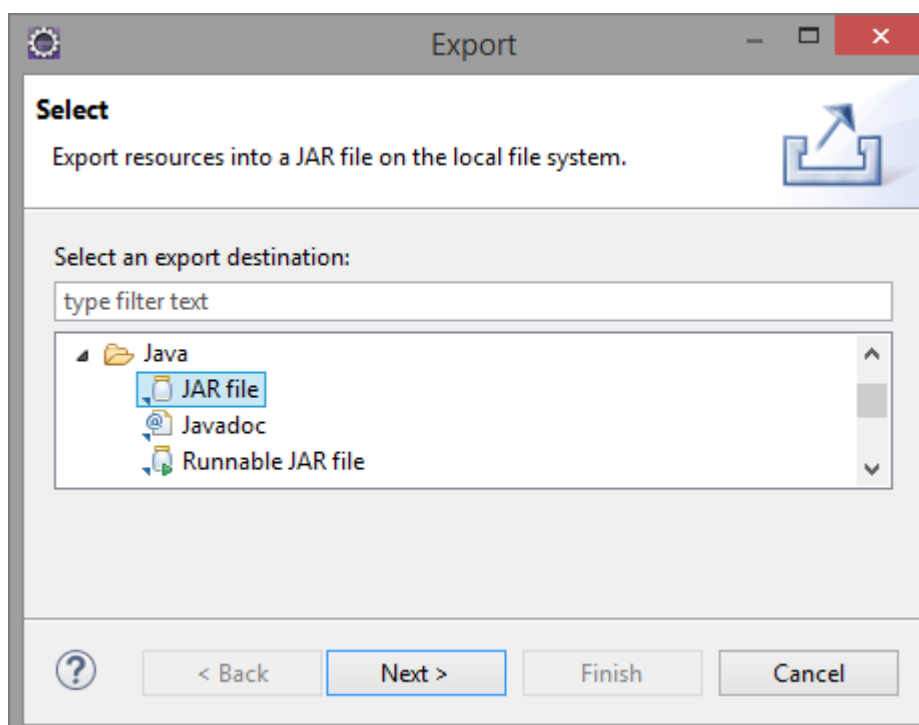




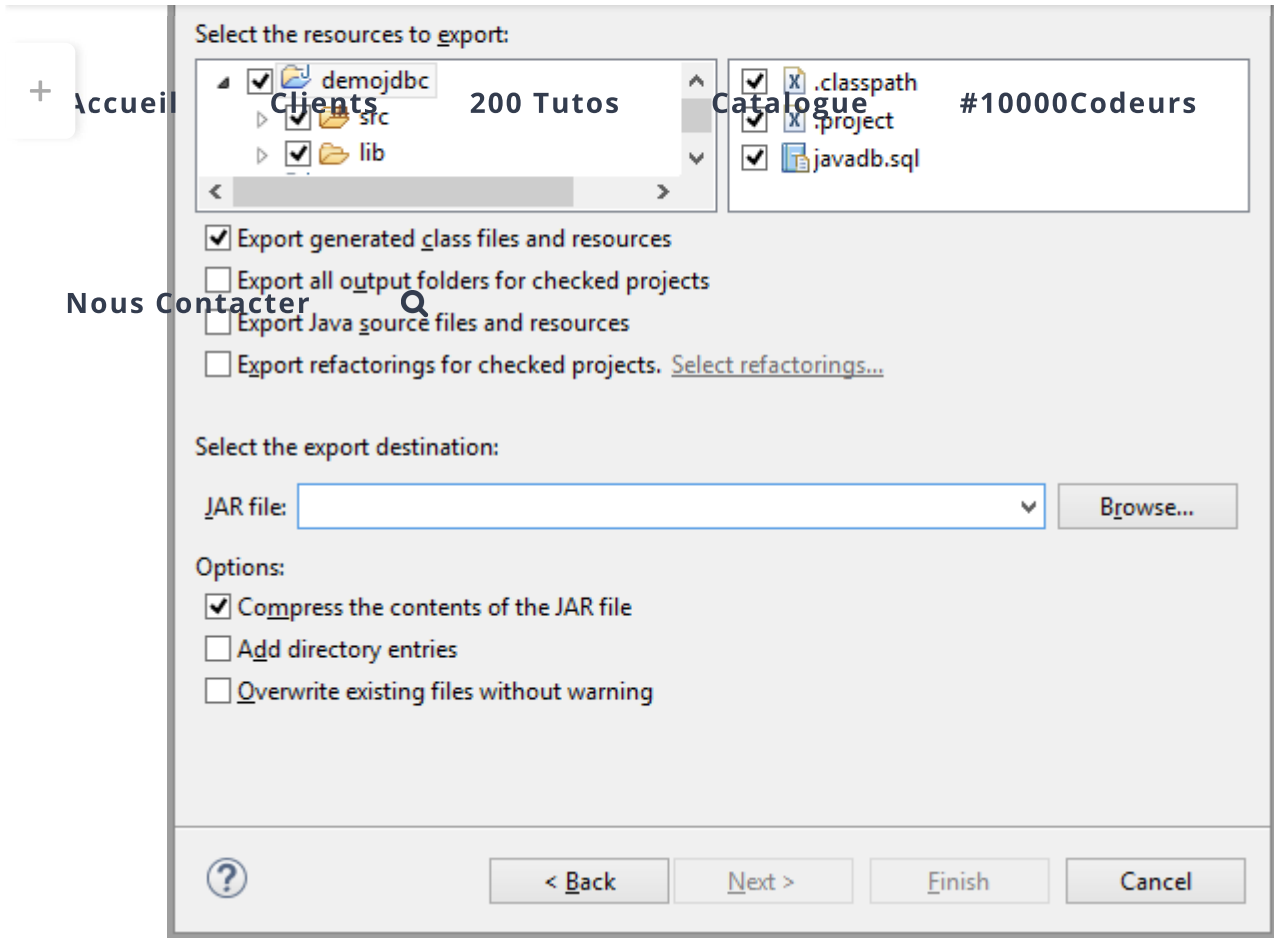
En appuyant sur la création de livrable, créer dans ce répertoire livraison le livrable demojdbc.jar

Cliquer droit sur le projet dans eclipse

Export



L'écran suivant apparaît



Notez que par défaut vous vous apprêtez donc à créer un livrable contenant : le source compilé (ça on en a besoin), le driver (on l'a déjà), et fichier eclipse .project et .classpath (on en a pas besoin en production !)

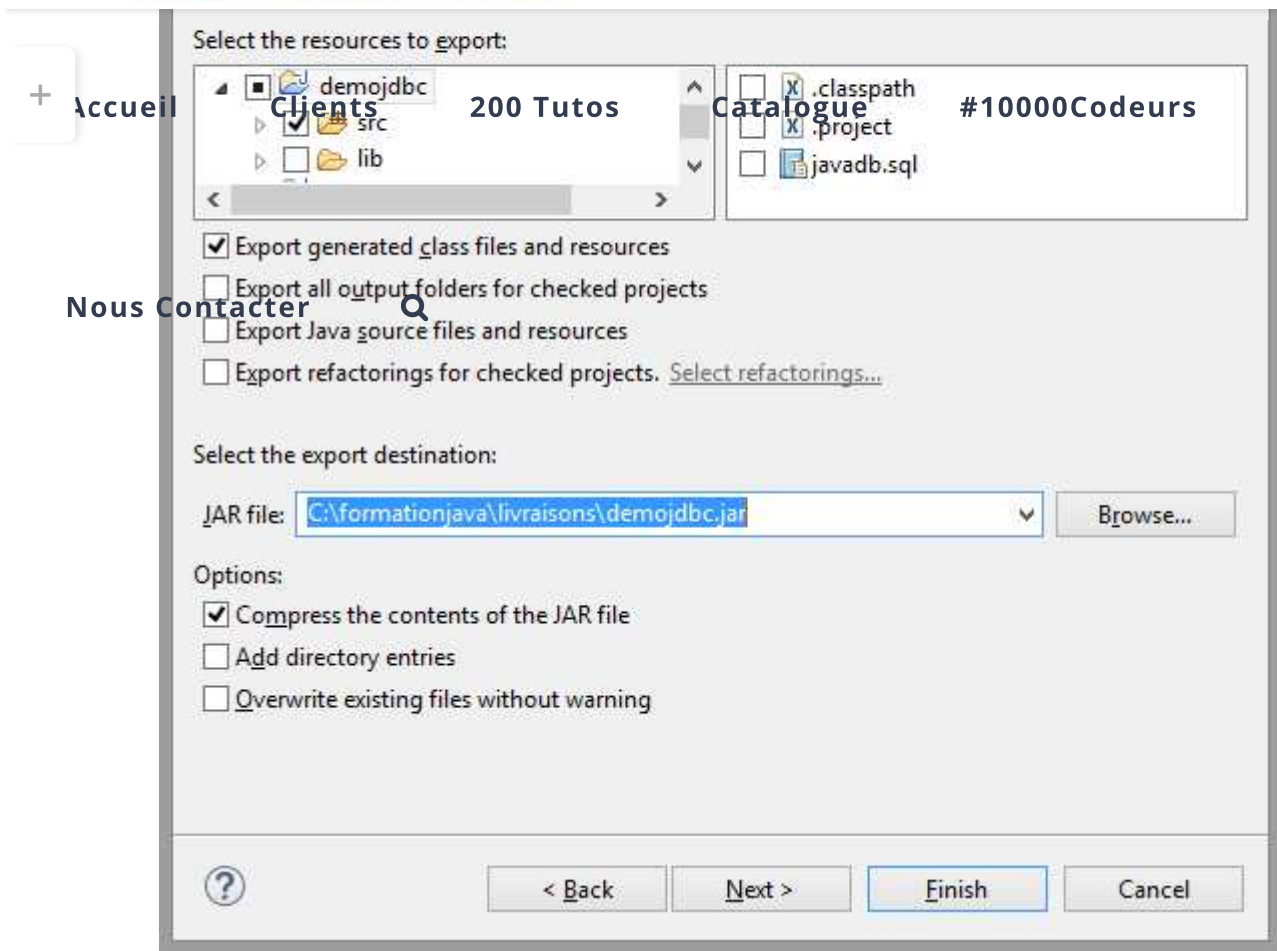
Spécifiez le répertoire dans lequel vous allez créer le livrable aussi.

Décochez tout ce qui n'est pas strictement lié au code source.

L'écran devient :



Nous allons vous faire aimer Java



Next



Nous allons vous faire aimer Java

+ Accueil Clients 200 Tutos Catalogue #10000Codeurs

Nous Contacter

Select options for handling problems:

- ☒ Export class files with compile errors
- ☒ Export class files with compile warnings
- ☐ Create source folder structure
- ☒ Build projects if not built automatically
- ☐ Save the description of this JAR in the workspace

Description file:

Next

JAR Export

JAR Manifest Specification

Customize the manifest file for the JAR file.

Specify the manifest:

- ☒ Generate the manifest file
 - ☐ Save the manifest in the workspace
 - ☐ Use the saved manifest in the generated JAR description file
- ☐ Use existing manifest from workspace

Manifest file:

Manifest file:

Next





+ Notez la petite taille du livrable. C'est une bonne pratique d'essayer d'avoir toujours un livrable de la taille la plus petite possible (on met juste ce qu'il faut).

Lancez la commande suivante à partir du répertoire livraisons : `java -cp demojdbc.jar`

Nous Contacter 
com.objis.demos.jdbc.DemoJdbc

Expliquer l'option **-cp**

```
C:\formationjava\livraisons>java -cp demojdbc.jar com.objis.demos.jdbc.DemoJdbc
java.lang.ClassNotFoundException: com.mysql.jdbc.Driver
    at java.net.URLClassLoader.findClass(Unknown Source)
    at java.lang.ClassLoader.loadClass(Unknown Source)
    at sun.misc.Launcher$AppClassLoader.loadClass(Unknown Source)
    at java.lang.ClassLoader.loadClass(Unknown Source)
    at java.lang.Class.forName0(Native Method)
    at java.lang.Class.forName(Unknown Source)
    at com.objis.demos.jdbc.DemoJdbc.sauverEnBase(DemoJdbc.java:28)
    at com.objis.demos.jdbc.DemoJdbc.main(DemoJdbc.java:12)
Exception in thread "main" java.lang.NullPointerException
    at com.objis.demos.jdbc.DemoJdbc.sauverEnBase(DemoJdbc.java:52)
    at com.objis.demos.jdbc.DemoJdbc.main(DemoJdbc.java:12)

C:\formationjava\livraisons>
```

Expliquez

Lancez la commande suivante : `java -cp demojdbc.jar;lib/mysql-connector-java-5.0.5-bin.jar com.objis.demos.jdbc.DemoJdbc`

```
C:\formationjava\livraisons>java -cp demojdbc.jar;lib/mysql-connector-java-5.0.5-bin.jar com.objis.demos.jdbc.DemoJdbc
Daniel
Jean DUPONT
C:\formationjava\livraisons>
```

Partie 7 : livraison en production (avec manipulation du Manifest)

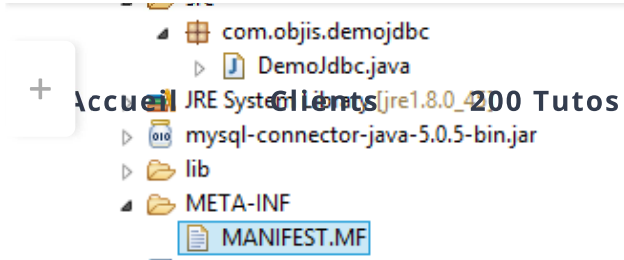
Créer à la racine du projet un répertoire 'META-INF'

Dans ce répertoire, créer un fichier texte MANIFEST.MF suivant





Nous allons vous faire aimer Java



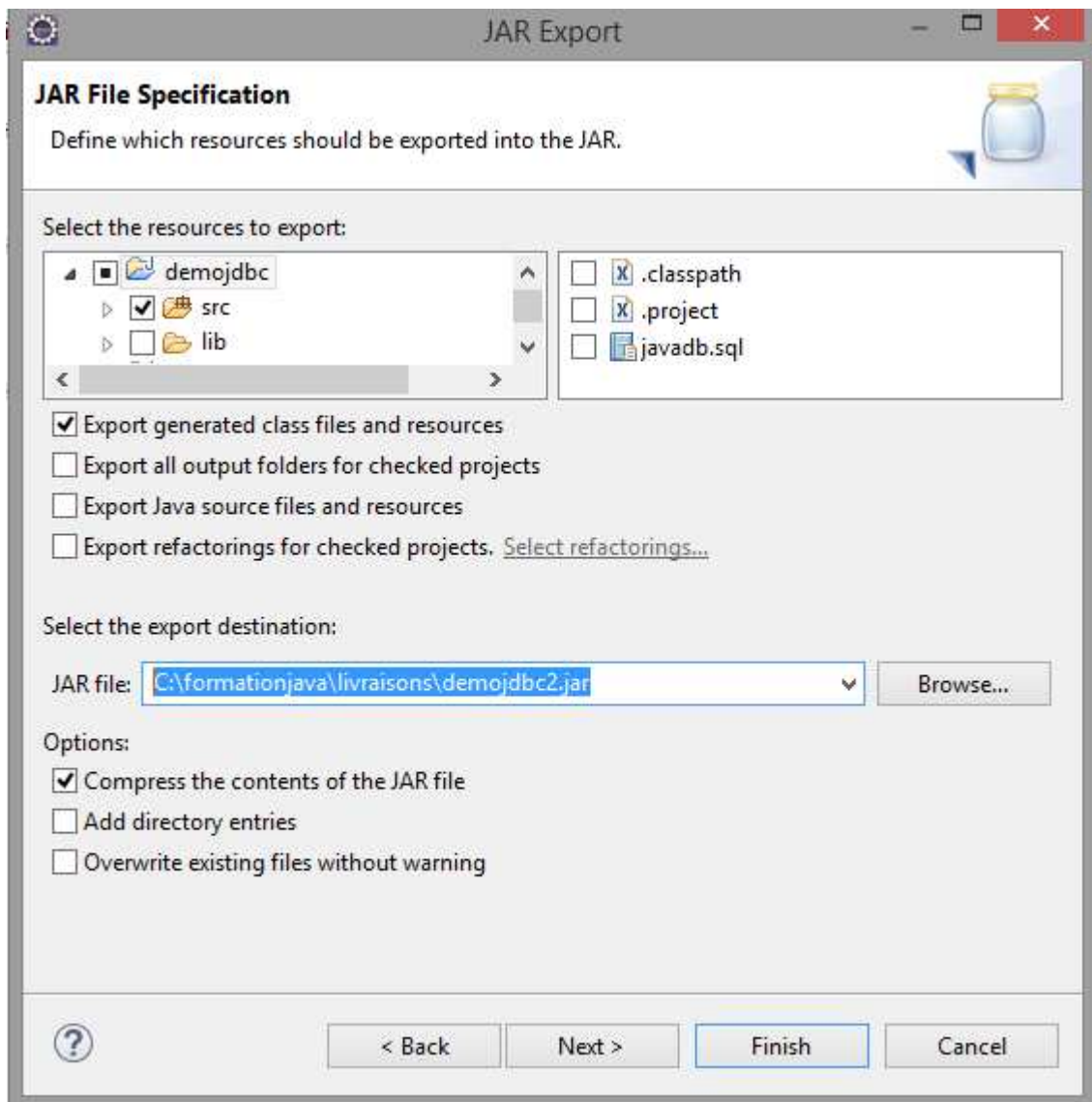
Catalogue

#10000Codeurs

Nous Contacter



Expliquer les 4 lignes du fichier





Nous allons vous faire aimer Java

+ Accueil

Select options for handling problems:

- ☒ Export class files with compile errors
- ☒ Export class files with compile warnings

☐ Create source folder structure☒ Build projects if not built automatically☐ Save the description of this JAR in the workspace

Description file:

Browse...



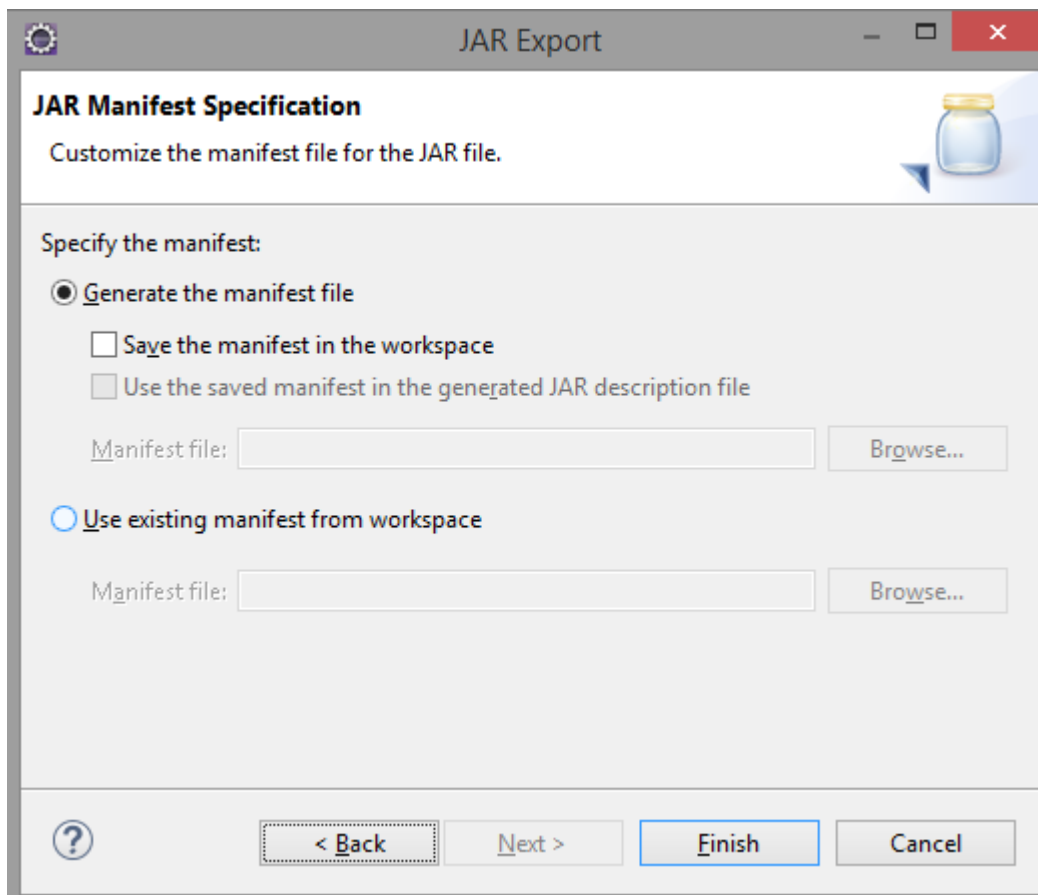
< Back

Next >

Finish

Cancel

Nous Contacter





Nous allons vous faire aimer Java

+ Accueil

Clients

200 Tutos

Catalogue

#10000Codeurs

Nous Contacter

Specify the manifest:

☐ Generate the manifest file

☐ Save the manifest in the workspace

☐ Use the saved manifest in the generated JAR description file

Manifest file:

Browse...

☒ Use existing manifest from workspace

Manifest file:

Browse...

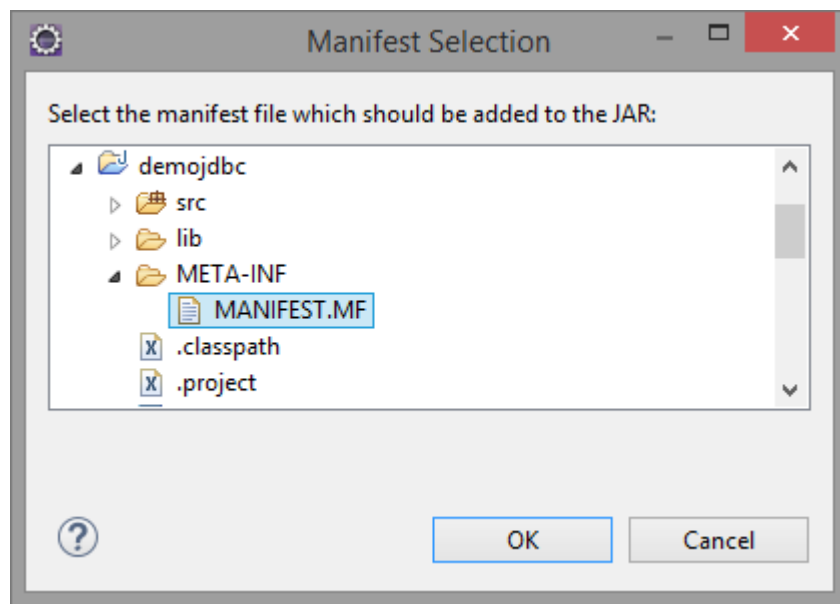


< Back

Next >

Finish

Cancel





Nous allons vous faire aimer Java

[+ Accueil](#)
[Clients](#)
[200 Tutos](#)
[Catalogue](#)
[#10000Codeurs](#)

Nous Contacter

Specify the manifest:

☐ Generate the manifest file

☐ Save the manifest in the workspace

☐ Use the saved manifest in the generated JAR description file

Manifest file:

☒ Use existing manifest from workspace

Manifest file:

Seal contents:

☐ Seal the JAR

☒ Seal some packages Nothing sealed

Select the class of the application entry point:

Main class:

C:\formationjava\livraisons	
Nom	Taille
lib	
demojdbc.jar	3 Ko
demojdbc2.jar	3 Ko

Vous venez de créer une deuxième version de votre livrable. L'intérêt ici est que vous allez pouvoir lancer l'application uniquement avec la commande : **java -jar demojdbc2.jar**

```
C:\formationjava\livraisons>java -jar demojdbc2.jar
Daniel
Jean DUPONT
```





+ [Livraison avec des bases de données manipulées](#) [200 Formes](#) [MANIFESTA](#) [Flugue](#) [#10000Codeurs](#)

[demojdbc-deploiement-jar](#)

[Nous Contacter](#)

Partie 8 : optimisations

Question : qu'est ce qu'un 'PreparedStatement' ? Quelle différence avec un Statement ?

Proposez un refactoring du code en utilisant les PreparedStatements. Quel avantage ?

Proposez une meilleure stratégie de gestion des exceptions. En particulier, créez un package java 'exceptions' et créez une exception 'DataBaseException' héritant de la classe Exception et dont la méthode getMessage() sera redéfinie pour renvoyer le message 'ERREUR ACCES BASE DE DONNEES'

Coté sécurité, citez au moins une faille à laquelle ce code s'expose. Comment s'en prémunir ?

Conclusion

Dans ce tutoriel, vous avez découvert les étapes clés de connexion à une base de données relationnelle avec JAVA.

[formation.sql.](#)

Professionnels : maîtrisez les nouveautés Java 8 et 9 !

A travers notre nouvelle [formation Nouveautés Java 8 et 9](#) (durée 4 jours) conçue par [l'auteur de ce tutoriel](#), découvrez par la pratique le nouveau système de **Modules** Java 9 (Encapsulation forte + Intercafes + Dépendances explicites) ainsi que les spécificités de Java 8



vos compétences et l'évolution de votre

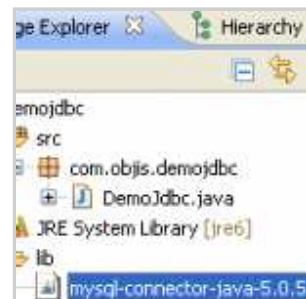
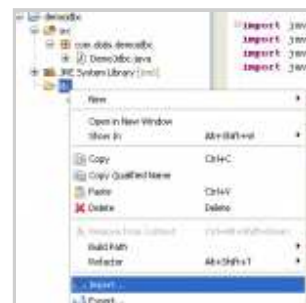
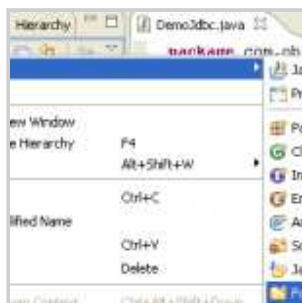
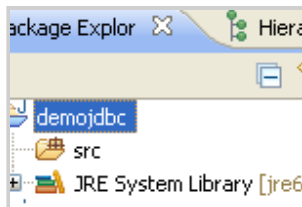
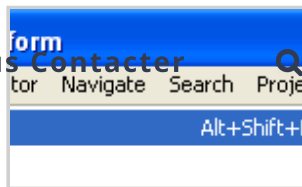
+ votre carrière professionnelle ! [Contactez-nous dès maintenant !](#)




Nous Contacter

tor Navigate Search Proje

Alt+Shift+F





```

# Run ASSET and store the resulting genome in
#
# @param genome - genome to be used for mapping
# @param ref - reference genome (Genome object)
# @param index - index
# @param output - output
# @return genome
#
def run(
    # Run ASSET on the genome
    genome = None,
    # Run ASSET on the genome
    ref = None,
    # Run ASSET on the genome
    index = None,
    # Run ASSET on the genome
    output = None,
    # Run ASSET on the genome
    **kwargs
):
    # Run ASSET on the genome
    genome = run_asset(
        genome=genome,
        ref=ref,
        index=index,
        output=output,
        **kwargs
    )
    return genome

```

Package Explorer

Hierarchy

demojdbc

src

com

obj

demojdbc

DemoJdbc.java

DemoJdbc

main(String[])


args: String[]

JRE System Library

Referenced Libraries

mysql-connector-java-5.0.5-bin.jar

lib



```

demojdbc
src
  com.objis.demos.jdbc
    DemoJdbc.java

```

[illegible][illegible]

emoJdbc [Java Application] C:\

informationjava\livraisons	
▲	Taille
b	

[illegible]



Nous allons vous faire aimer Java

+

Nous Contacter


Taille

informationjava\livraisons

emojjdbc.jar

```
java -cp demojdbc.jar c
tion: com.mysql.jdbc.Dr
oader.findClass(Unknown
oader.loadClass(Unknown
AppClassLoader.loadCl
oader.loadClass(Unknown
rName0(Native Method)
rName0(Unknown Source)
c.DemoJdbc.saverEnBase
c.DemoJdbc.main(DemoJdb
ava.lang.NullPointerExc
c.DemoJdbc.saverEnBase
c.DemoJdbc.main(DemoJdb
```

b/mysql-conne



```
MANIFEST.MF
1 Manifest-Version: 1.0
2 Main-Class: com.obj
3 Class-Path: lib/mys
4
```

[illegible]

Microsoft Word - New Project

Template

- Normal.dot

Template description

This is the default template for the MS Word application. It contains the default settings for the application, including the font, paragraph, and page settings. It also contains the default settings for the application, including the font, paragraph, and page settings. It also contains the default settings for the application, including the font, paragraph, and page settings.

Files of type: Word 97-2003 Template (*.dot)

Look in: C:\Program Files\Microsoft Office\Templates

Files

- Normal.dot

File name:

Save in: C:\Program Files\Microsoft Office\Templates

Save as type: Word 97-2003 Template (*.dot)

Save as:

OK **Cancel**

MainHost Specification

Generate the main host file

☒ Select the installed in the workspace

☐ Use the specified content of the generated file description file

Previous Next

Back Next Cancel OK

Host Test Specification

To match the specified

Add the matched

Generate the matched file

☒ Save the matched file in the workspace

☐ Use the saved matched file in the generated JAR description file

Browse...

Use existing matched files in workspace

Matched file: Browse...

OK Cancel Apply Cancel

Manifest Selection

Select a manifest file which should be added to the JAR:

- demojdrbc
 - src
 - lib
 - META-INF
 - MANIFEST.MF**
 - .classpath
 - .project

OK Cancel

[illegible]

nationjava\livraisons	
	Taille
mojdbc.jar	3
mojdbc2.jar	3

```
raisons>java
```

OBJIS

LIENS LES PLUS

FORMATIONS

JAVA 1

FORMATIONS

JAVA 2

AUTRES FORMATIONS

