

Tuto installation en local du projet MO50

Logiciels utilisés :

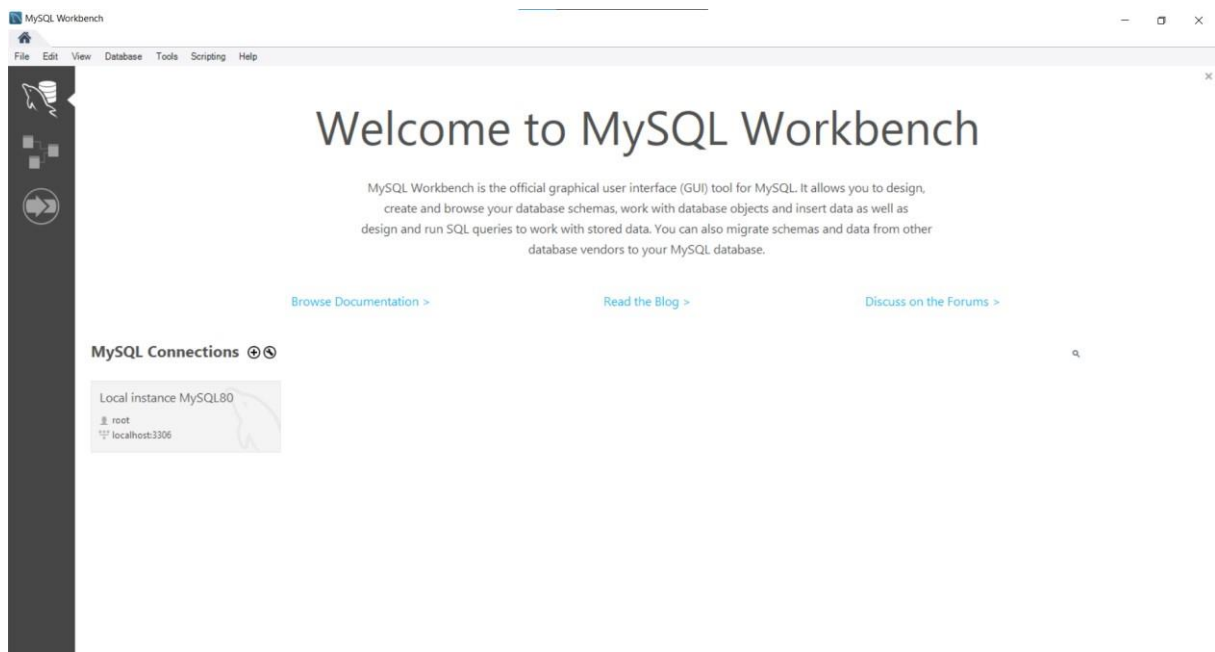
- Eclipse IDE for Enterprise Java and Web Développeur : <https://www.eclipse.org/downloads/>
- MySQL : <https://dev.mysql.com/downloads/installer/>

L'installation d'Eclipse IDE se fait avec l'installer avec le lien ci-dessus. Aucun autre plugin n'est nécessaire.


L'installation de MySQL est bien plus longue et nécessite un petit guide : <https://dev.mysql.com/doc/mysql-getting-started/en/>

Pendant l'installation, un nom d'utilisateur / mot de passe doit être choisi. Ce sont ces identifiants de connexion qui seront utiles pour se connecter à la base de données local.

Une fois MySQL d'installer, il faut créer une nouvelle base de données en ouvrant le Workbench MySQL.




Dans le Workbench, ouvrir la session locale.

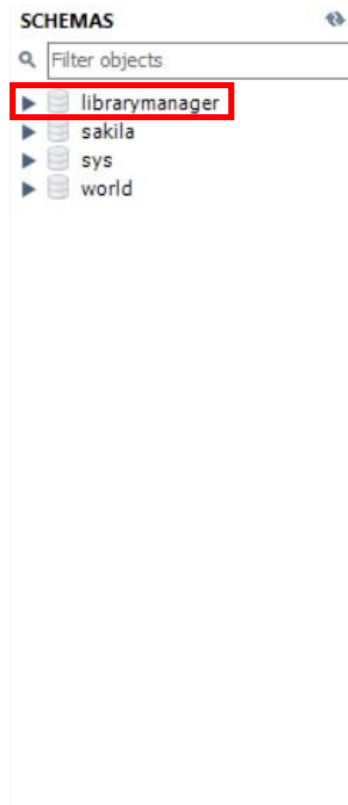
Créer une nouvelle query en cliquant sur le bouton  en haut à gauche.

Dans cette query, créer la base de données « librarymanager » en entrant la ligne de commande :

CREATE DATABASE librarymanager;

Puis exécuter cette query en cliquant sur 

Une fois exécuter, la nouvelle base de données devrait apparaître à gauche de l'écran.



Dans la query, supprimer la ligne de commande pour créer la base de données et copier-coller la totalité du fichier « librarymanager.sql » présent dans le dossier « ressource » du bitbucket.

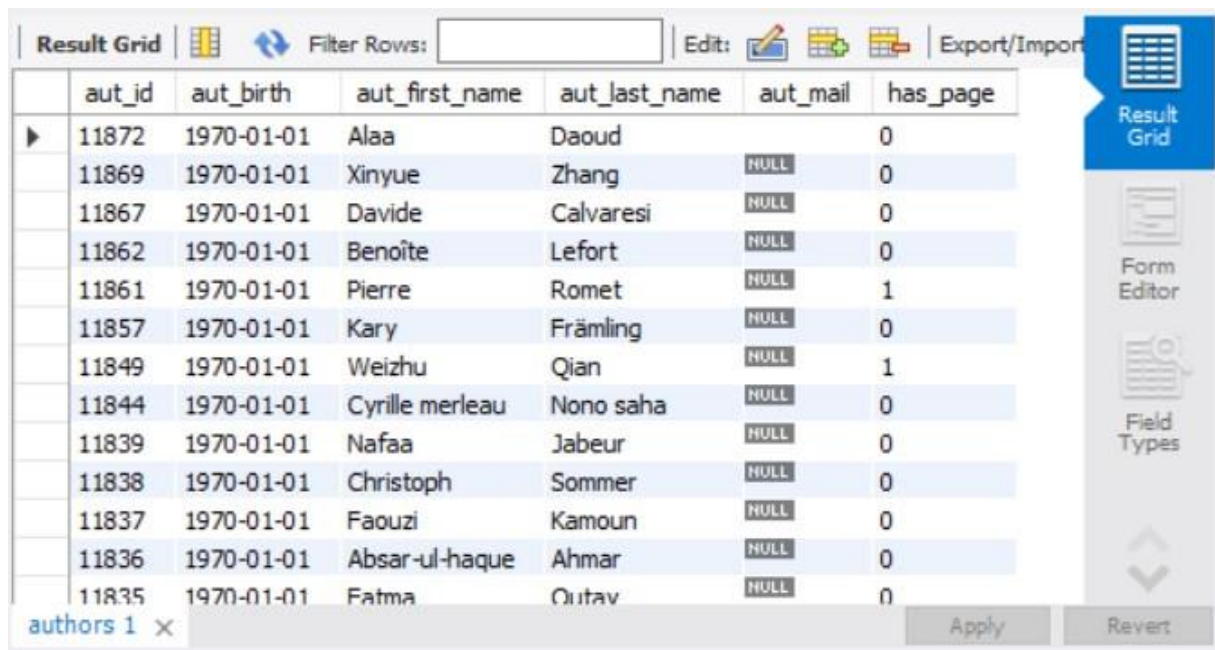
Rajouter la ligne de commande « USE librarymanager; » au début de la query.

Exécuter la query et vérifier que les données ont bien été ajoutées dans la base de données.

Aller dans « librarymanager » -> « tables » et, par exemple, sur la table « authors », cliquer sur la petite icône en forme de carré :



Une petite fenêtre devrait s'ouvrir avec les auteurs de la base de données :



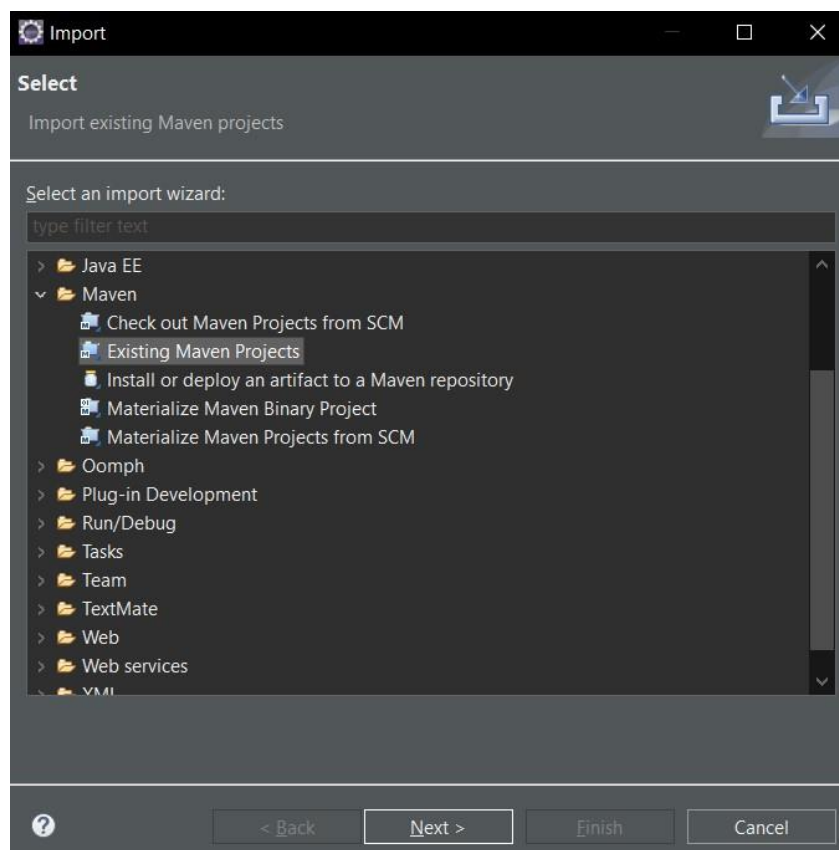
The screenshot shows a database application interface. At the top, there's a toolbar with icons for 'Result Grid', 'Filter Rows', 'Edit', and 'Export/Import'. Below the toolbar is a table with the following columns: aut_id, aut_birth, aut_first_name, aut_last_name, aut_mail, and has_page. The table contains 15 rows of data. To the right of the table is a sidebar with icons for 'Result Grid', 'Form Editor', and 'Field Types'. At the bottom of the table, there's a label 'authors 1' and two buttons: 'Apply' and 'Revert'.

	aut_id	aut_birth	aut_first_name	aut_last_name	aut_mail	has_page
▶	11872	1970-01-01	Alaa	Daoud		0
	11869	1970-01-01	Xinyue	Zhang	NULL	0
	11867	1970-01-01	Davide	Calvaresi	NULL	0
	11862	1970-01-01	Benoîte	Lefort	NULL	0
	11861	1970-01-01	Pierre	Romet	NULL	1
	11857	1970-01-01	Kary	Främling	NULL	0
	11849	1970-01-01	Weizhu	Qian	NULL	1
	11844	1970-01-01	Cyrille merleau	Nono saha	NULL	0
	11839	1970-01-01	Nafaa	Jabeur	NULL	0
	11838	1970-01-01	Christoph	Sommer	NULL	0
	11837	1970-01-01	Faouzi	Kamoun	NULL	0
	11836	1970-01-01	Absar-ul-haque	Ahmar	NULL	0
	11835	1970-01-01	Fatma	Qutav	NULL	0

La base de donnée est maintenant prête, il ne reste plus qu'à lancer le projet avec Eclipse.

Lancer Eclipse en utilisant un Workspace qui ne soit pas le fichier du code source.

Cliquer sur « File » -> « Import » et sélectionner le dossier « Maven » puis « Existing Maven project ».



Dans la fenêtre suivante, dans la case « Root Directory », allez chercher le fichier qui contient le code source qui doit normalement porter comme nom « pub-provider-2 ».

La fenêtre en dessous devrait afficher le fichier pom.xml avec la case cochée. Finish.

Le projet est maintenant importé dans Eclipse.

Dans le fichier « application.yml » présent dans le dossier « src/main/resources », remplacer les identifiants de connexion « #Local use » par l'identifiant et le mot de passe créés au moment de l'installation de MySQL.

```
datasource:
  driver-class-name: com.mysql.cj.jdbc.Driver
  url: jdbc:mysql://localhost:3306/librarymanager?serverTimezone=UTC

  #Deployed use :
#   username: ciadlabwpbdd
#   password: ci@dlAbpas$

  #Local use :
  username: myUsername
  password: myPassword

servlet:
  multipart:
    enabled: true
    max-file-size: 100MB
    max-request-size: 100MB
```

Enfin, dans le fichier « PubProviderApplication.java » présent dans le dossier « src/main/java » et dans le package « fr.ciadlab.pubprovider » il faut commenter la partie « when deployed » et décommenter la partie « localhost use ».

```
@SpringBootApplication
@Configuration
@EnableSpringConfigured
public class PubProviderApplication extends SpringServletInitializer {

    //For localhost use:
    public static final String PROXYURL="proxy.utbm.fr";
    public static final int PROXYPORT=3128;
    public static final String DownloadablesPath="Downloadables/";
    //Also go rename the logback file to "something logback.xml something.txt"

    //For when deployed:
    // public static final String PROXYURL = "";
    // public static final int PROXYPORT = 0;
    //public static final String DownloadablesPath = "/var/www/ciad-lab.fr/Downloadables/";
    //Also go rename the logback file to logback.xml

    public static void main(String[] args) {
        SpringApplication.run(PubProviderApplication.class, args);
    }
}
```

Lancer le projet depuis le fichier « PubProviderApplication.java » en « Java application »

Springboot devrait apparaître dans la console pour savoir si le projet a bien été lancé en local :

<http://localhost:8080/SpringRestHibernate/publicationsList>

Vérifier que le lien ci-dessus marche et qu'il affiche bien les données. S'il y a une erreur dans le lancement du projet, cela vient très probablement de la partie base de données.