

TP 7 : Utilisation de JDBC et de JavaFX

Matière : ATELIER PROGRAMMATION OBJET AVANCEE

Niveau : DSI 2

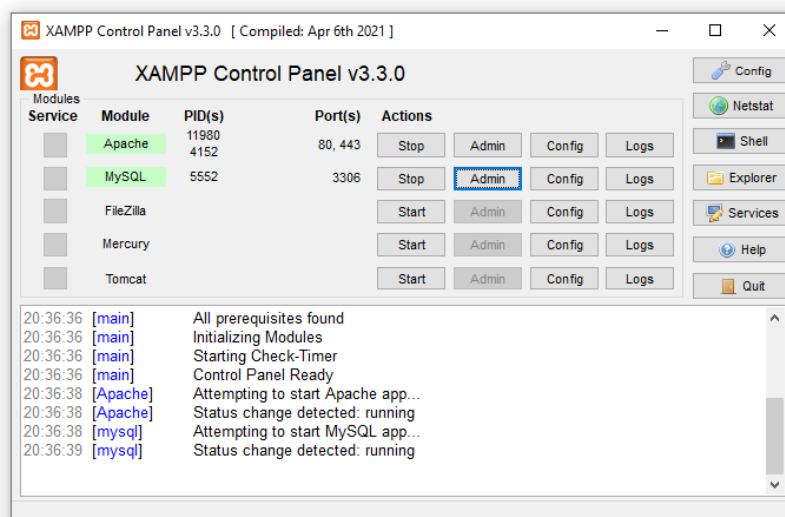
Enseignants : Equipe pédagogique

Etape 1 : Configuration de la base de données Mysql

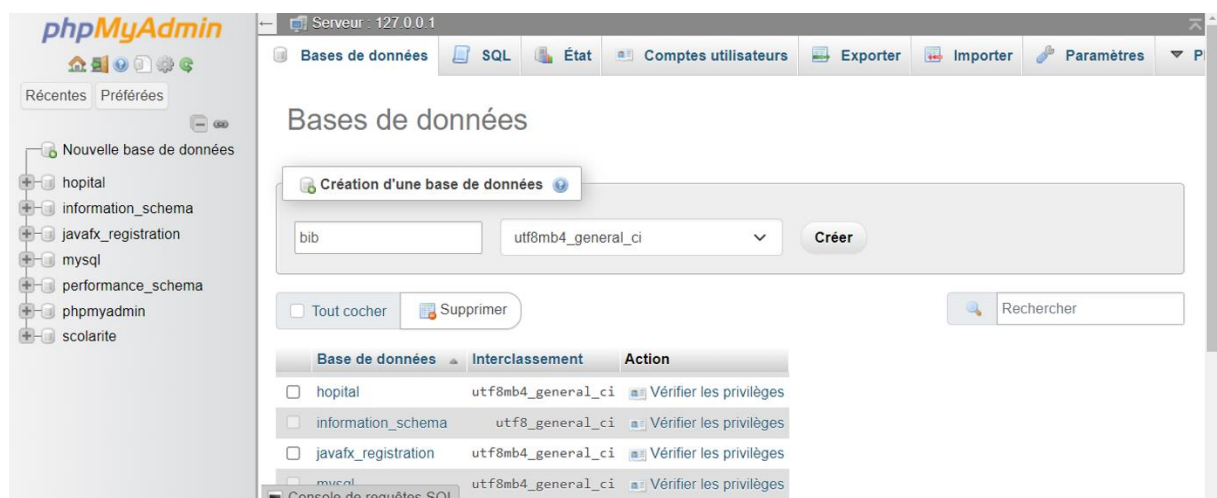
1. Installer Xampp

<https://www.apachefriends.org/fr/index.html>

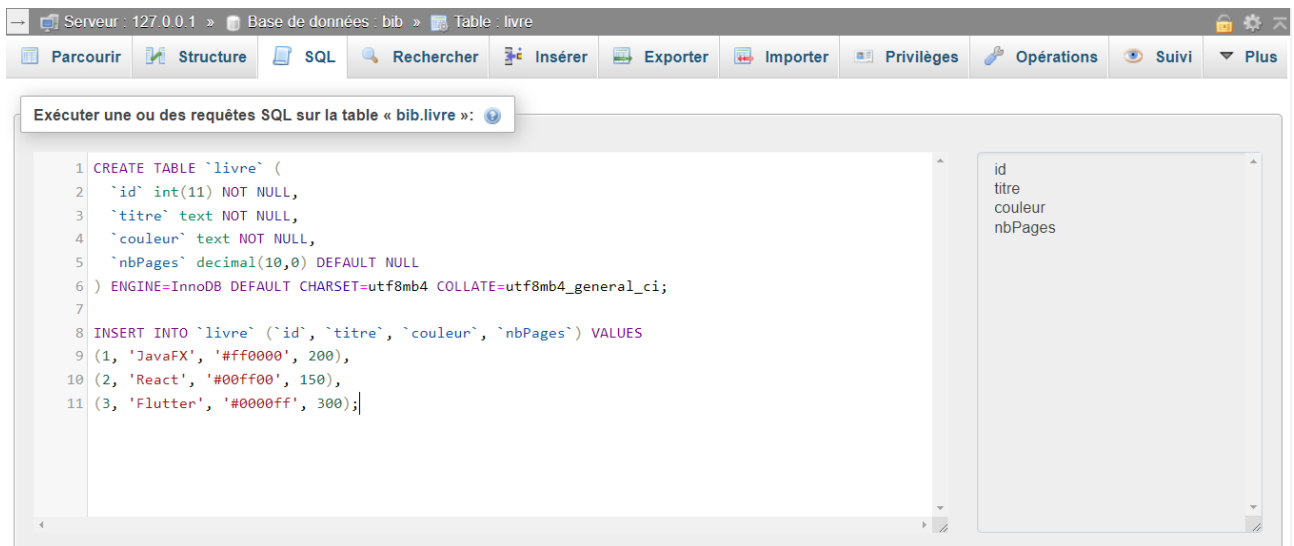
2. Dans le panneau de commande de Xampp, cliquer sur le bouton Admin du module Mysql afin d'ouvrir phpMyAdmin



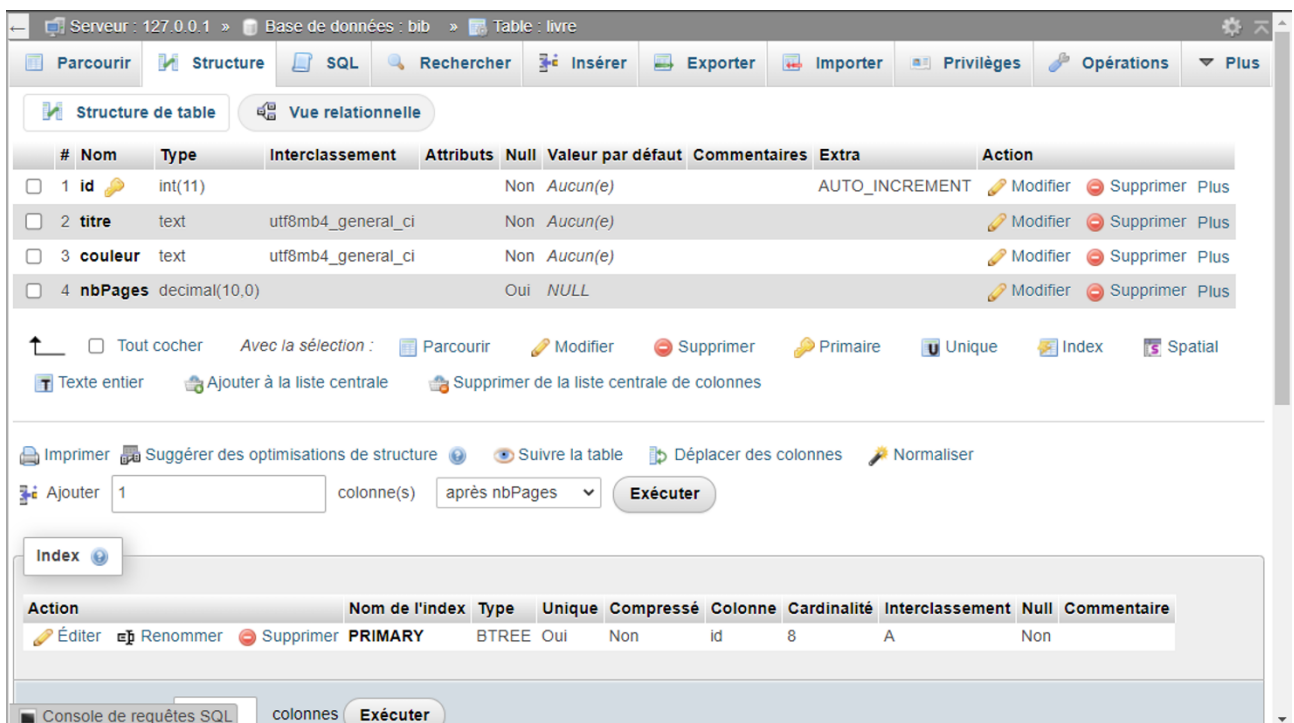
3. Dans phpMyAdmin, cliquer sur Nouvelle base de données, entrer « bib » puis cliquer sur Créer



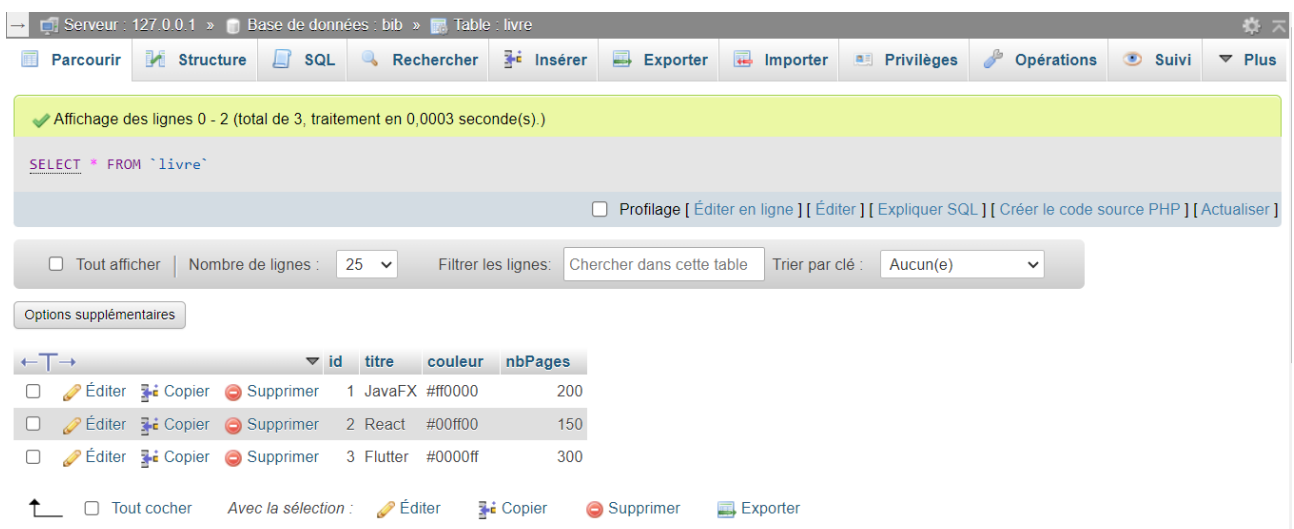
4. Cliquer sur l'onglet SQL et écrire le script de la création de la table Livre



5. Afficher la structure de la table livre

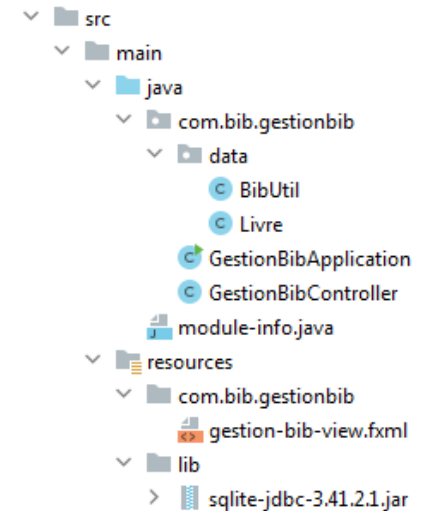


6. Sélectionner l'onglet Parcourir et afficher le contenu de la table livre



Etape 2 : Création de l'application JavaFX "Gestionbib"

1. Télécharger le dossier "APOA_TP08_EN.zip" et décompresser le.
2. Créer un nouveau projet JAVA FX avec maven
3. Utiliser copier/coller pour obtenir cette structure :



4. Dans "module-info.java", ajouter :

```
requires java.sql;  
opens com.bib.gestionbib.data to  
javafx.base;
```
5. Ajouter le pilote MySQL Connector/J dans le fichier pom.xml du projet :

```
<dependency>  
  <groupId>mysql</groupId>  
  <artifactId>mysql-connector-java</artifactId>  
  <version>8.0.32</version>  
</dependency>
```

6. Dans le fichier "BibUtil.java", définir le pilote JDBC pour MySQL dans la variable JDBC_Driver.

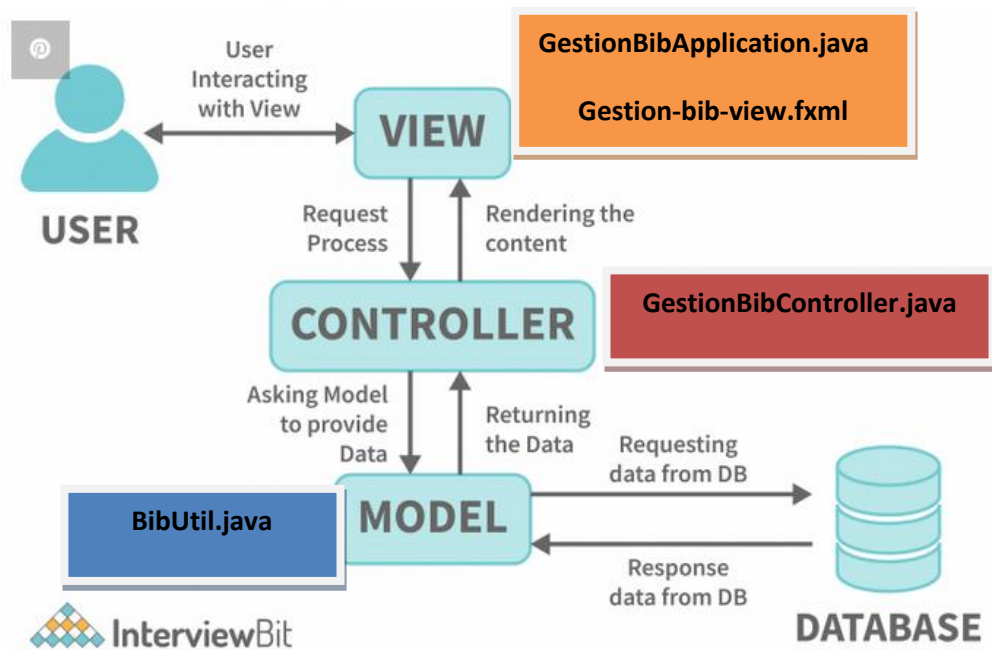
```
private static String JDBC_Driver = "com.mysql.cj.jdbc.Driver";
```

7. Définir également la chaîne de connexion dans la variable DB_Url

Nom d'hôte : "localhost:3306"	Utilisateur : "root"
Base de données : "bib"	Mot de passe : ""

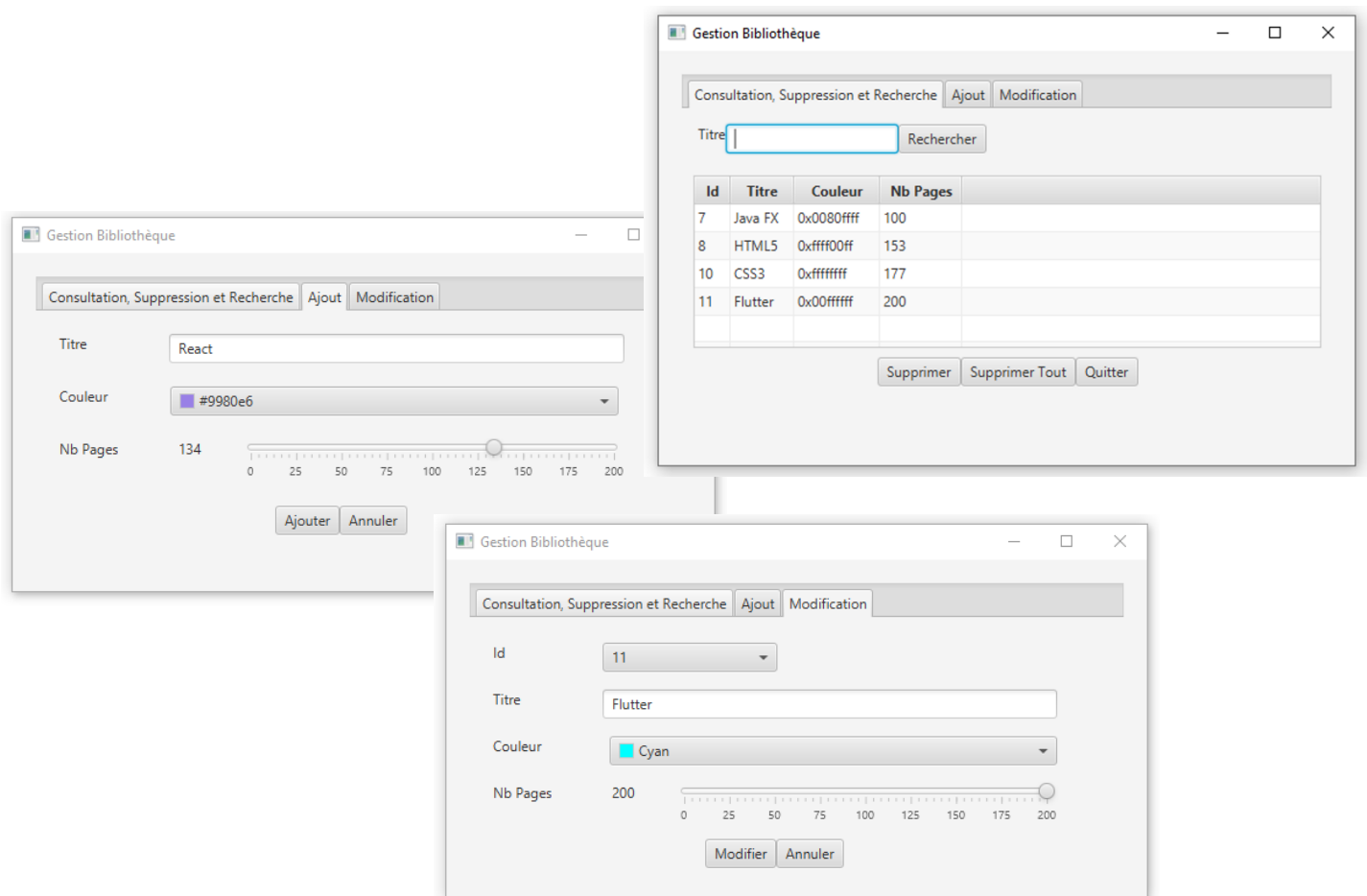
```
private static String DB_Url =  
"jdbc:mysql://localhost:3306/bib?user=root&password=";
```

L'application suivant l'architecture MVC :



Architecture MVC [<https://www.interviewbit.com/blog/mvc-architecture/>]

Les interfaces graphiques sont :



8. Compléter la classe "BibUtil" : afin de manipuler la table livre (**id**, titre, couleur, nbPage)

Méthode	Description
ObservableList<Livre> getLivres()	Retourne la liste des livres en exécutant la requête : select * from livre order by id;
Livre getLivreById(int id)	Retourne le livre qui a l'id passé en paramètre en exécutant la requête : select * from livre where id=?;
boolean ajouterLivre(String titre, String couleur, int nbPage)	Ajoute un livre en exécutant la requête : insert into livre(titre,couleur, nbPages) values (?, ?, ?);
boolean modifierLivre(int id, String titre, String couleur, int nbPage)	Modifie un livre en exécutant la requête : update Livre set titre=?,couleur=?, nbPages=? where id=?;
ObservableList<Livre> rechercherLivres(String titreR)	Recherche et retourne la liste des livres qui contiennent titreR dans leurs noms en exécutant la requête : select * from livre where titre like ? order by id;
boolean supprimerTout()	Supprime tous les livres en exécutant la requête : delete from livre;
boolean supprimerLivre(int id)	Supprime le livre qui a l'id passé en paramètre en exécutant la requête : delete from livre where id= ?;