API JDBC

Pour voir les résultats de ce TP, Vous devez disposer de :

- **Une base de données installée (MySQL dans notre cas)**
- **Le driver de la base de données (mysql-connector-java-X.X.XX-bin.jar, avec X.X.XX représente la version)**

Dans ce TP, nous allons travailler avec une base de données locale sous le nom « FSTBASE » et contenant la table « FST_COURS » contenant la référence et le libellé du cours.

1- Couche DAO (Data Access Object)

1. Créer un projet sous le nom AccesJdbcProjet
2. Créer un package ma.config qui contient un fichier des paramètres de notre base de données

CoursBO.java | db.properties | persistence.cfg.xml

```
url=jdbc:mysql://127.0.0.1:3306/fstbase
login=root
pwd=root
```

3. Créer un package ma.gov.testjdbc.commun qui contient la classe ConnectDB.JAVA
4. Créer dans la classe ConnectDB.JAVA la méthode avoirConnection() dans laquelle vous lisez les paramètres de param.properties.

```
public Connection avoirConnection() throws ClassNotFoundException, SQLException
{
    // Lecture des paramètres
    ResourceBundle rb = PropertyResourceBundle.getBundle("ma.config.param");
    String url =rb.getString("url");
    String login =rb.getString("login");
    String pass =rb.getString("pass");
    // Création d'un objet Connection
    Class.forName("com.mysql.jdbc.Driver");
    Connection con = DriverManager.getConnection(url,login,pass);
    return con;
}
```

Cette méthode doit remonter les exceptions ClassNotFoundException si le driver n'est pas disponible et SQLException si les paramètres de la base de données ne sont pas correcte.
Charger le driver via Class.forName() ;
Récupérer un objet Connection en appelant la méthode getConnection() de la classe DriverManager. Cette méthode est statique. C'est quoi votre preuve

5. Créer une autre méthode fermerConncetion() qui ferme la Connection s'il n'est pas null.

```
public void fermerConnection(Connection con) throws SQLException
{
    if(con!=null)
    {
        con.close();
    }
}
```

6. **Rendez la classe ConnectDB.JAVA singleton.**

```
private static ConnectDB instance = null;
private ConnectDB() {}
public static ConnectDB getInstance()
{
    if(instance!=null)
    {
        instance =new ConnectDB();
    }
    return instance;

}
```

7. Créer un package ma.gov.testjdbc.metier qui contient la classe metier CoursBO.JAVA
   Un cours est identifiant par son référence et son libellé
8. Créer un package ma.gov.testjdbc.dao qui contient la classe CoursDao.JAVA
9. Créer la méthode insertCours() dans la classe CoursDao.JAVA
   Cette méthode prend un cours et une connexion comme paramètres

```java
public void insertCours(CoursBO cours, Connection con) throws SQLException {
    String reqSQL = "INSERT INTO EMSI_COURS VALUES" +
            "('" + cours.getReference() + "','" + cours.getLibelle() + "')";
    Statement st = con.createStatement();
    st.executeUpdate(reqSQL);
}
```

10. Créer un package ma.gov.testjdbc.applicatif contenant l'interface ICoursService.JAVA et son implémentation CoursServiceImpl.JAVA
Créer la méthode ajouterCours() qui prend le paramètre CoursBO et proposer l'implémentation suivante :

```java
public void ajouterCours(CoursBO cours)
throws ClassNotFoundException, SQLException
{
    ConnectDB conDb =ConnectDB.getInstance();
    Connection con = conDb.avoirConnection();
    CoursDao dao =new CoursDao();
    dao.insertCours(cours, con);
}
```

11. Créer un package ma.gov.testjdbc.presentation contenant une classe avec le main().
Appeler la méthode ajouterCours de la couche applicative.

```java
public class Traitement {
    public static void main(String[] args) {
        CoursBO cours = new CoursBO();
        cours.setReference("maRef") ;
        cours.setLibelle("lib") ;
        ICoursService service = new CoursServiceImpl();
        try {
            service.ajouterCours(cours);
        } catch (ClassNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```