

Bibliothèque Studi

Examen certifiant du bloc - 1er dépôt - Le développement d'une solution digitale



POSADAS RODRIGUEZ Jesus Humberto

20/10/2022

Document technique

Lien du projet déployé

Le projet est en ligne et peut être consulté dans le lien suivant :

<http://bibbliostudi.sh2.hidora.net/>

Attention, depuis la dernière version du projet, j'ai implémenté une Remote Database pour rendre mon application plus facile à installer pour le formateur en locale, mais cela m'a occasionné un petit problème que j'imagine se trouve dans le fait que l'hébergeur de ma base de données Remote se trouve en Amérique (c'est Banahosting, l'hébergeur de l'entreprise qui m'accueille pour faire mon stage de fin de formation) et l'hébergeur de mon application se trouve en Europe.

Le problème c'est que lorsque je charge la page la première fois une erreur 500 de driver mysql avec le message suivant :



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/bibliotheque/'. Below the address bar, a dark blue header bar reads 'État HTTP 500 – Erreur interne du serveur'. The main content area displays an exception report with the following details:

- Type:** Rapport d'exception
- message:** java.sql.SQLException: No suitable driver found for jdbc:mysql://198.136.62.112:3306/fubnvjop_bibliodb
- description:** Le serveur a rencontré une erreur interne qui l'a empêché de satisfaire la requête.
- exception:**

```
java.lang.RuntimeException: java.sql.SQLException: No suitable driver found for jdbc:mysql://198.136.62.112:3306/fubnvjop_bibliodb
    org.hposadas.bibliotheque.filters.ConnectionDBFilter.doFilter(ConnectionDBFilter.java:47)
```
- cause mère:**

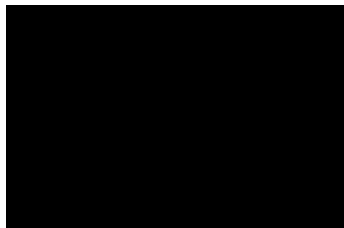
```
java.sql.SQLException: No suitable driver found for jdbc:mysql://198.136.62.112:3306/fubnvjop_bibliodb
    java.sql/java.sql.DriverManager.getConnection(DriverManager.java:708)
    java.sql/java.sql.DriverManager.getConnection(DriverManager.java:230)
    org.hposadas.bibliotheque.utilities.ConnectionDB.getConnection(ConnectionDB.java:16)
    org.hposadas.bibliotheque.filters.ConnectionDBFilter.doFilter(ConnectionDBFilter.java:22)
```
- note:** La trace complète de la cause mère de cette erreur est disponible dans les fichiers journaux de ce serveur.

At the bottom of the error page, a dark blue footer bar displays 'Apache Tomcat/10.0.23'.

Ce problème va se résoudre en rechargeant la page. Alors si vous faites recharger la page vous allez avoir l'application qui apparaît soit en local une fois l'application installée dans votre ordinateur ou bien sur le lien d'accès au projet déployé.



Couleurs



#000000



#ffc107



#0d6efd

- Pour le choix des couleurs j'ai pris le couleur noir qui représente l'environnement ombre nécessaire pour une bonne visibilité de l'écran en empêchant la fatigue visuelle.

- La couleur jaune vient du logo Studi.
- La couleur bleue représente ma couleur préférée

Contexte

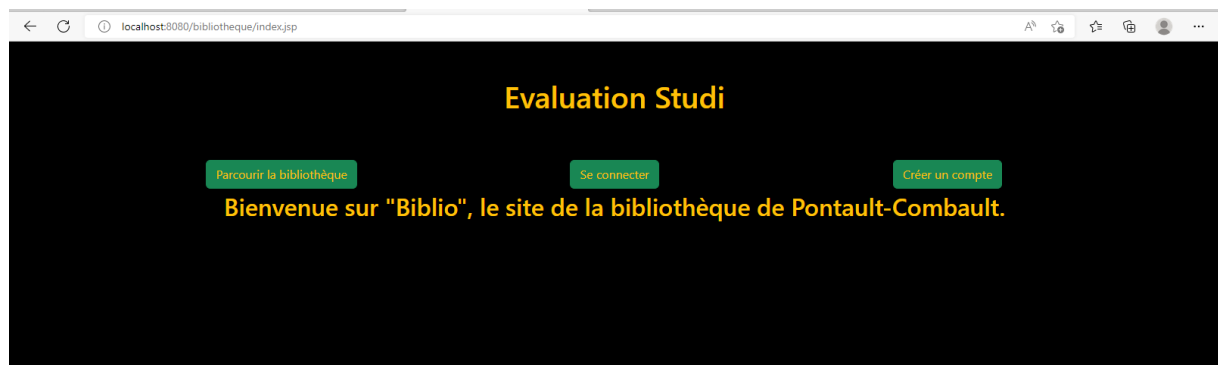
Il s'agit d'une application capable de résoudre un besoin en spécifique de la part du client, la gestion de la bibliothèque qui n'avait pas de système de gestion informatique car tout était géré en papier.

Ce système, déjà commencé par le chef de projet, devait respecter les standards opquast accessibles dans le lien suivant :

<https://checklists.opquast.com/>

Ecrans

Index :



Créer un compte :

← ↻ ⓘ localhost:8080/bibliotheque/creationCompte.jsp A🔖🌟🔍👤...

Parcourir la bibliothèque Se connecter Accueil

Création du compte

Nom:
Votre nom ici.
Ce champ est obligatoire *

Prénom
Votre prénom ici.
Ce champ est obligatoire *

Email:
example@domain.com
Ce champ est obligatoire *

Mot de passe

Ce champ est obligatoire *

Créer

Connexion :

← ↻ ⓘ localhost:8080/bibliotheque/connexion A🔖🌟🔍👤...

Parcourir la bibliothèque Accueil Créer un compte

Connexion

Email:
example@domain.com

Mot de passe:

Se connecter

Modifier informations :

localhost:8080/bibliotheque/infoUser

Parcourir la bibliothèque Prets en cours Accueil

Information du compte

ID: 10
Nom: Alex manuel
Prenom: Marin lopez
Email: am@test.com
Password: 12345

Modifier

Rechercher des livres :

Liste des livres

Titre	Genre	Author	Langue	Editeur	Date de publication	Demander en prêt
Le petit prince	Roman	Antoine de Saint-Exupéry	Français	Reynal & Hitchcock	2020-12-10	Demander en prêt
Odyssée	mythologique	Homère	Français	Reynal & Hitchcock	2020-12-06	Demander en prêt
Voyage au bout de la nuit	Roman	Louis-Ferdinand Céline	Français	Denoël et Steele	2020-12-09	Demander en prêt
Paroles 1	Fiction	Jacques Prévert	Français	Le Point du Jour	2020-12-08	Demander en prêt
Madame Bovary	Realisme	Gustave Flaubert	Français	Michel Lévy frères	2020-12-07	Demander en prêt

Accueil

Recherche par titre

Suivre les prêts en cours :

Prêts en cours

Parcourir la bibliothèque Information du compte Se déconnecter

Afficher prêts

Id	Utilisateur	Livre	Date debut	Date fin	Prolongements
1	Posadas Jesus	Madame Bovary	20-10-2022	4-11-2022	0
1	Posadas Jesus	Le petit prince	20-10-2022	4-11-2022	0
1	Posadas Jesus	Voyage au bout de la nuit	20-10-2022	4-11-2022	0

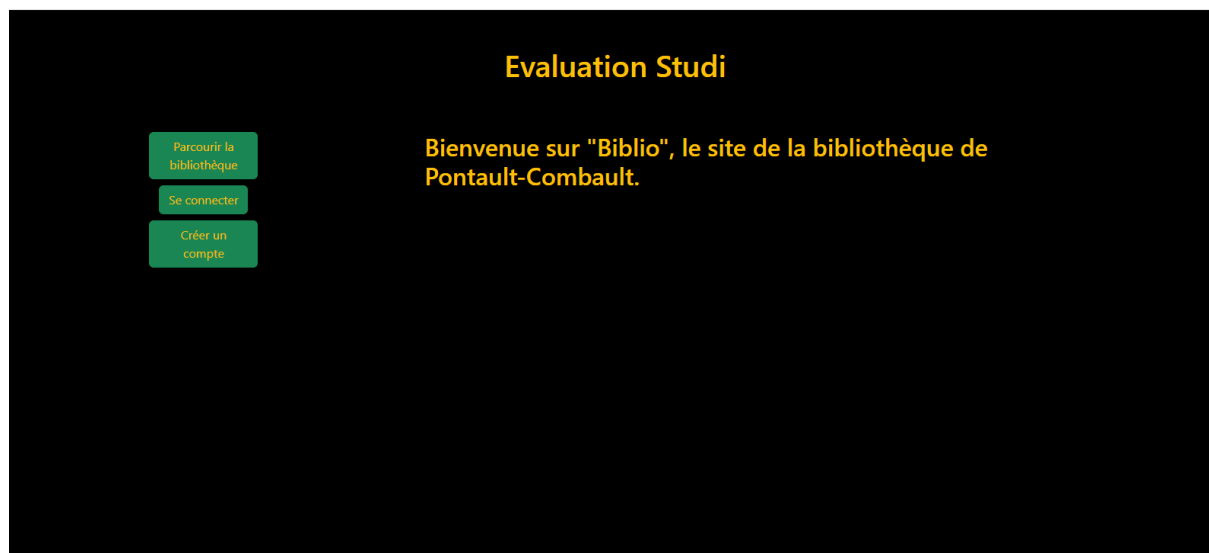
[Demander d'autres prêts](#)
[Home](#)

Fonctionnalités

Les fonctionnalités présentes dans le projet sont les suivantes.

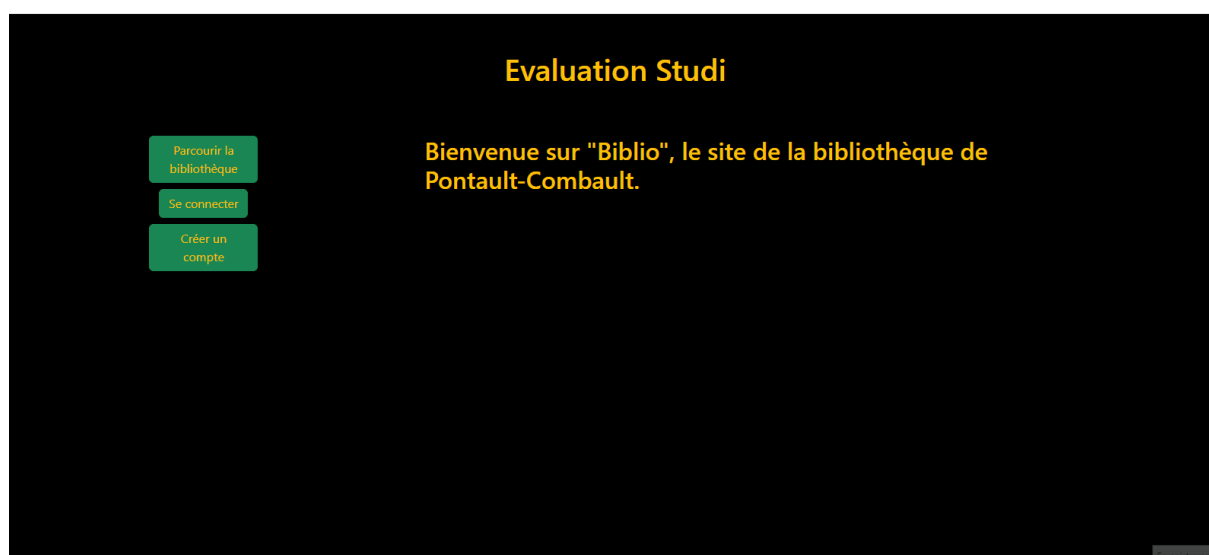
Créer un compte utilisateur

Depuis l'écran d'accueil il est possible de créer un compte utilisateur en cliquant sur le bouton « Créer un compte ».



S'identifier

Depuis l'écran d'accueil il est possible de s'identifier en cliquant sur le bouton « Se connecter ».



Afficher / Modifier les informations du compte utilisateur

Depuis l'écran de prêts en cours, l'utilisateur authentifié est capable d'afficher ou bien modifier les informations de son compte en cliquant sur le bouton « Information du compte ».

The screenshot shows a web interface with a dark background. At the top, the title "Prêts en cours" is displayed in orange. Below it, there are three green buttons: "Parcourir la bibliothèque", "Information du compte", and "Se déconnecter". Underneath the buttons, the title "Afficher prêts" is shown in orange. A table with six columns (Id, Utilisateur, Livre, Date début, Date fin, Prolongements) contains two rows of loan data. At the bottom, there are two blue links: "Demander d'autres prêts" and "Home".

Id	Utilisateur	Livre	Date début	Date fin	Prolongements
1	Posadas Jesus	Voyage au bout de la nuit	20-10-2022	4-11-2022	0
1	Posadas Jesus	Madame Bovary	20-10-2022	4-11-2022	0

Se déconnecter

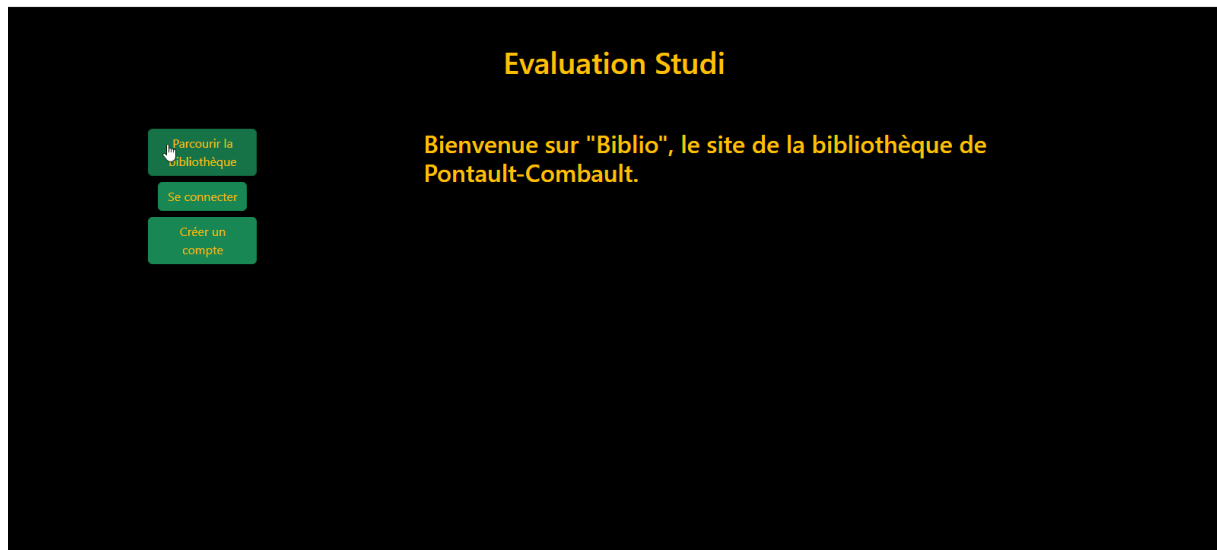
Depuis l'écran de prêts en cours et l'écran pour se connecter il existe un bouton pour se déconnecter.

This screenshot is identical to the one above, showing the "Prêts en cours" page. It includes the title, navigation buttons, the "Afficher prêts" title, the table of active loans, and the bottom links. The "Se déconnecter" button is clearly visible among the top navigation options.

Id	Utilisateur	Livre	Date début	Date fin	Prolongements
1	Posadas Jesus	Voyage au bout de la nuit	20-10-2022	4-11-2022	0
1	Posadas Jesus	Madame Bovary	20-10-2022	4-11-2022	0

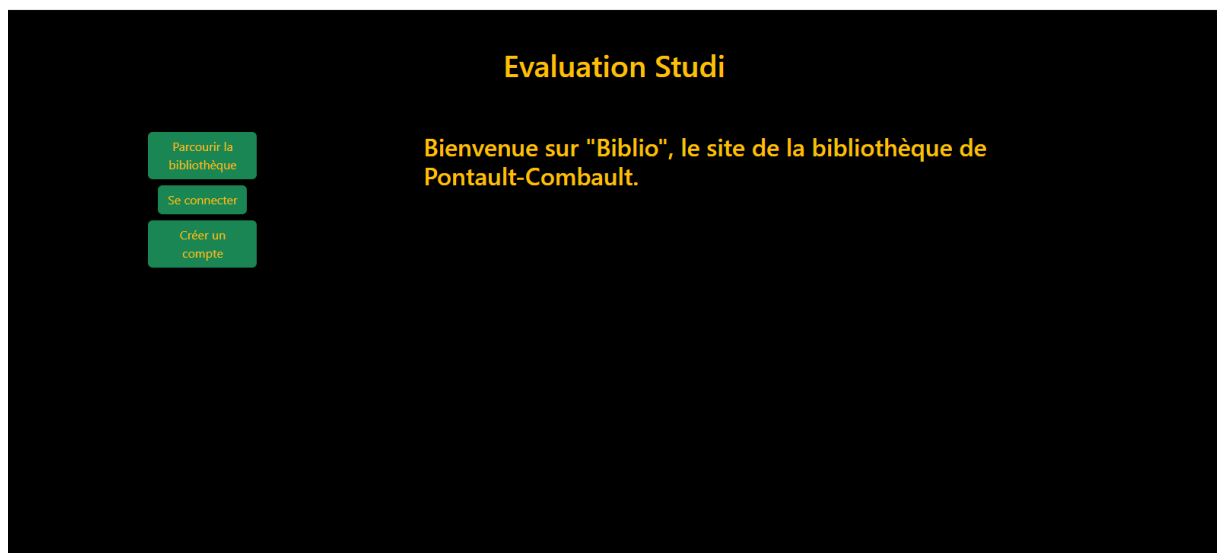
Afficher des livres dans la bibliothèque

Depuis l'écran d'accueil en cliquant sur le bouton Parcourir la bibliothèque.



Chercher des livres

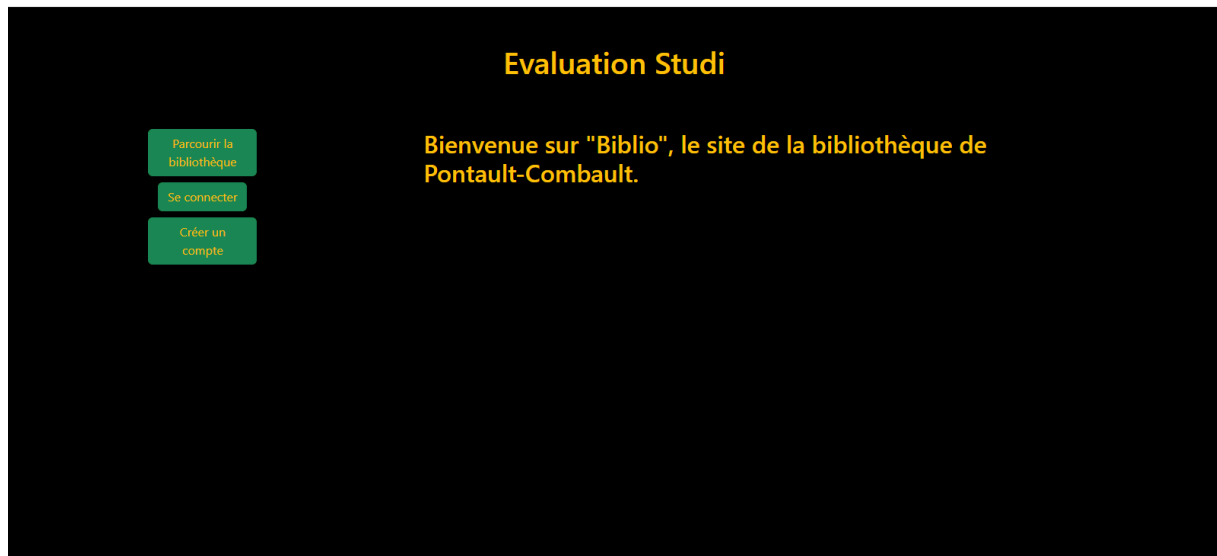
Il est possible de chercher des livres par titre depuis l'écran liste de livres.



Demander un livre en prêt

Il faut impérativement que l'utilisateur soit authentifié pour voir apparaître l'option Demander en prêt. Il suffit de se diriger vers

l'option « parcourir la bibliothèque » et la liste des livres va s'afficher avec l'option demander en prêt.



Suivre les prêts en cours

Une fois qu'on a demandé le prêt d'un livre automatiquement l'utilisateur sera dirigé vers l'écran « prêts en cours ».

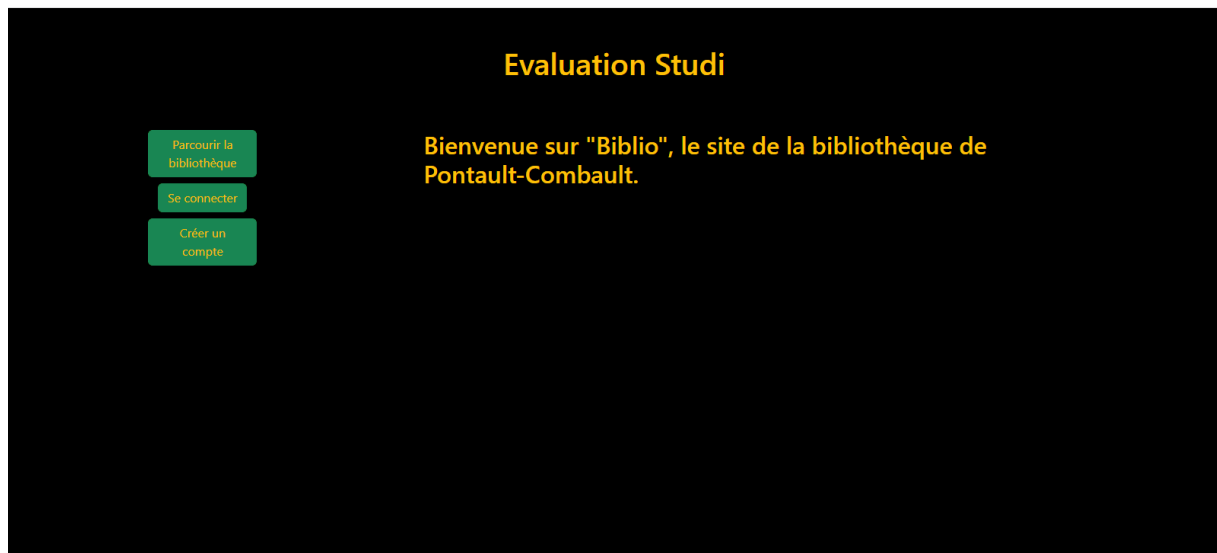
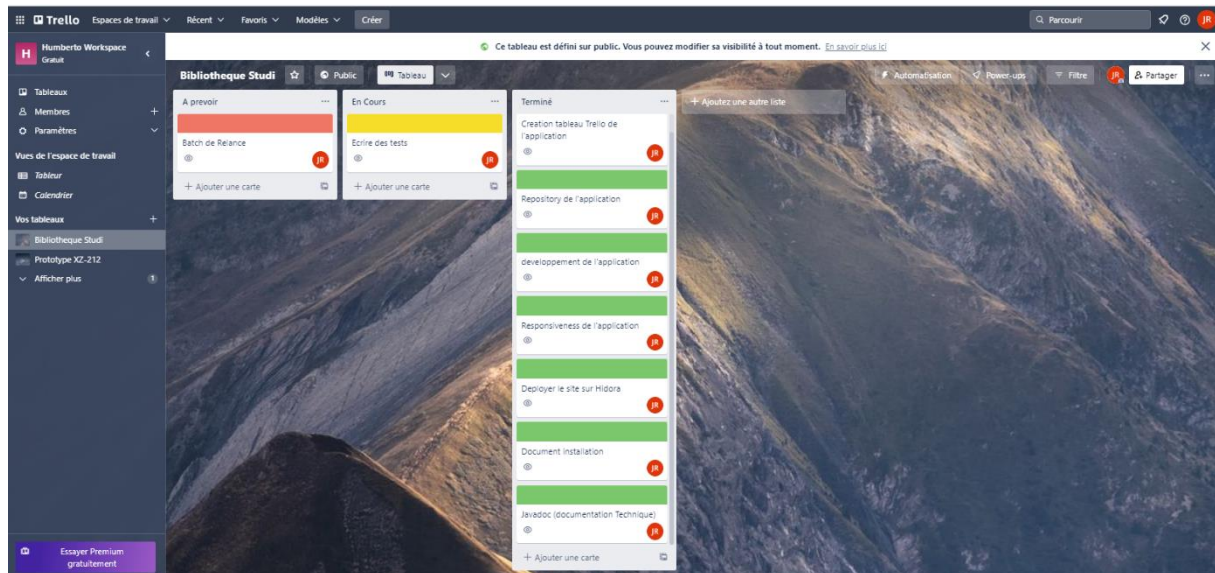


Tableau Trello – Etapes du projet

Pour décrire les étapes du projet ainsi que son état vous pouvez visiter le lien <https://trello.com/b/RR69Srdb/bibliotheque-studi>



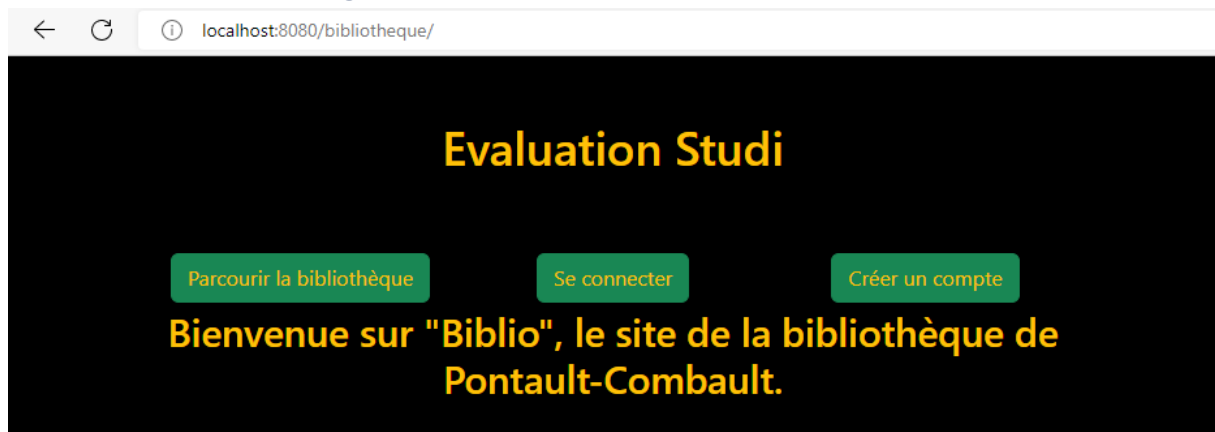
Version Responsive

L'application propose une adaptation sur tous les écrans pouvant interagir sur toutes les plateformes.

L'application est très légère d'un point de vue visuel, c'est pour cette raison que c'est très simple de la rendre responsive, il suffit d'établir le point de rupture pour le menu supérieur de 3 boutons pour apparaitre en block quand le point de rupture l'indique.

Pour obtenir cette version responsive j'ai utilisé Bootstrap. Je vous présente quelques exemples des écrans en différentes tailles pour montrer l'adaptation aux tailles comme vous allez le voir en suite :

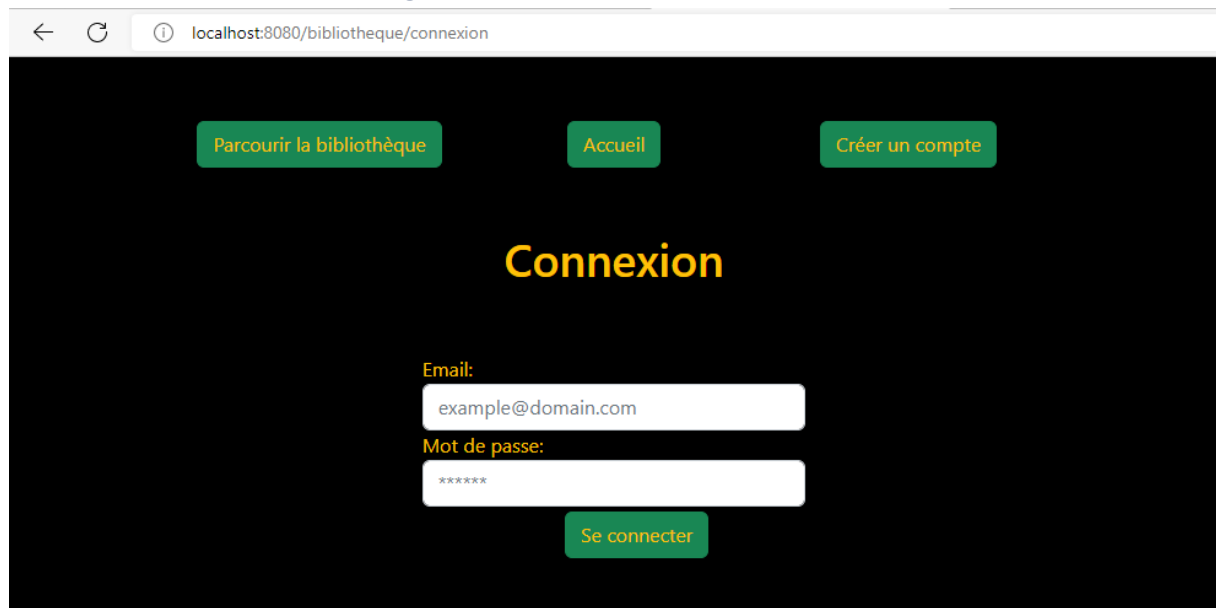
Accueil en taille Large



Accueil en taille Small



Connexion en taille Large



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/bibliotheque/connexion'. The page has a dark background and features three green buttons at the top: 'Parcourir la bibliothèque', 'Accueil', and 'Créer un compte'. The main heading 'Connexion' is centered in a large, bold, orange font. Below the heading, there are two input fields: 'Email:' with the value 'example@domain.com' and 'Mot de passe:' with masked characters '*****'. A green button labeled 'Se connecter' is positioned below the password field.

← ↻ ⓘ localhost:8080/bibliotheque/connexion

Parcourir la bibliothèque Accueil Créer un compte


Connexion

Email:
example@domain.com

Mot de passe:

Se connecter

Connexion en Taille Small



← ↻ ⓘ localhost:8080/bibliotheque/connexion

Parcourir la bibliothèque

Accueil

Créer un compte

Connexion

Email:

example@domi

Mot de passe:

Se connecter

Création de compte en Large

[←](#) [↻](#) [localhost:8080/bibliotheque/creationCompte.jsp](#)

[Parcourir la bibliothèque](#) [Se connecter](#) [Accueil](#)

Creation du compte

Nom:

Ce champ est obligatoire *

Prénom

Ce champ est obligatoire *

Email:

Ce champ est obligatoire *

Mot de passe

Ce champ est obligatoire *

[Créer](#)

Création de compte en Small

← ↻ ⓘ localhost:8080/bibliotheque/creationCompte.jsp

Parcourir la bibliothèque

Se connecter

Accueil

Creation du compte

Nom:

Ce champ est obligatoire *

Prénom

Ce champ est obligatoire *

Email:

Ce champ est obligatoire *

Mot de passe

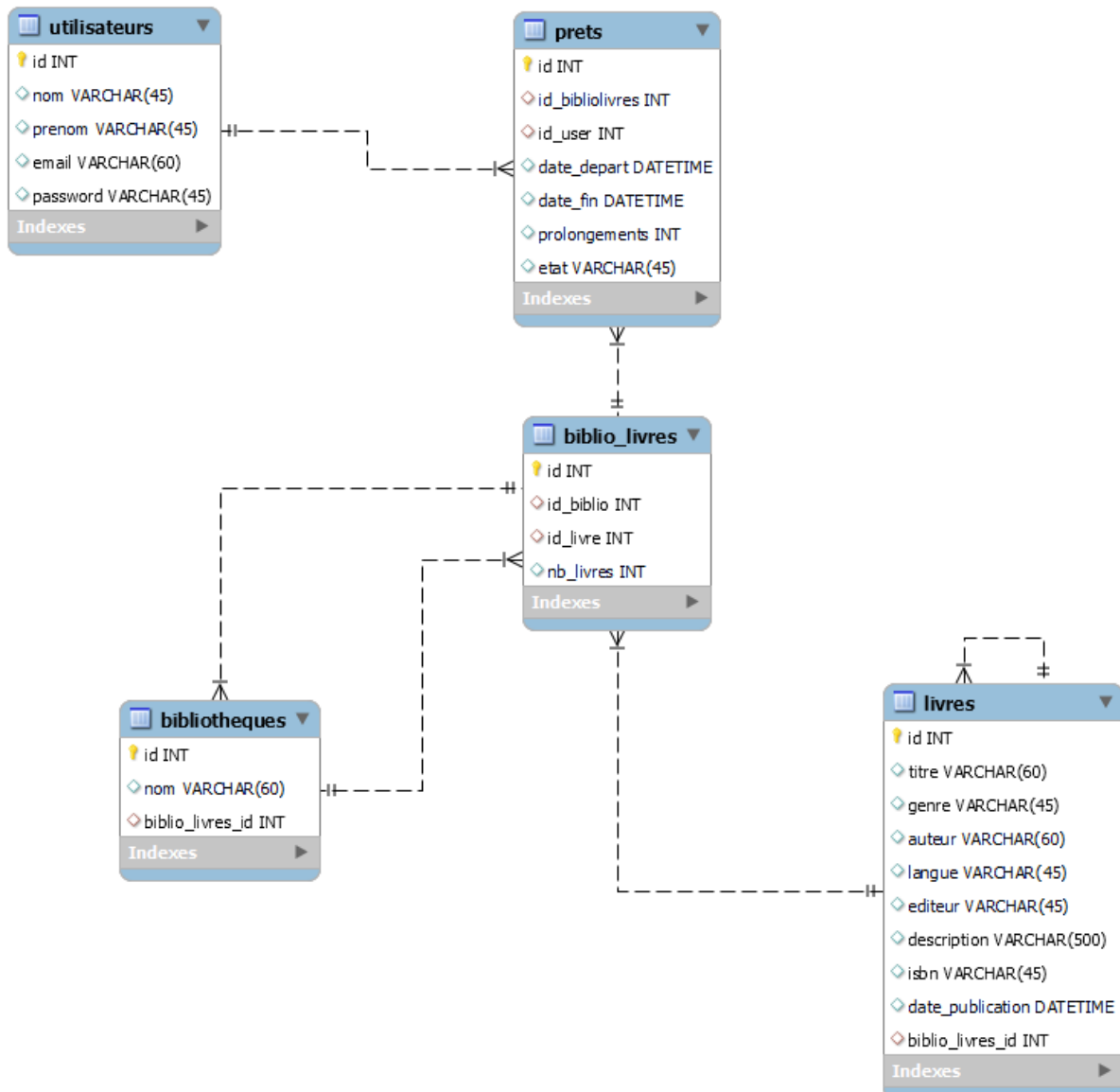
Ce champ est obligatoire *

Créer

Base de données

L'application fonctionne dans sa dernière version avec une Remote Database à l'aide d'un hébergeur fourni par mon entreprise où je suis en train d'effectuer mon stage de fin de formation.

Vous trouverez en suite le diagramme de ma base de données :



Installation du projet

Pour installer le projet il faudra suivre les étapes suivantes :

Installer Apache Tomcat 10

Se rendre dans la page <https://tomcat.apache.org/download-10.cgi> et télécharger apache tomcat 10 pour votre Système Opératif.

Rajouter le Driver JDBC pour mysql

Dans le repository

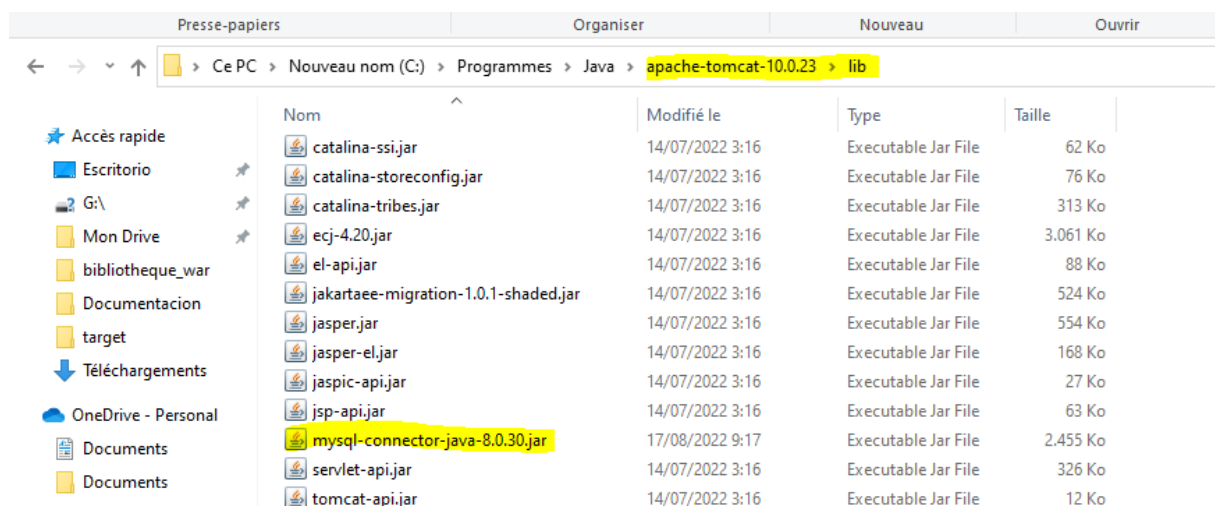
```
<dependencies>
  <dependency>
    <groupId>jakarta.platform</groupId>
    <artifactId>jakarta.jakartaee-api</artifactId>
    <version>9.0.0</version>
  </dependency>
  <dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.13.3</version>
  </dependency>
</dependencies>

<build>
  <finalName>${project.artifactId}</finalName>
  <plugins>
    <plugin>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.8.1</version>
    </plugin>
    <plugin>
      <groupId>org.apache.tomcat.maven</groupId>
      <artifactId>tomcat7-maven-plugin</artifactId>
      <version>2.2</version>
      <configuration>
        <url>http://localhost:8080/manager/text</url>
        <username>admin</username>
        <password>12345</password>
      </configuration>
    </plugin>
    <plugin>
      <artifactId>maven-war-plugin</artifactId>
      <version>3.2.2</version>
      <configuration>
        <failOnMissingWebXml>>false</failOnMissingWebXml>
      </configuration>
    </plugin>
  </plugins>
</build>
```

Ensuite il faut créer la base de données mysql en utilisant le fichier du script que j'ai mis sur le repository github dans le repository suivant :

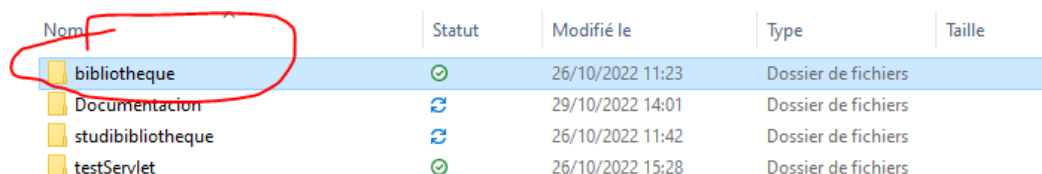
<https://github.com/PosadasHumberto/BibliothequeMaven.git>

A l'intérieur du dossier Documentation il faut télécharger le fichier « [mysql-connector-java-8.0.30.jar](#) » et en suite se diriger dans le dossier lib de notre installation du Tomcat, en suite il faudra déposer le driver a l'intérieur pour que notre serveur Apache Tomcat soit capable d'établir la connection vers la base de données.



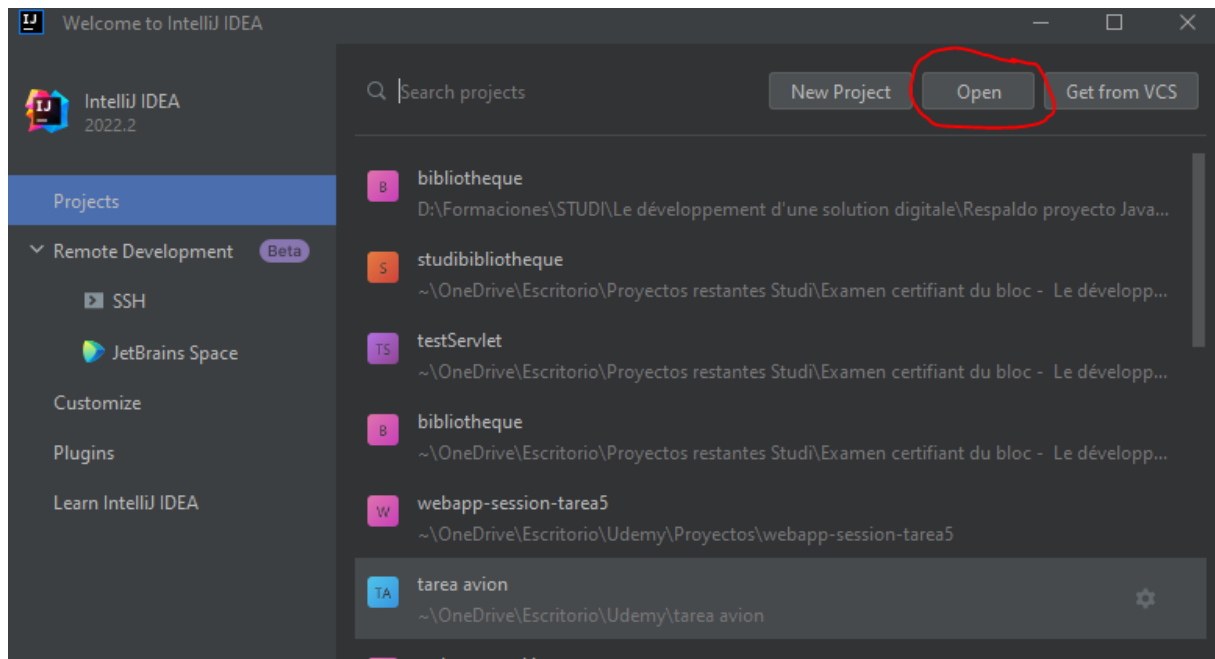
Ouvrir le projet avec IntelliJ IDEA

Dans le même repository Github se trouve le projet compressé nommé bibliotheque.rar, il faudra le télécharger pour ensuite le décompresser à l'emplacement souhaité.

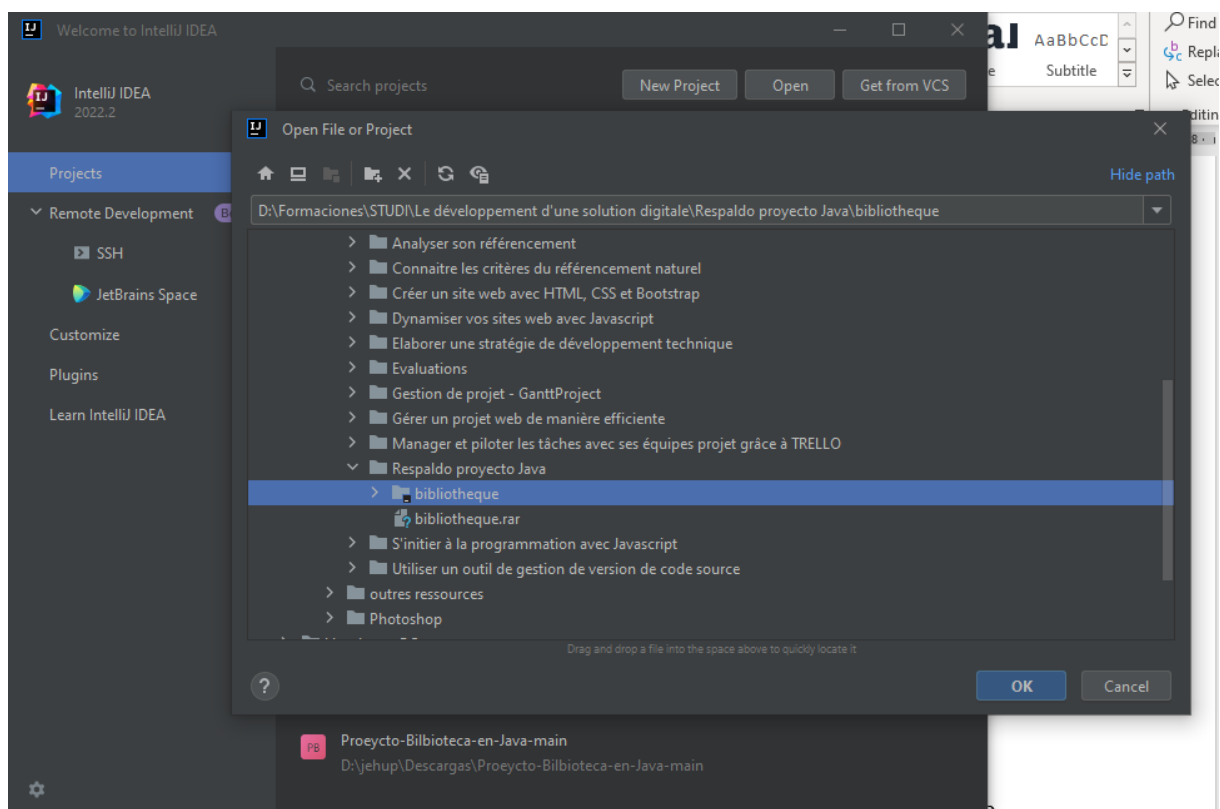


Il faudra en suite ouvrir ce projet avec votre IDE IntelliJ IDEA de la façon suivante :

Open :

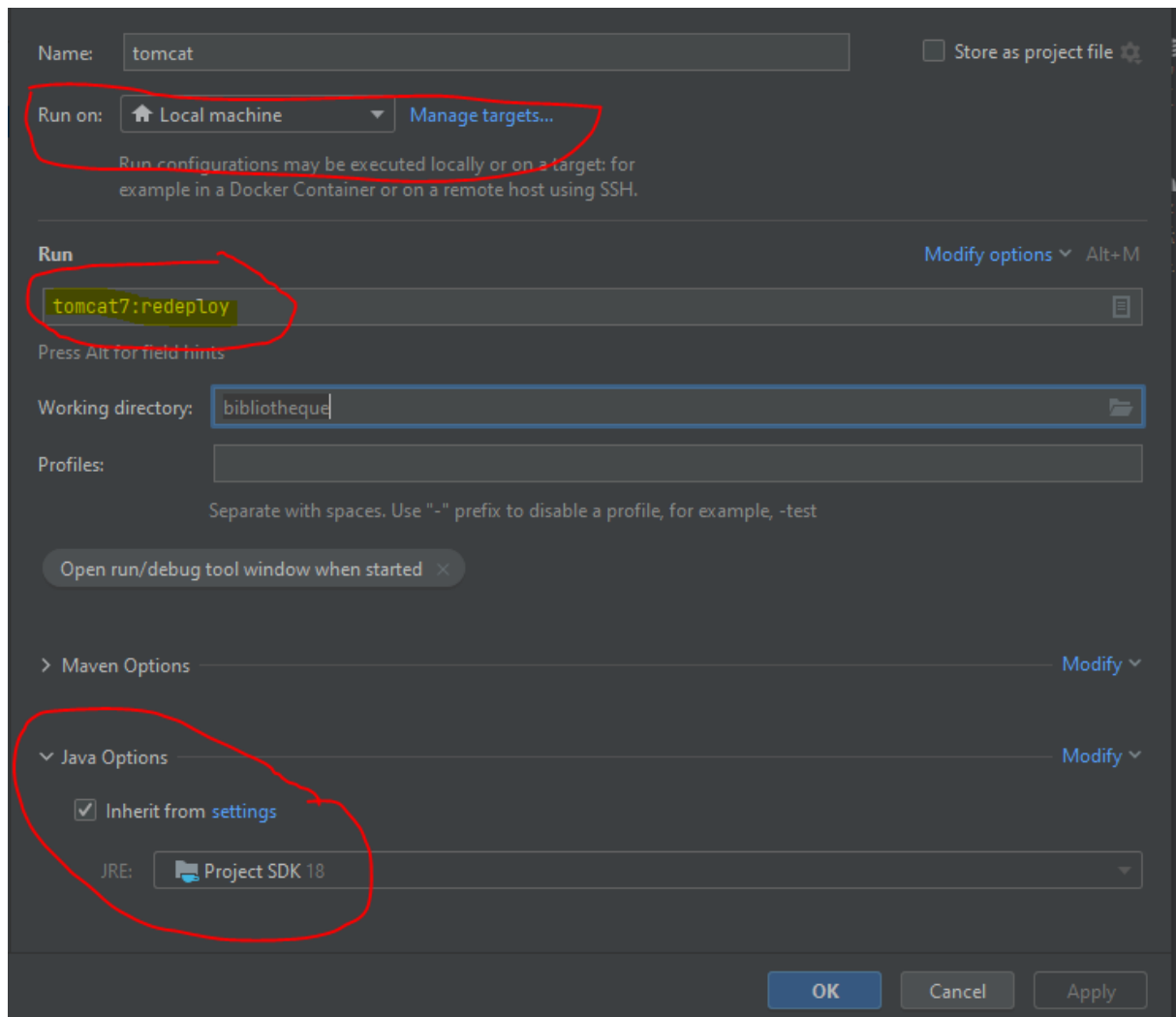


Se rendre sur l'emplacement du projet :



Editer la configuration pour le déploiement en local

La configuration du run dans l'ide doit se faire de la façon suivante :



Démarrer le serveur Apache

A l'aide de la terminal de votre IDE ou bien du cmd.exe ou powershell il faut se rendre a l'emplacement de notre dossier de notre Apache tomcat 10 et se positionner dans le dossier /bin. Il faudra en suite lancer la commande /startup.bat, cette commande va démarrer notre serveur. Le projet est désormais apte pour son exécution.

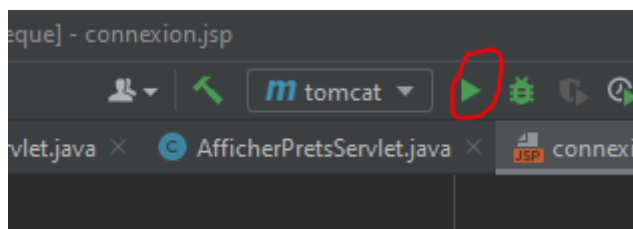
```
Terminal: C:\Window...rshell.exe x C:\Window...rshell.exe (2) x + v
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/powershell

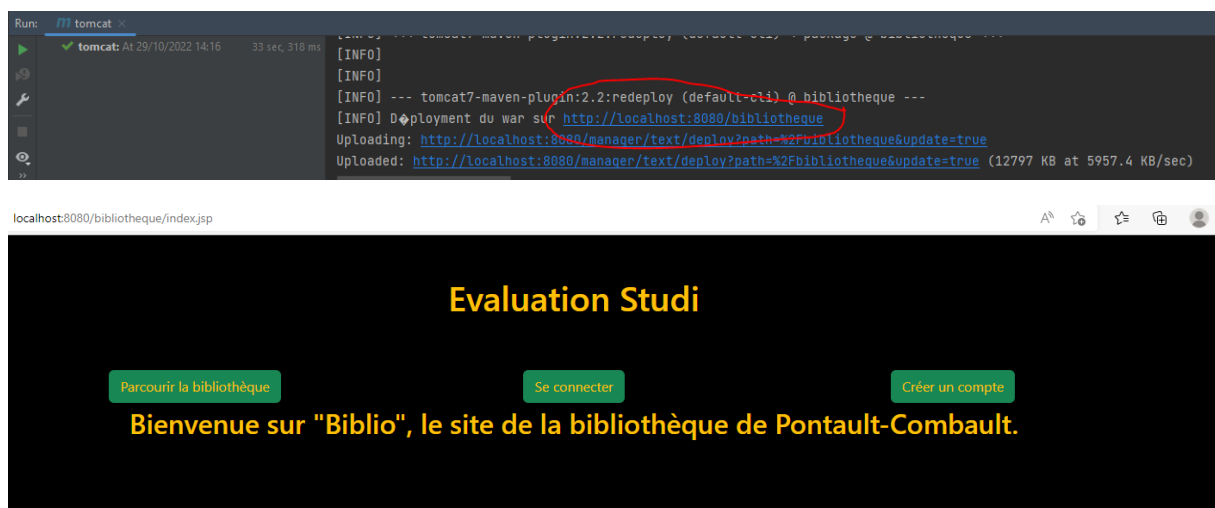
PS C:\Users\jehup\OneDrive\Escritorio\Proyectos restantes Studi\Examen
.23\bin\'
PS C:\Program Files\Java\apache-tomcat-10.0.23\bin> .\startup.bat
```

Lancer l'exécution de l'application

Il est désormais possible d'exécuter notre application, pour cela il faut faire run sur la configuration que nous avons effectué précédemment de la façon suivante :



Il suffit d'attendre quelques secondes pour obtenir le lien pour pouvoir utiliser l'application sur votre navigateur préféré.



Tests

Pour l'instant j'ai effectué les tests unitaires suivants avec JUnit

```
import java.sql.SQLException;
import java.util.List;

import static org.junit.jupiter.api.Assertions.*;

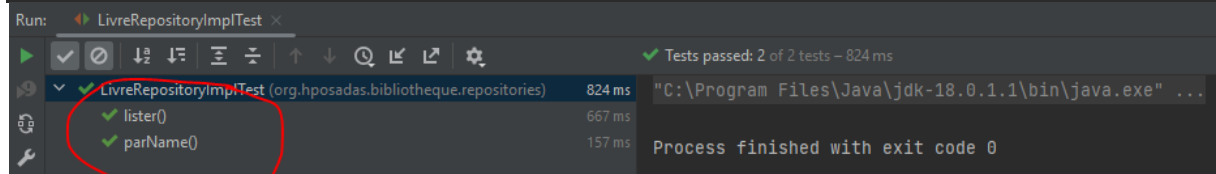
class LivreRepositoryImplTest {

    Connection conn = ConnectionDB.getConnection();

    LivreRepositoryImplTest() throws SQLException {
    }

    @Test
    void lister() {
        LivreService service = new LivreServiceImpl(conn);
        List<Livre> livres = service.lister();
        assertNotNull(livres);
    }

    @Test
    void parName() {
        LivreService service = new LivreServiceImpl(conn);
        List<Livre> livre = service.byName("Humberto");
        assertNotNull(livre);
    }
}
```



Test	Duration	Exit Code
LivreRepositoryImplTest (org.hposadas.bibliotheque.repositories)	824 ms	0
lister()	667 ms	0
parName()	157 ms	0

Process finished with exit code 0

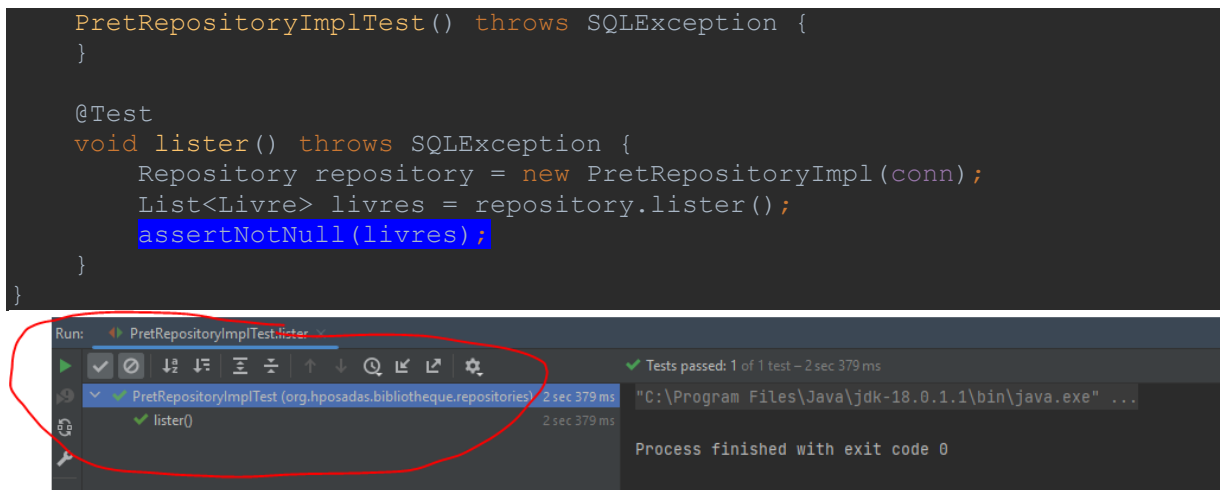
```
package org.hposadas.bibliotheque.repositories;

import org.hposadas.bibliotheque.models.Livre;
import org.hposadas.bibliotheque.utilities.ConnectionDB;
import org.junit.jupiter.api.Test;

import java.sql.Connection;
import java.sql.SQLException;
import java.util.List;

import static org.junit.jupiter.api.Assertions.*;

class PretRepositoryImplTest {
    Connection conn = ConnectionDB.getConnection();
```

Ces tests permettent de s'assurer que la ou les requêtes SQL nous rendent des résultats, c'est-à-dire que lorsque j'essaye d'accéder à l'information de la base de données, plus particulièrement aux tables utilisateurs et livres il y a un retour d'information.

Vidéo explicatif du projet

J'ai effectué un vidéo ou j'explique au détail les étapes pour l'élaboration de ce projet ainsi que le tutoriel d'utilisation à l'adresse suivante :

1^{ère} version du projet

https://drive.google.com/file/d/1-XWw3_f3PN7-hldRXefv6m0blOU5dc8i/view?usp=sharing

Version finale du projet

<https://drive.google.com/file/d/1Tnd5zZmXa-pCOSJ8FI0fVq30HGnFgyKS/view?usp=sharing>