Université numérique CHEIKH HAMIDOU KANE

Foo nekk foofu la

# Sujet d'examen : Bibliothèque universitaire hybride (REST+SOAP)

## Membres :

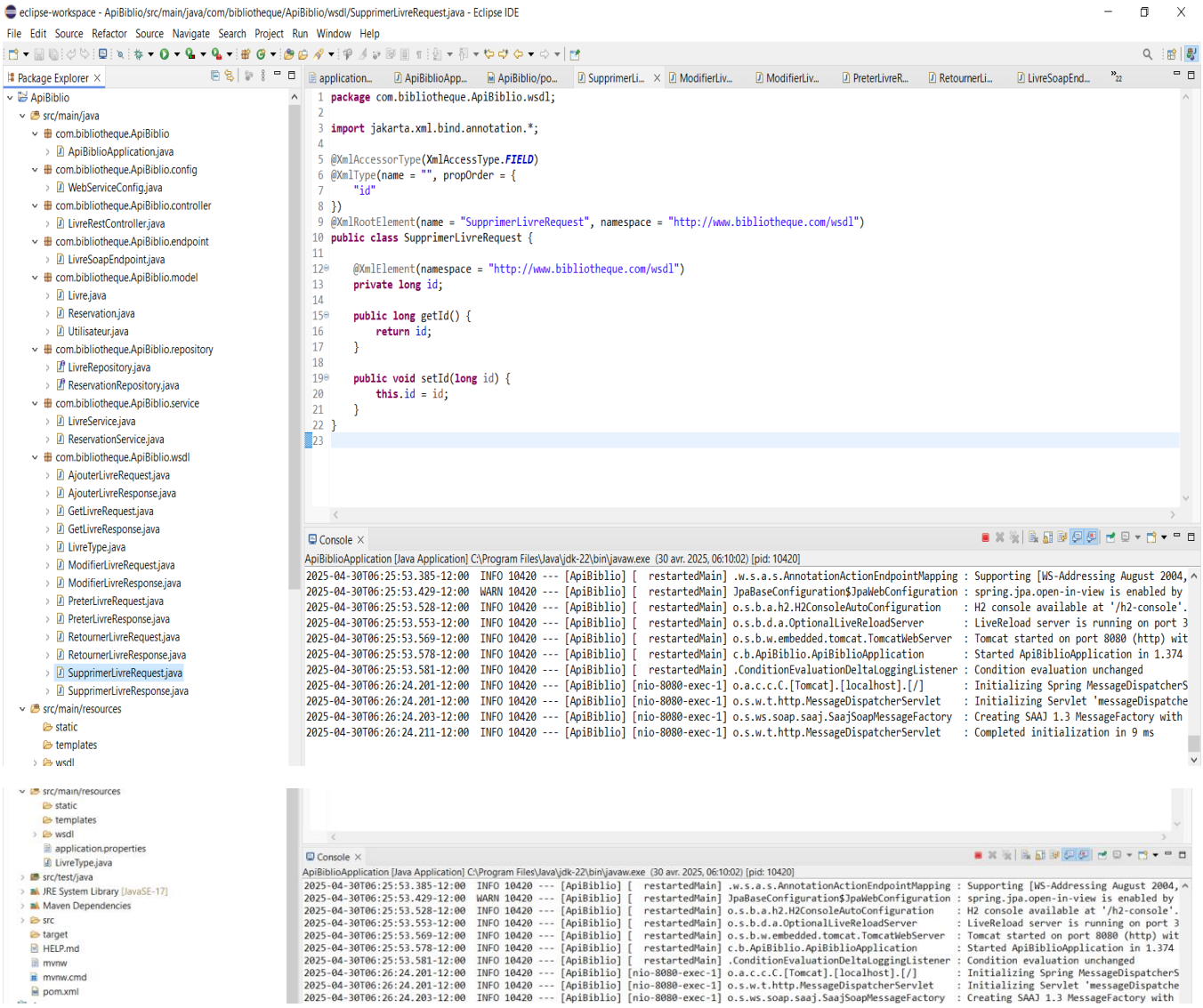
*Khadim Faye (khadim.faye2@unchk.edu.sn / N01362520191)*

*Mame Diarra Lô (Mamediarra.lo2@unchk.edu.sn /N00982420192)*

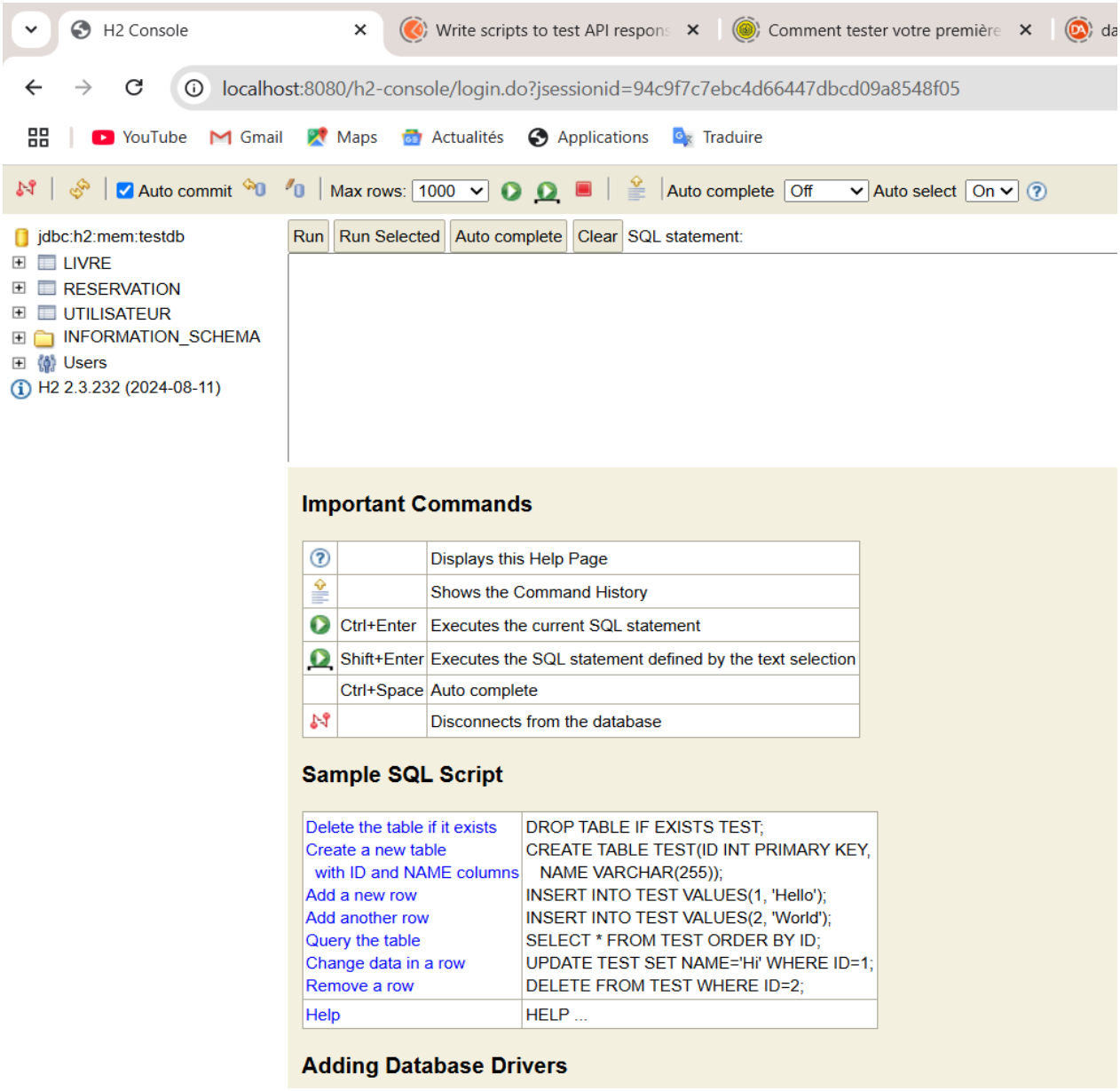
## Fonctionnalités minimales attendues :

## I. REST (Consultation* publique) :

### 1 /L'environnement de développement Eclipse Backend REST et SOAP : Spring Boot :




# 2/La connexion de le Basse de Données H2 a notre Projet :




# 3/ Les requêtes avec la Basse de Données H2 console :

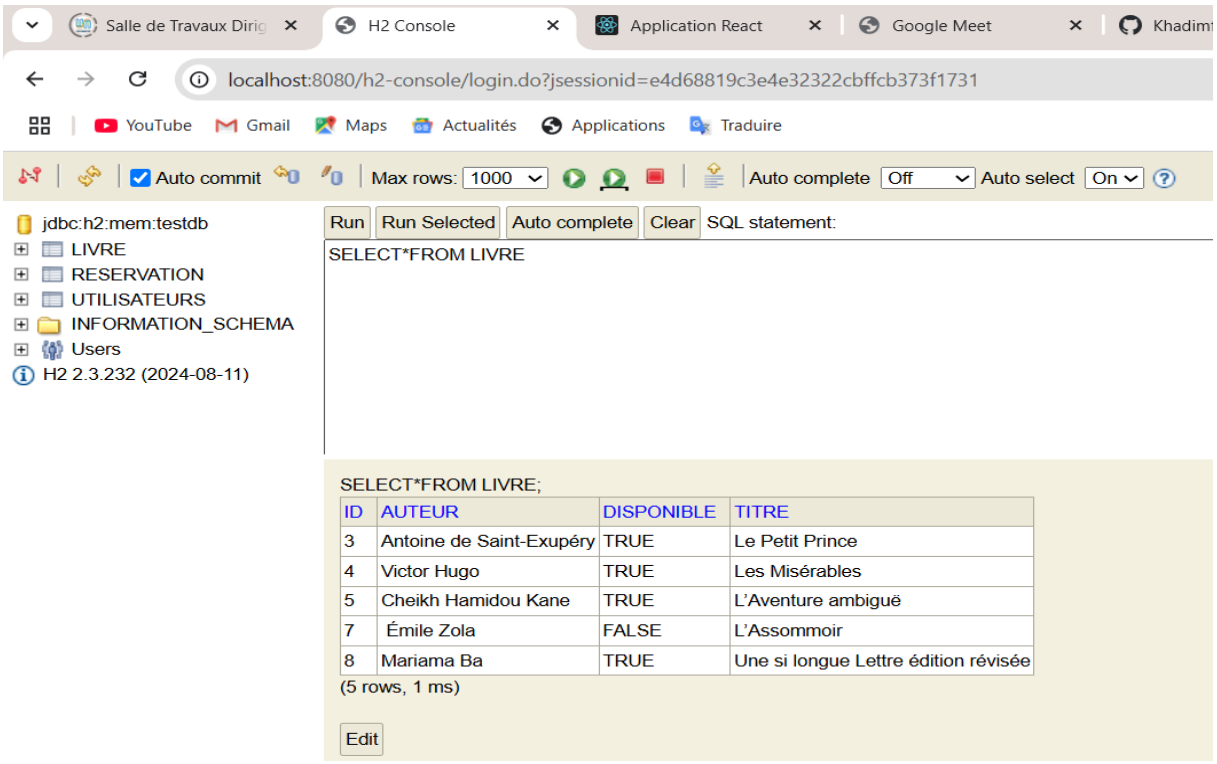
## Ajouter un livre

Run | Run Selected | Auto complete | Clear | SQL statement:

```
INSERT INTO LIVRE (TITRE, AUTEUR, DISPONIBLE) VALUES ('Le Cavalier et son ombre
', 'Boubacar Boris Diop', TRUE);
```

```
INSERT INTO LIVRE (TITRE, AUTEUR, DISPONIBLE) VALUES ('Le Cavalier et son ombre
', 'Boubacar Boris Diop', TRUE);
Update count: 1
(1 ms)
```

## Afficher les livres




## *Edit livre :

@edit SELECT*FROM LIVRE;

| Action | ID | AUTEUR | DISPONIBLE | TITRE |
|---|---|---|---|---|
| | 3 | Antoine de Saint-Exupéry | TRUE | Le Petit Prince |
| | 4 | Victor Hugo | TRUE | Les Misérables |
| | 5 | Cheikh Hamidou Kane | TRUE | L'Aventure ambiguë |
| | 7 | Émile Zola | FALSE | L'Assommoir |
| | 8 | Mariama Ba | TRUE | Une si longue Lettre édition révisée |
| | | | | |

(5 rows, 1 ms)

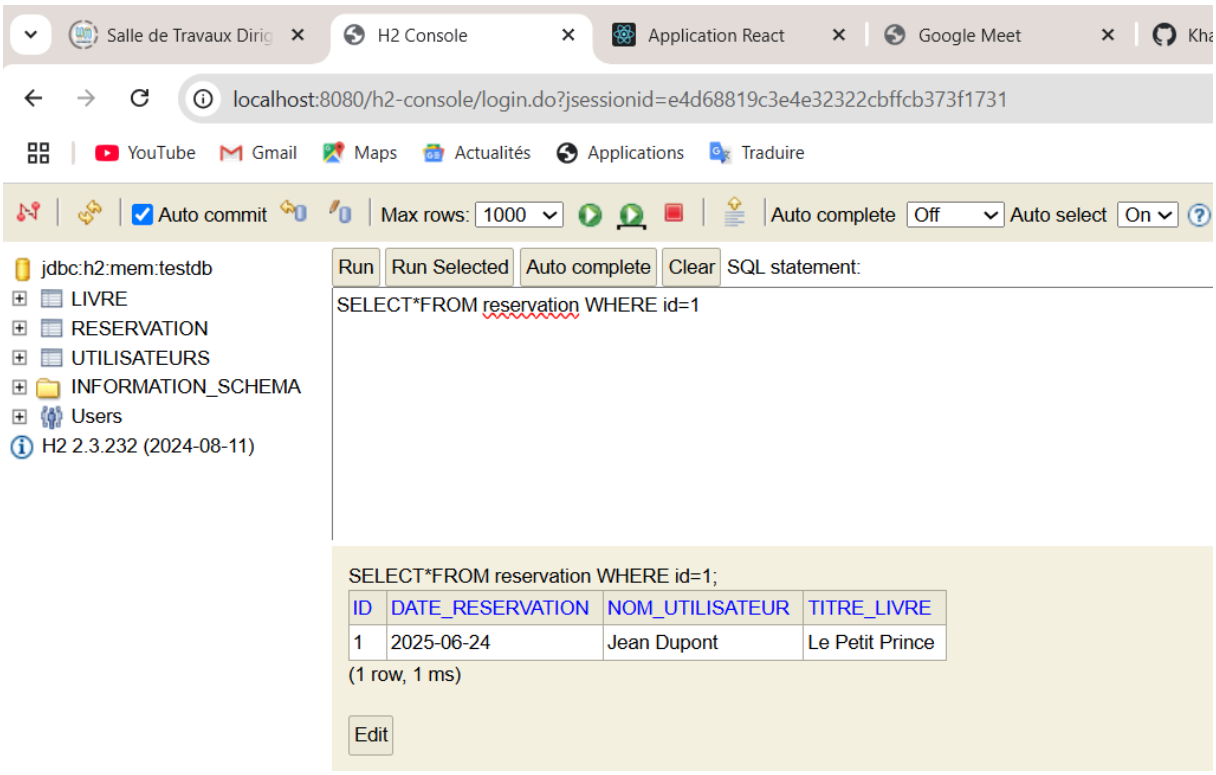
## Ajouter une réservation

Run | Run Selected | Auto complete | Clear | SQL statement:

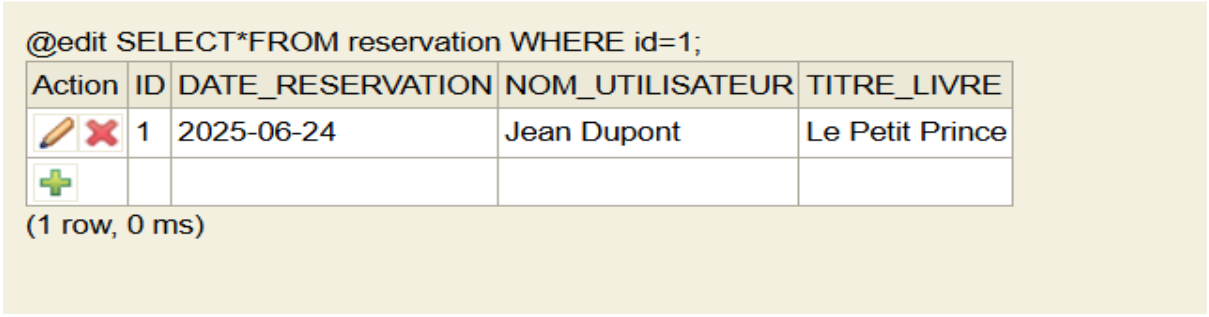
```
INSERT INTO reservation (nom_utilisateur, titre_livre, date_reservation) VALUES ('Coumba Sarr', 'Le Cavalier et son ombre', '2025-07-01');
```

INSERT INTO reservation (nom_utilisateur, titre_livre, date_reservation) VALUES ('Coumba Sarr', 'Le Cavalier et son ombre', '2025-07-01');
Update count: 1
(0 ms)

## Afficher les réservations

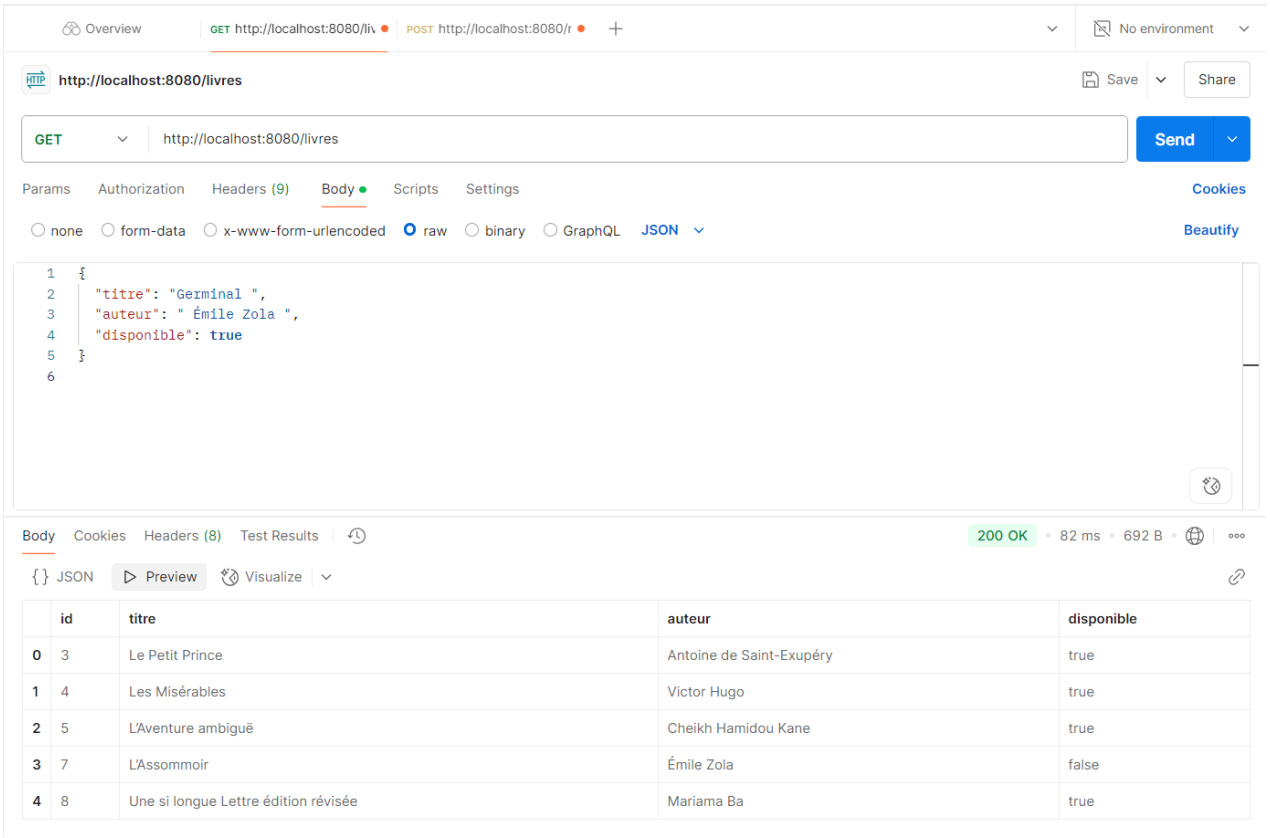



## *Edit réservation :



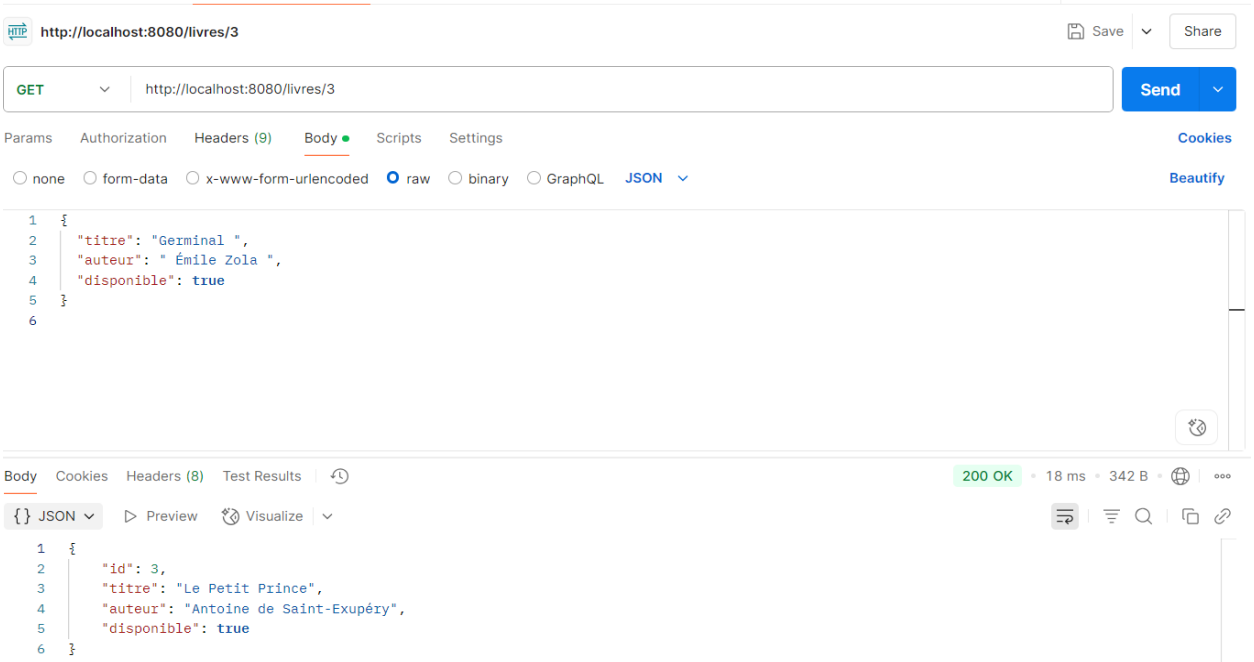

## 4/ Les requêtes avec Postman :

- **('GET /livres') : Récupération de la liste des livres**

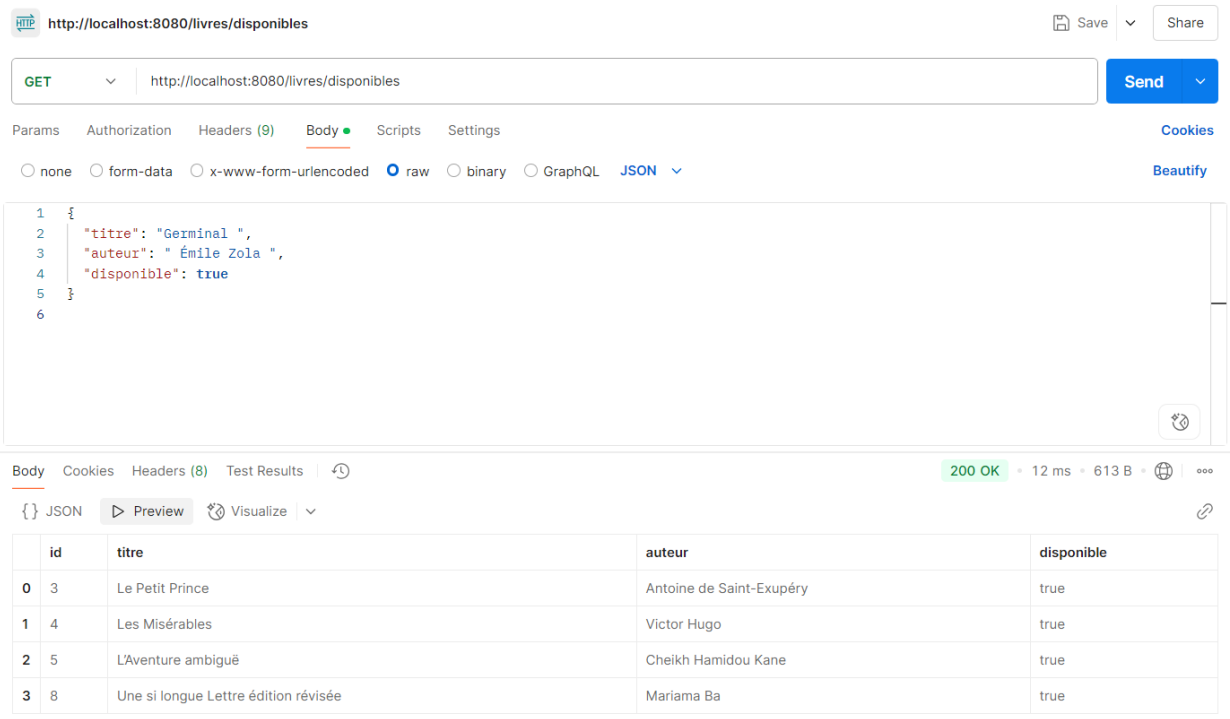



| | id | titre | auteur | disponible |
|---|---|---|---|---|
| 0 | 3 | Le Petit Prince | Antoine de Saint-Exupéry | true |
| 1 | 4 | Les Misérables | Victor Hugo | true |
| 2 | 5 | L'Aventure ambiguë | Cheikh Hamidou Kane | true |
| 3 | 7 | L'Assommoir | Émile Zola | false |
| 4 | 8 | Une si longue Lettre édition révisée | Mariama Ba | true |

- **('GET /livres/{id}') :** Afficher les informations d'un livre

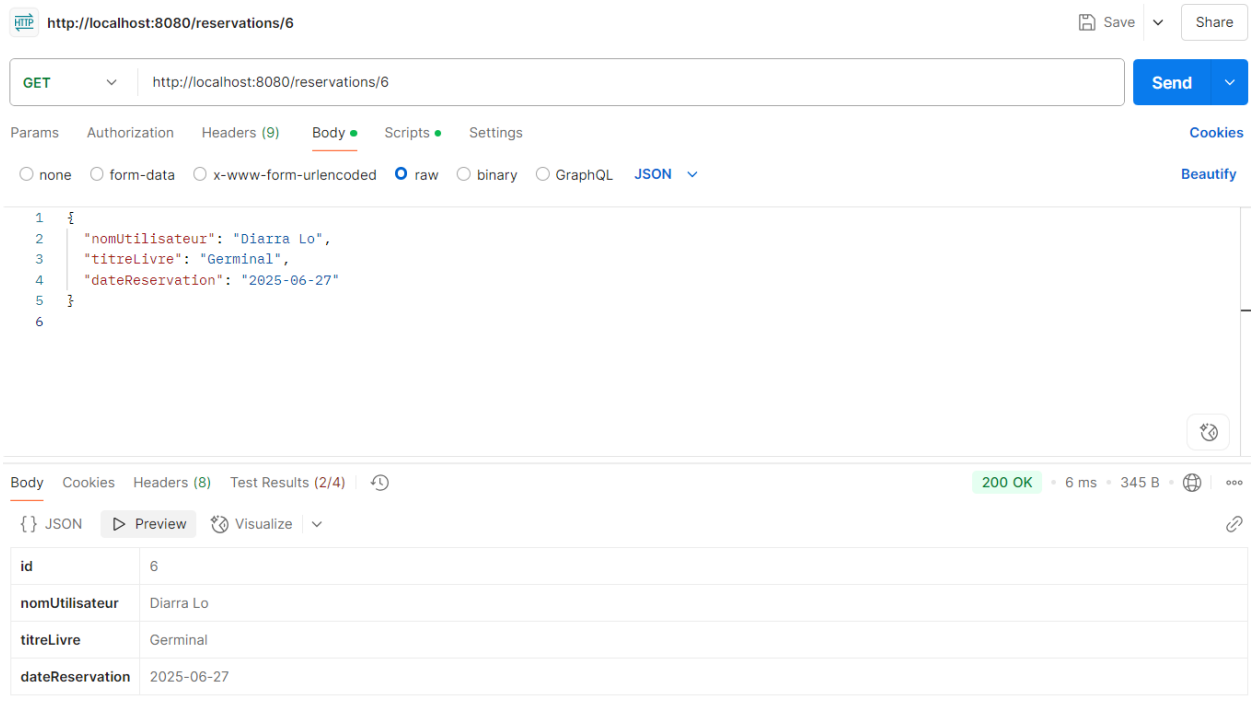



- **('GET /livres/disponibles') :** Retourne les livres disponibles c'est-à-dire non prêtés et non réservés.



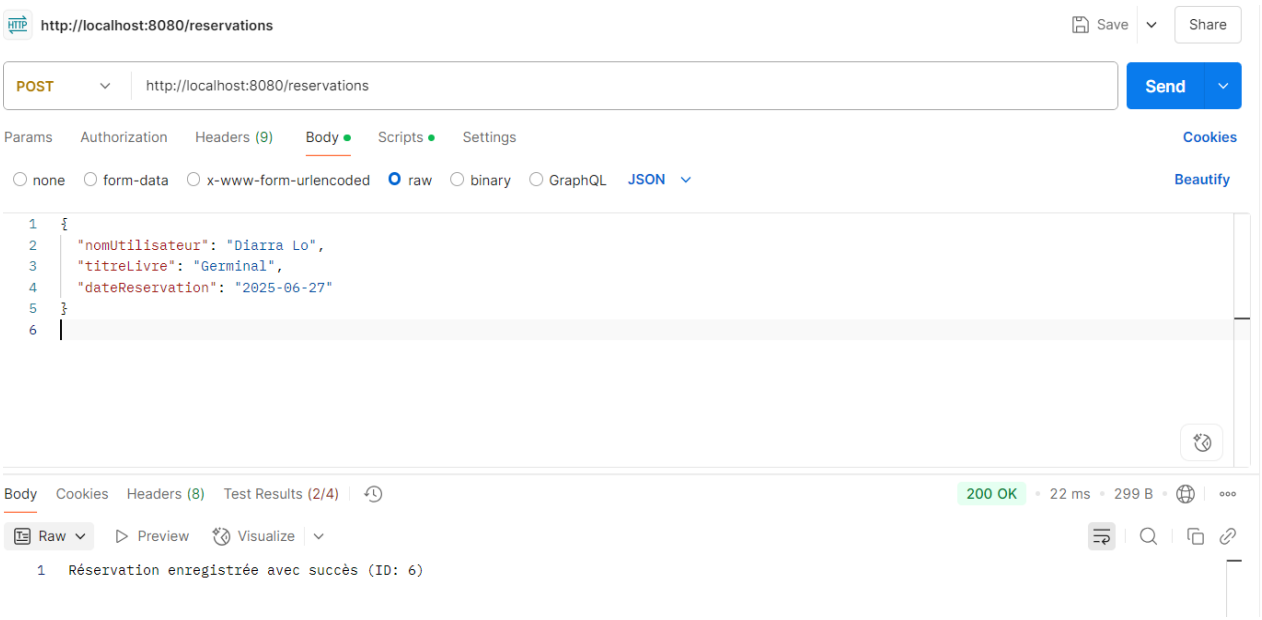

| | id | titre | auteur | disponible |
|---|---|---|---|---|
| 0 | 3 | Le Petit Prince | Antoine de Saint-Exupéry | true |
| 1 | 4 | Les Misérables | Victor Hugo | true |
| 2 | 5 | L'Aventure ambiguë | Cheikh Hamidou Kane | true |
| 3 | 8 | Une si longue Lettre édition révisée | Mariama Ba | true |

- **('GET /reservations/{id}') : Pour le suivi d'une réservation**




| | |
|---|---|
| id | 6 |
| nomUtilisateur | Diarra Lo |
| titreLivre | Germinal |
| dateReservation | 2025-06-27 |

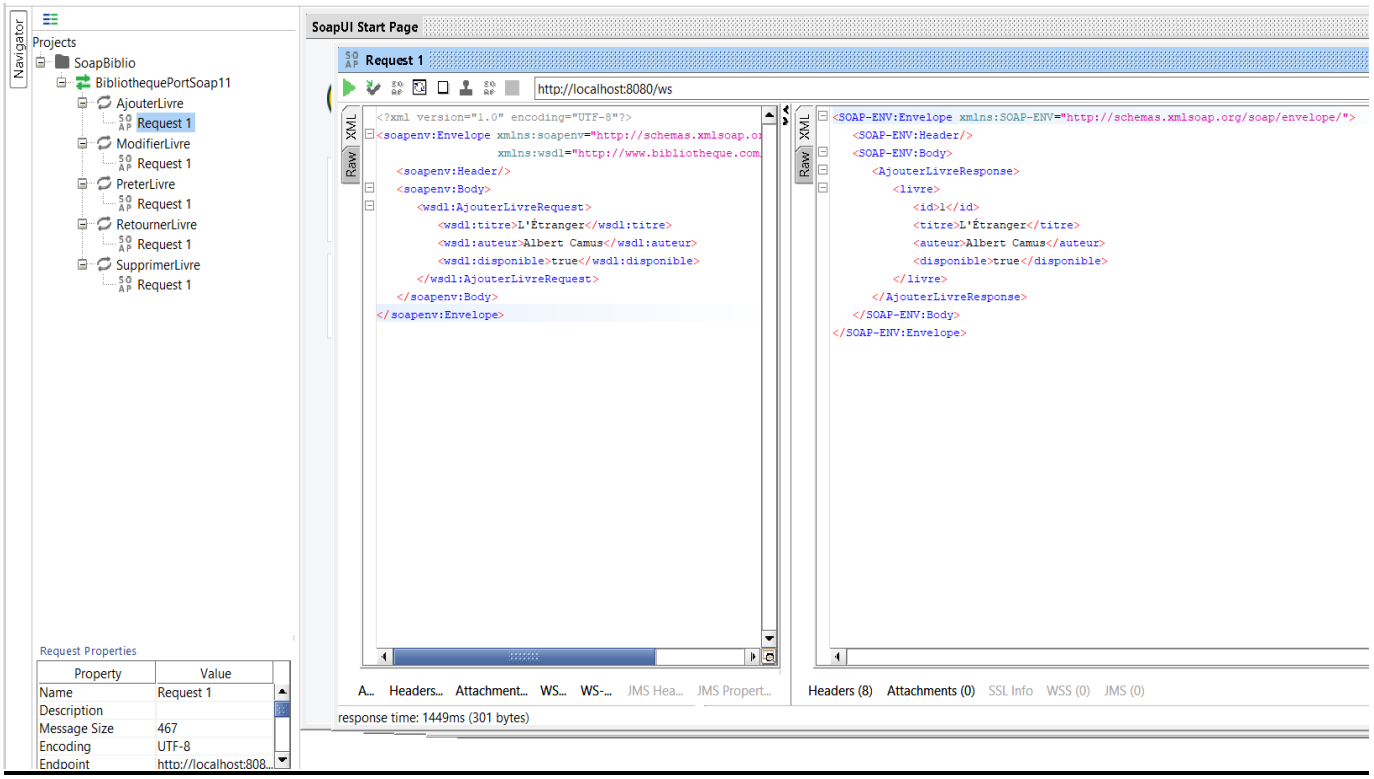
- **('POST /reservations') : Réservation d'un livre donné à une période précise (Passer par un endpoint SOAP)**



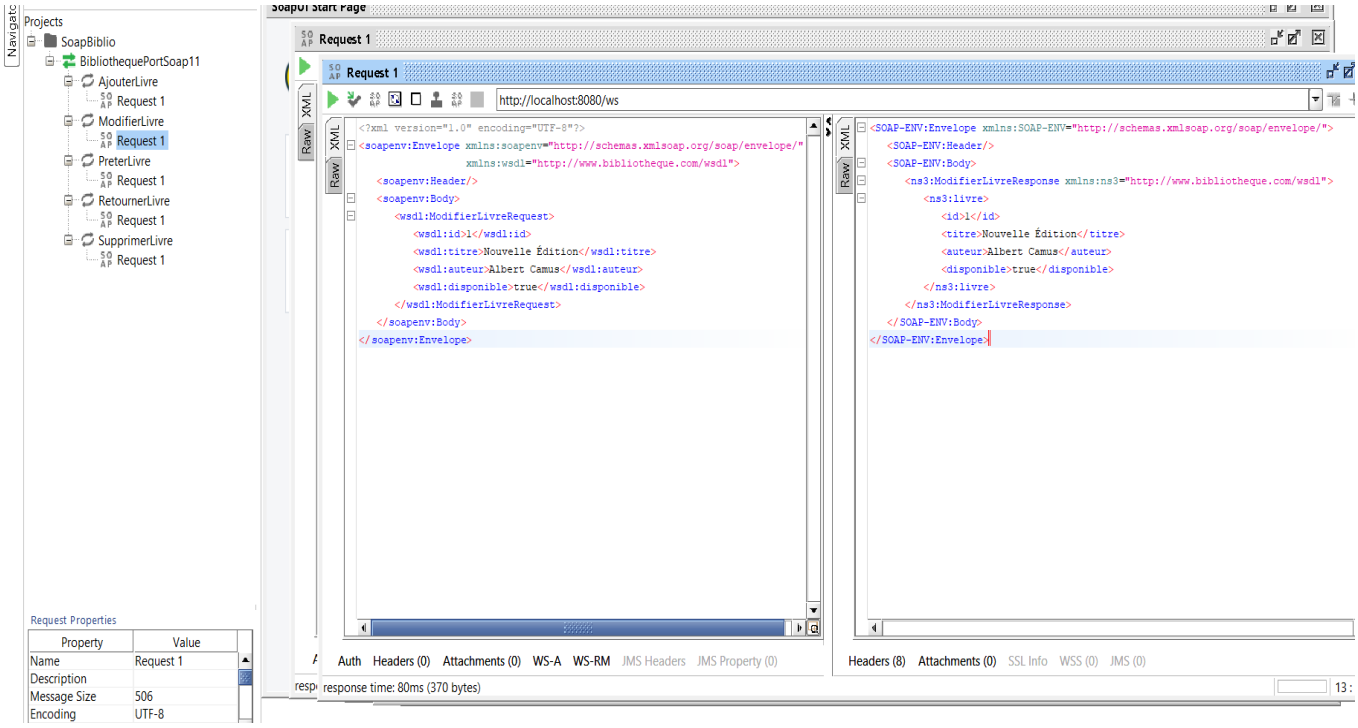

# II. SOAP (Gestion interne) :

## 5/Les requêtes lors du Test avec SOAPUI :

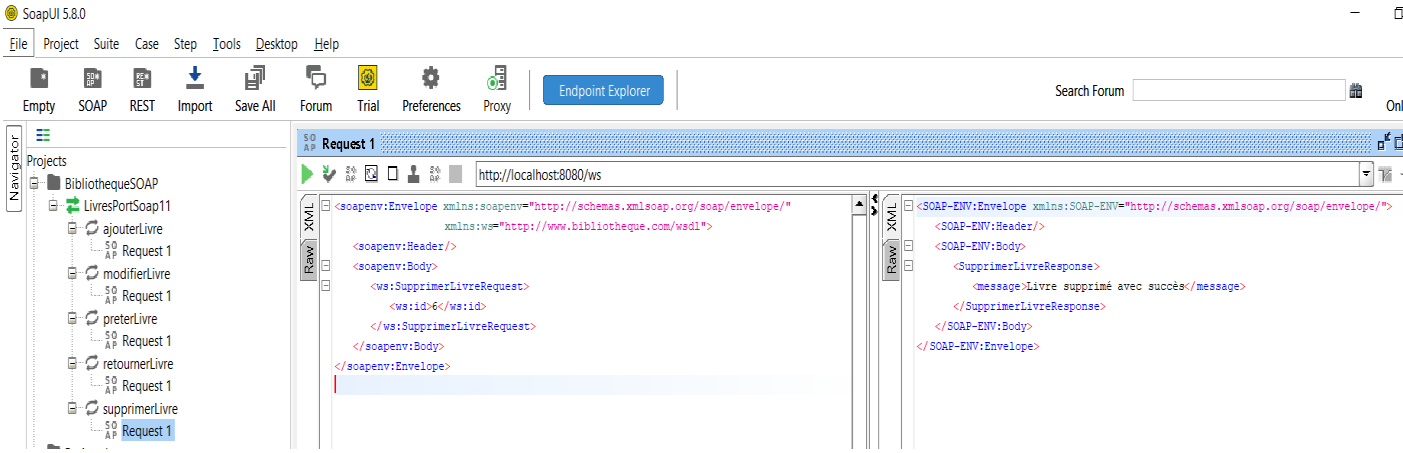
- **AjouterLivre(Livre livre)**



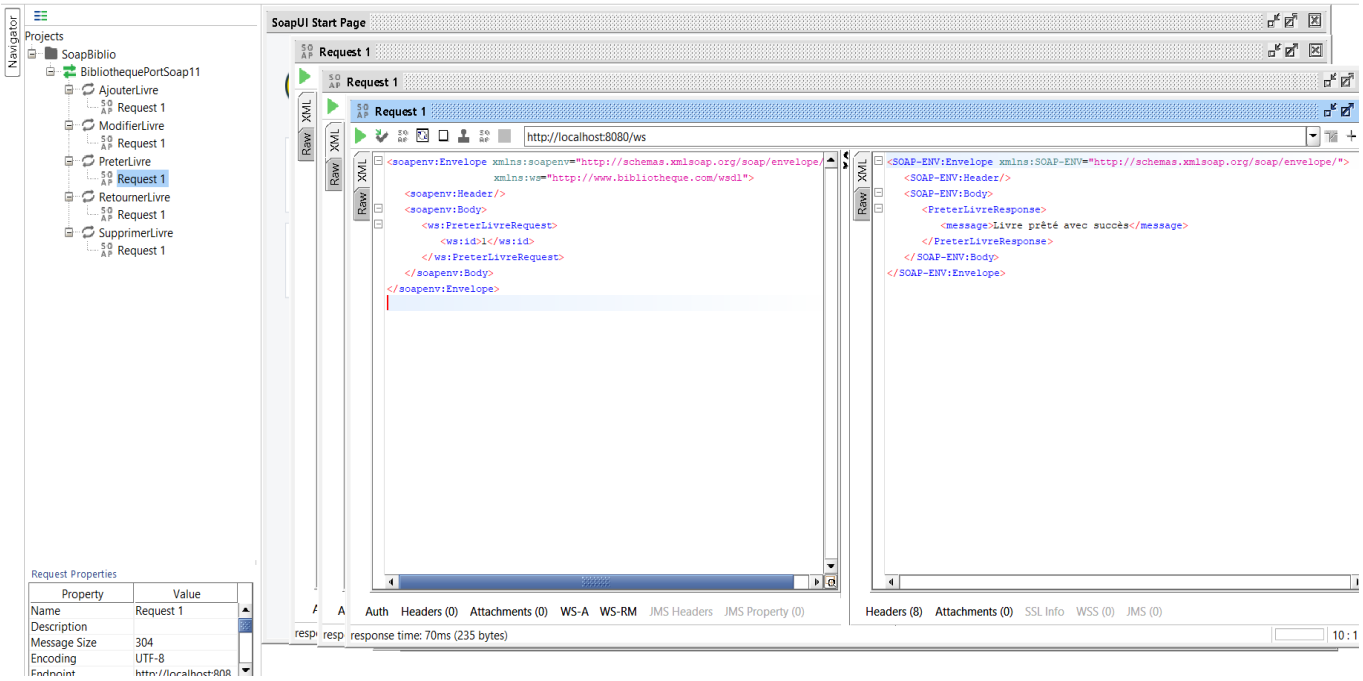

- **ModifierLivre(Long livreID, Livre livre)**



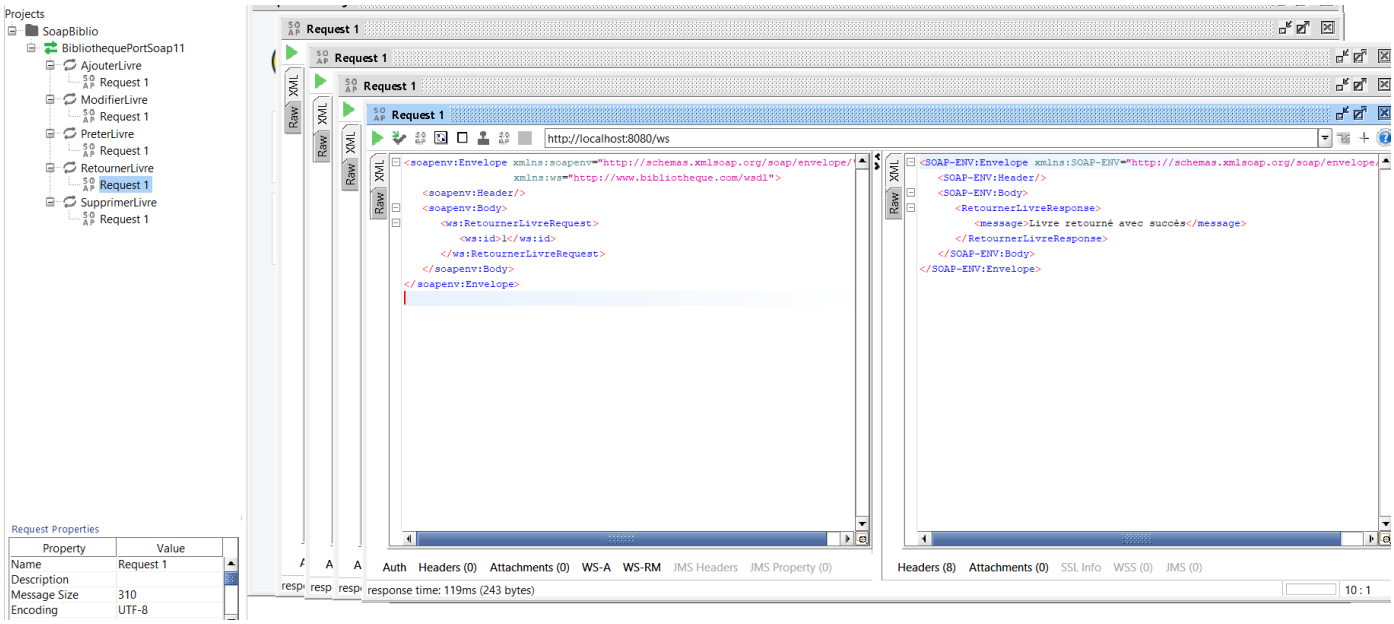

- **SupprimerLivre(Long livreID)**



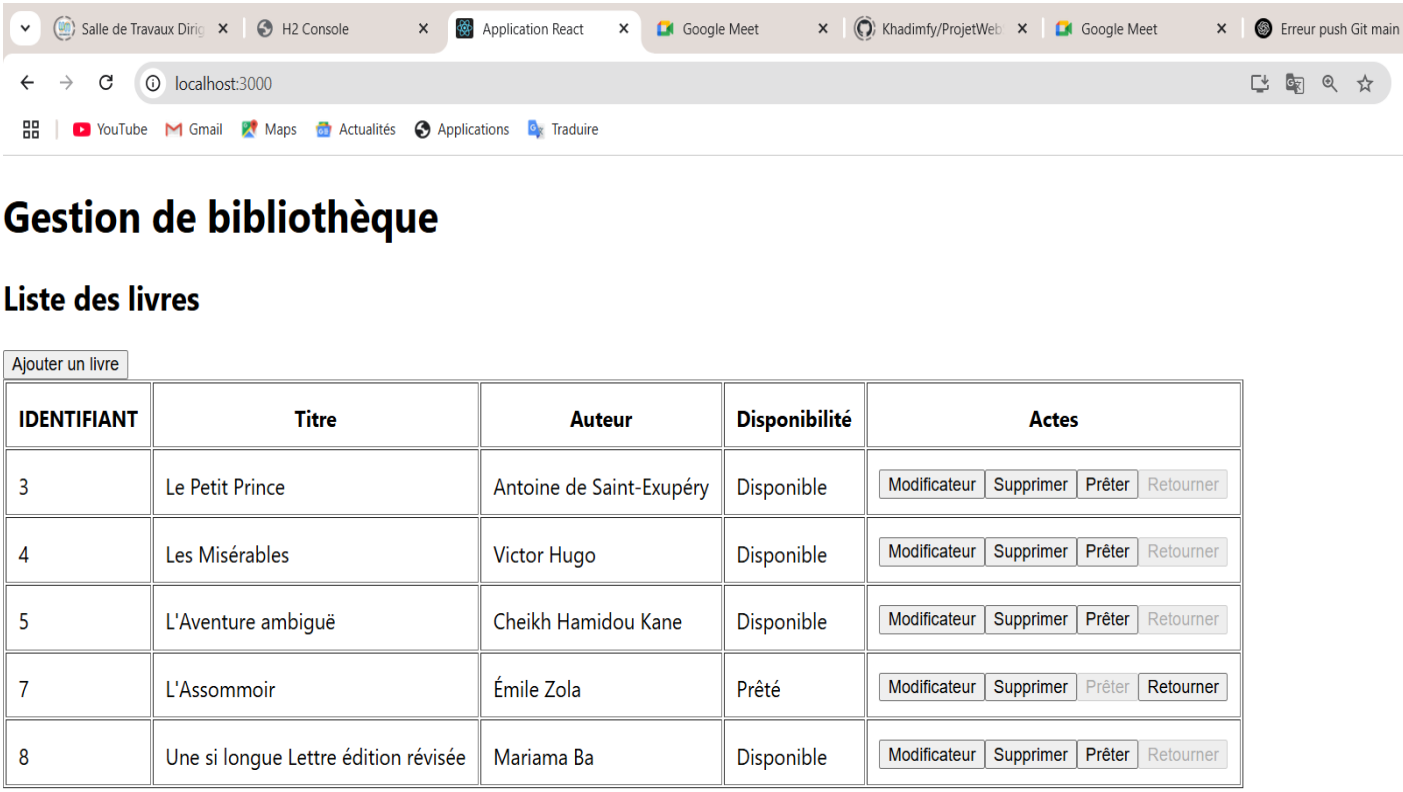

- **PreterLivre(Long userID, Long livreID)**




- **RetournerLivre(Long userID, Long livreID)**




## 6/L'utilisation du Frontend React pour (Ajouter, Modifier, Supprimer, Prêter et Retourner) des Livres :




## Retours d'expérience

Ce projet nous a permis de mettre en pratique l'intégration d'un backend SOAP développé avec Spring Boot à un frontend React. Il nous a apporté une compréhension plus approfondie du fonctionnement des services web SOAP et de la manière dont ils peuvent être consommés depuis une interface moderne. Nous avons également appris à manipuler des outils comme SoapUI pour tester les endpoints SOAP, et à gérer les échanges de données XML à l'aide de fichiers XSD.

La transition entre le backend Java et le frontend JavaScript a renforcé notre capacité à travailler dans des environnements hétérogènes, à gérer les communications HTTP (via un proxy Express), et à organiser efficacement un projet en couches — modèle, service, contrôleur et endpoint.

## Difficultés rencontrées

1. Gestion des namespaces SOAP

- L'une des erreurs les plus fréquentes concernait le nommage incorrect ou l'absence de namespace dans les requêtes SOAP.

- Solution : Nous avons soigneusement défini le targetNamespace dans le fichier XSD et veillé à ce qu'il soit correctement repris dans les annotations Java des classes générées et des endpoints.

2. Adaptation des points de terminaison (endpoints)

- Spring Web Services ne reconnaissait pas certains endpoints à cause de méthodes mal annotées ou de noms de requêtes erronés.

- Solution : Nous avons systématiquement utilisé @Endpoint combiné avec @PayloadRoot, en précisant le bon namespace et le localPart correspondant aux requêtes.

3. Consommation du backend SOAP avec React

- Étant donné que React ne traite pas nativement le XML, il a été nécessaire de créer un proxy côté serveur.

- Solution : Nous avons mis en place un serveur Express (index.js) en Node.js pour faire la conversion entre les requêtes SOAP XML et des appels REST/JSON exploitables par le frontend React.

4. Réservations et relations entre entités :

- La fonctionnalité de réservation a exigé une gestion rigoureuse des relations entre entités telles que Livre, Utilisateur et Réservation.

- Solution : Nous avons utilisé des classes DTO pour organiser proprement les données échangées, et défini des services métier clairs pour centraliser la logique de réservation.

5. Erreurs HTTP 500 liées à JAXB

- Plusieurs erreurs critiques provenaient de la non-conformité au schéma XSD ou de l'absence de classes JAXB.

- Solution : Nous avons corrigé le fichier XSD, régénéré les classes à l'aide de jaxb2-maven-plugin, et ajusté les requêtes XML pour respecter scrupuleusement la structure définie.

§FIN§