

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

### **Membres :**

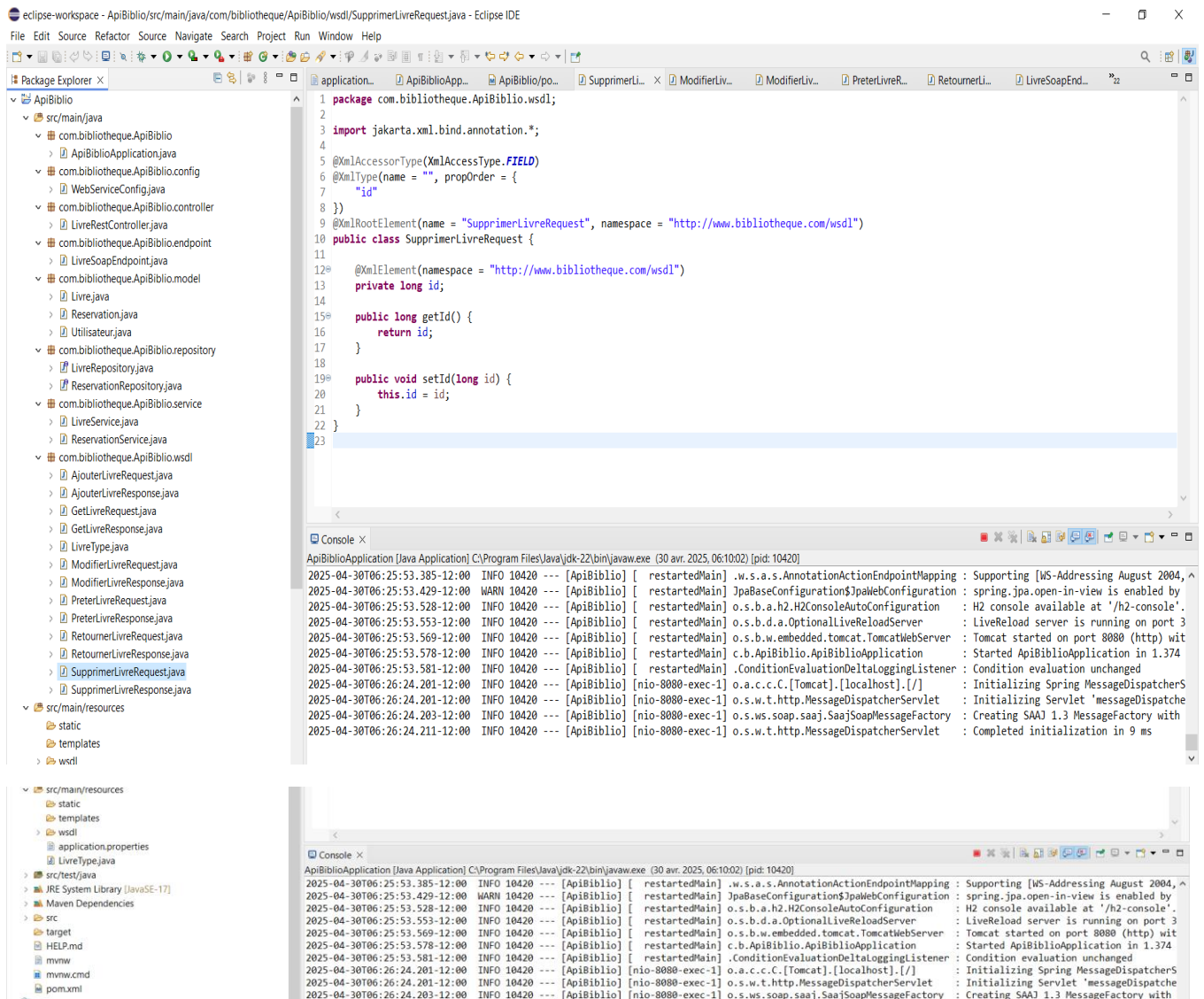
*Khadim Faye ([khadim.faye2@unchk.edu.sn](mailto:khadim.faye2@unchk.edu.sn) / N01362520191)*

*Mame Diarra Lô ([Mamediarra.lo2@unchk.edu.sn](mailto: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 :

H2 Console

Write scripts to test API respons

Comment tester votre première

da

localhost:8080/h2-console/login.do?jsessionid=94c9f7c7ebc4d66447dbcd09a8548f05

YouTube

Gmail

Maps

Actualités

Applications

Traduire

Auto commit

Max rows: 1000

Auto complete

Off

Auto select

On

jdbc:h2:mem:testdb

LIVRE

RESERVATION

UTILISATEUR

INFORMATION\_SCHEMA

Users

H2 2.3.232 (2024-08-11)

Run

Run Selected

Auto complete

Clear

SQL statement:

?

Displays this Help Page

Shows the Command History

Ctrl+Enter

Executes the current SQL statement

Shift+Enter

Executes the SQL statement defined by the text selection

Ctrl+Space

Auto complete

Disconnects from the database

Sample SQL Script

Delete the table if it exists

Create a new table with ID and NAME columns

Add a new row

Add another row

Query the table

Change data in a row

Remove a row

Help

DROP TABLE IF EXISTS TEST;

CREATE TABLE TEST(ID INT PRIMARY KEY, NAME VARCHAR(255));

INSERT INTO TEST VALUES(1, 'Hello');

INSERT INTO TEST VALUES(2, 'World');

SELECT \* FROM TEST ORDER BY ID;


UPDATE TEST SET NAME='Hi' WHERE ID=1;

DELETE FROM TEST WHERE ID=2;

HELP ...

Adding Database Drivers

### 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

Salle de Travaux Dirigé x H2 Console x Application React x Google Meet x Khadim

localhost:8080/h2-console/login.do?jsessionid=e4d68819c3e4e32322cbffcb373f1731

YouTube Gmail Maps Actualités Applications Traduire

Auto commit Max rows: 1000 Auto complete Off Auto select On

jdbc:h2:mem:testdb

- LIVRE
- RESERVATION
- UTILISATEURS
- INFORMATION\_SCHEMA
- Users
- H2 2.3.232 (2024-08-11)

Run Run Selected Auto complete Clear SQL statement:

SELECT \* FROM LIVRE

SELECT \* FROM LIVRE;

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)

Edit

\*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

Salle de Travaux Dirigés x H2 Console x Application React x Google Meet x Kha

localhost:8080/h2-console/login.do?jsessionId=e4d68819c3e4e32322cbffcb373f1731

YouTube Gmail Maps Actualités Applications Traduire

Auto commit Max rows: 1000 Auto complete Off Auto select On

jdbc:h2:mem:testdb

- LIVRE
- RESERVATION
- UTILISATEURS
- INFORMATION\_SCHEMA
- Users
- H2 2.3.232 (2024-08-11)

Run Run Selected Auto complete Clear SQL statement:

SELECT\*FROM reservation WHERE id=1

SELECT\*FROM reservation WHERE id=1;




ID	DATE_RESERVATION	NOM_UTILISATEUR	TITRE_LIVRE
1	2025-06-24	Jean Dupont	Le Petit Prince

(1 row, 1 ms)

Edit

## \*Edit réservation :

@edit SELECT\*FROM reservation WHERE id=1;

Action	ID	DATE_RESERVATION	NOM_UTILISATEUR	TITRE_LIVRE
 	1	2025-06-24	Jean Dupont	Le Petit Prince
				

(1 row, 0 ms)

## 4/ Les requêtes avec Postman :

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

Overview GET http://localhost:8080/liv POST http://localhost:8080/r No environment

http://localhost:8080/livres Save Share

GET http://localhost:8080/livres Send

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "titre": "Germinal ",
3   "auteur": " Émile Zola ",
4   "disponible": true
5 }
6
```

Body Cookies Headers (8) Test Results 200 OK 82 ms 692 B

{ } JSON Preview Visualize

	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

http://localhost:8080/livres/3 Save Share

GET http://localhost:8080/livres/3 Send

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "titre": "Germinal ",
3   "auteur": " Émile Zola ",
4   "disponible": true
5 }
6
```

Body Cookies Headers (8) Test Results 200 OK 18 ms 342 B

{ } JSON Preview Visualize

```
1 {
2   "id": 3,
3   "titre": "Le Petit Prince",
4   "auteur": "Antoine de Saint-Exupéry",
5   "disponible": true
6 }
```

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

HTTP <http://localhost:8080/livres/disponibles> Save Share

GET <http://localhost:8080/livres/disponibles> Send

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "titre": "Germinal ",
3   "auteur": " Émile Zola ",
4   "disponible": true
5 }
6
```

Body Cookies Headers (8) Test Results 200 OK • 12 ms • 613 B

{ } JSON Preview Visualize

	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

HTTP <http://localhost:8080/reservations/6> Save Share

GET <http://localhost:8080/reservations/6> Send

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "nomUtilisateur": "Diarra Lo",
3   "titreLivre": "Germinal",
4   "dateReservation": "2025-06-27"
5 }
6
```

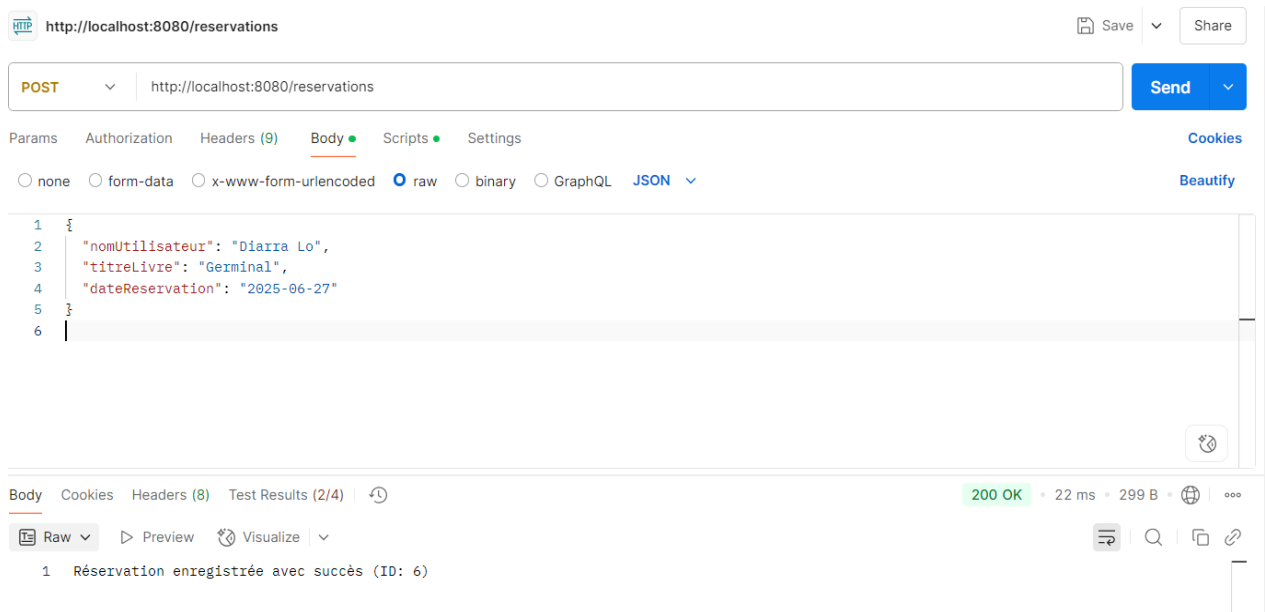
Body Cookies Headers (8) Test Results (2/4) 200 OK • 6 ms • 345 B

{ } JSON Preview Visualize

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)

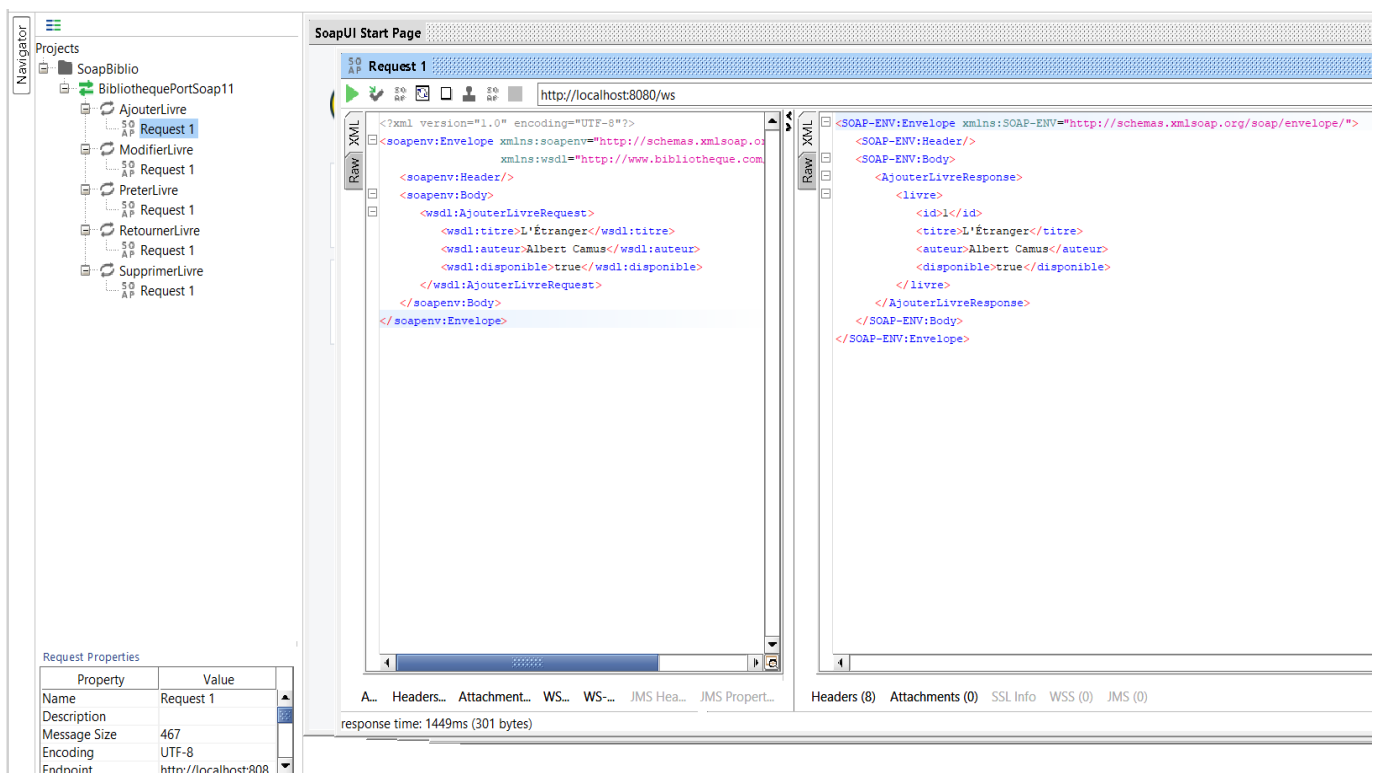




## II. SOAP (Gestion interne) :

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

#### • AjouterLivre(Livre livre)



#### • ModifierLivre(Long livreID, Livre livre)

soapui start page

Request 1

http://localhost:8080/ws

Raw XML

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:wsdl="http://www.bibliotheque.com/wsdl">
  <soapenv:Header/>
  <soapenv:Body>
    <wsdl:ModifierLivreRequest>
      <wsdl:id>1</wsdl:id>
      <wsdl:titre>Nouvelle Édition</wsdl:titre>
      <wsdl:auteur>Albert Camus</wsdl:auteur>
      <wsdl:disponible>true</wsdl:disponible>
    </wsdl:ModifierLivreRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	506
Encoding	UTF-8

Headers (8) Attachments (0) SSL Info WSS (0) JMS (0)

response time: 80ms (370 bytes)

Raw XML

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns3:ModifierLivreResponse xmlns:ns3="http://www.bibliotheque.com/wsdl">
      <id>1</id>
      <titre>Nouvelle Édition</titre>
      <auteur>Albert Camus</auteur>
      <disponible>true</disponible>
    </ns3:livre>
  </ns3:ModifierLivreResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## • SupprimerLivre(Long livreID)

SoapUI 5.8.0

File Project Suite Case Step Tools Desktop Help

Empty SOAP REST Import Save All Forum Trial Preferences Proxy Endpoint Explorer Search Forum

Request 1

http://localhost:8080/ws

Raw XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ws="http://www.bibliotheque.com/wsdl">
  <soapenv:Header/>
  <soapenv:Body>
    <ws:SupprimerLivreRequest>
      <ws:id>6</ws:id>
    </ws:SupprimerLivreRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Raw XML

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <SupprimerLivreResponse>
      <message>Livre supprimé avec succès</message>
    </SupprimerLivreResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## • PreterLivre(Long userID, Long livreID)

SoapUI Start Page

Request 1

Request 1

http://localhost:8080/ws

```
<?xml version='1.0' encoding='UTF-8'>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ws="http://www.bibliotheque.com/wsdl">
  <soapenv:Header/>
  <soapenv:Body>
    <ws:PreterLivreRequest>
      <ws:id>1</ws:id>
    </ws:PreterLivreRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<?xml version='1.0' encoding='UTF-8'>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <PreterLivreResponse>
      <message>Livre prêté avec succès</message>
    </PreterLivreResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	304
Encoding	UTF-8
Endpoint	http://localhost:8080

Headers (8) Attachments (0) SSL Info WSS (0) JMS (0)

response time: 70ms (235 bytes)

## • RetournerLivres(Long userID, Long livreID)

Request 1

Request 1

http://localhost:8080/ws

```
<?xml version='1.0' encoding='UTF-8'>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ws="http://www.bibliotheque.com/wsdl">
  <soapenv:Header/>
  <soapenv:Body>
    <ws:RetournerLivresRequest>
      <ws:id>1</ws:id>
    </ws:RetournerLivresRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<?xml version='1.0' encoding='UTF-8'>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <RetournerLivresResponse>
      <message>Livres retournés avec succès</message>
    </RetournerLivresResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	310
Encoding	UTF-8

Headers (8) Attachments (0) SSL Info WSS (0) JMS (0)

response time: 119ms (243 bytes)

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

# Gestion de bibliothèque

## Liste des livres

Ajouter un livre

IDENTIFIANT	Titre	Auteur	Disponibilité	Actes
3	Le Petit Prince	Antoine de Saint-Exupéry	Disponible	<div>Modificateur Supprimer Prêter Retourner</div>
4	Les Misérables	Victor Hugo	Disponible	<div>Modificateur Supprimer Prêter Retourner</div>
5	L'Aventure ambiguë	Cheikh Hamidou Kane	Disponible	<div>Modificateur Supprimer Prêter Retourner</div>
7	L'Assommoir	Émile Zola	Prêté	<div>Modificateur Supprimer Prêter Retourner</div>
8	Une si longue Lettre édition révisée	Mariama Ba	Disponible	<div>Modificateur Supprimer Prêter Retourner</div>

## 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.

SFINS