

Présentation du projet Web Services

Thème :

Système de gestion de bibliothèque

Présenté par :

Amadou Sow (N00731920191)

Moussa Mané (N01392720191)

Tuteur :

Jiga Sène

Année universitaire :

2024-2025

1. Introduction

Objectif du projet :

Ce projet consiste à développer une application de gestion de bibliothèque combinant deux types de services web :

- API REST pour la consultation publique des livres.
- Service SOAP pour la gestion interne (opérations CRUD).

2. Technologies utilisées

- Backend : Java, Spring Boot
- Commande principale : `mvn spring-boot:run` (compile, démarre l'application, expose les endpoints REST/SOAP, met à disposition les fichiers WSDL/XSD)
- Base de données : H2 (en mémoire)



Fig 1: Lancement du projet

Fig 2: Outil utilisé

3. Architecture générale

L'application repose sur une architecture à trois couches :

- Contrôleur : REST + SOAP
- Service : logique métier
- Repository : accès base de données

Avec configuration pour Swagger, SOAP, et la base de données, et un mapping entre DTO et entités.

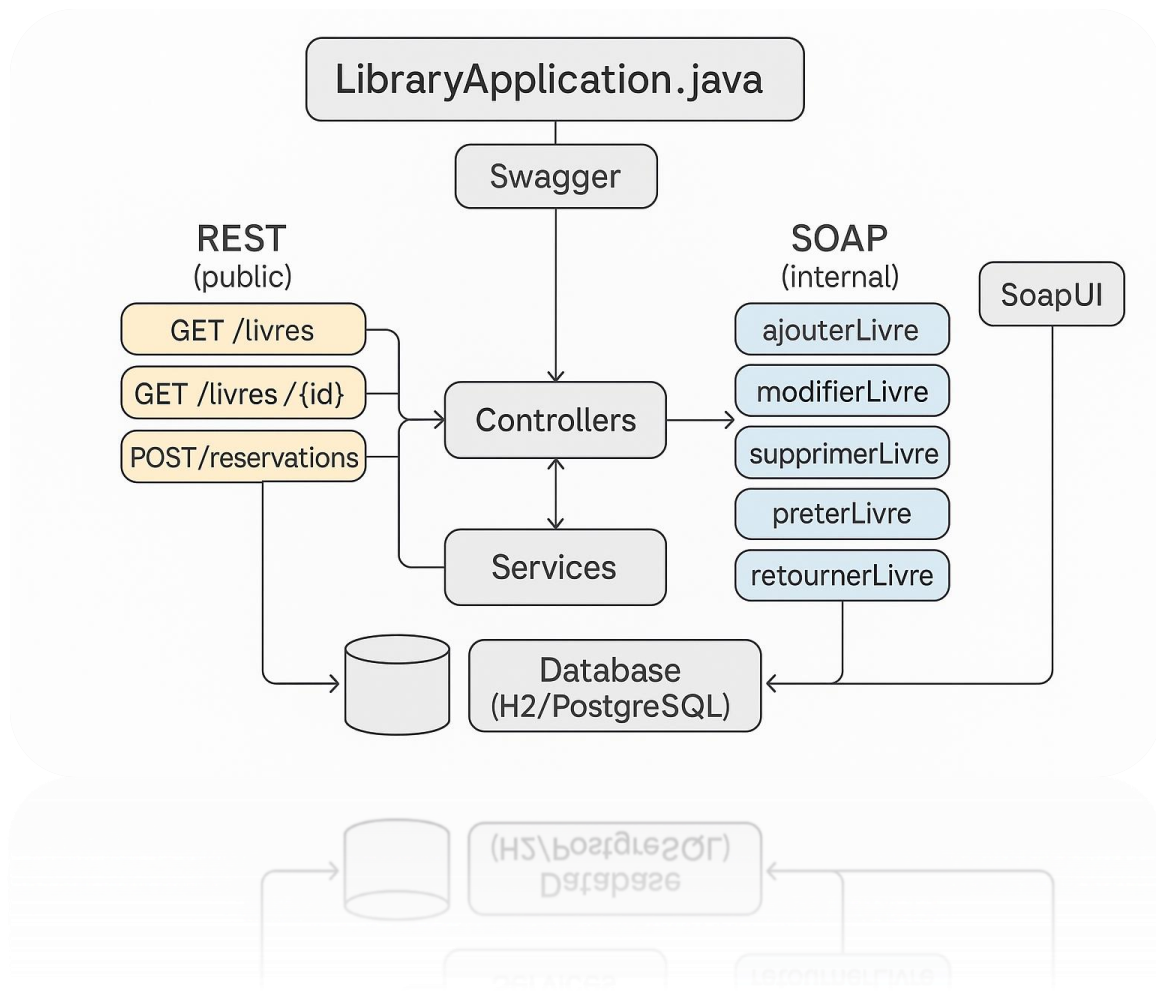


Fig 3: Architecture du projet

4. Résultat final attendu

- Console H2 : <http://localhost:8081/h2-console>
- Endpoint SOAP : <http://localhost:8081/ws>
- Tests possibles via SoapUI et Postman

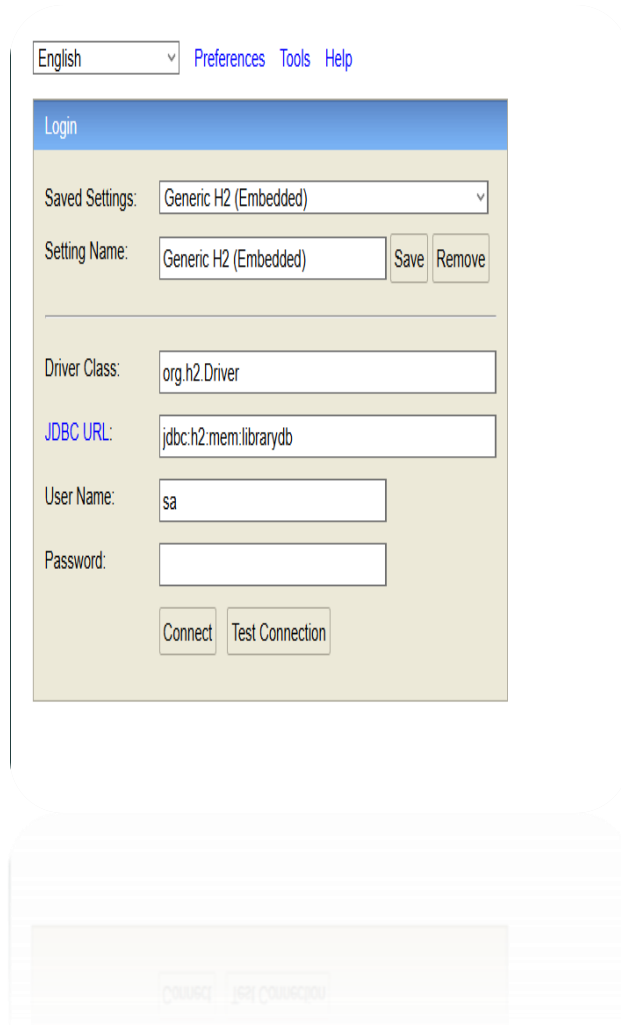


Fig 4: Console H2

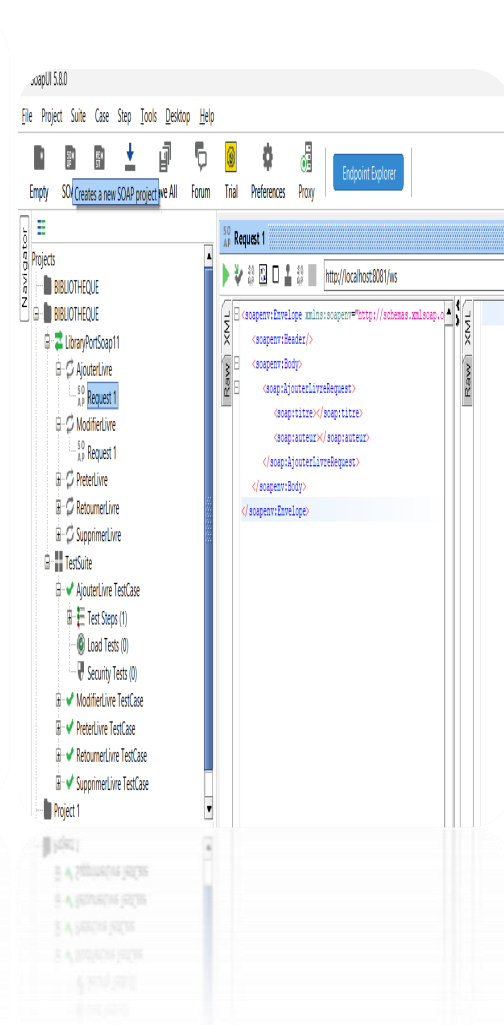


Fig 5: Endpoint SOAP

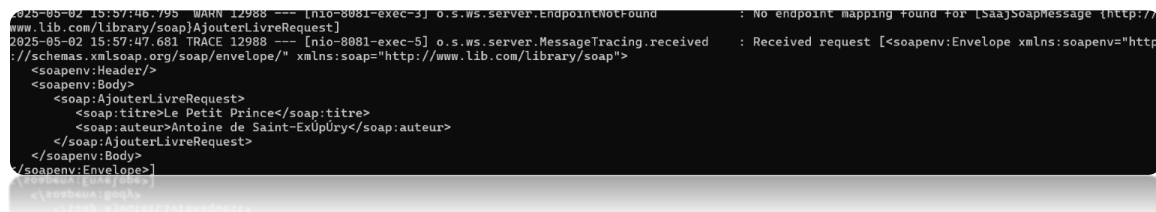
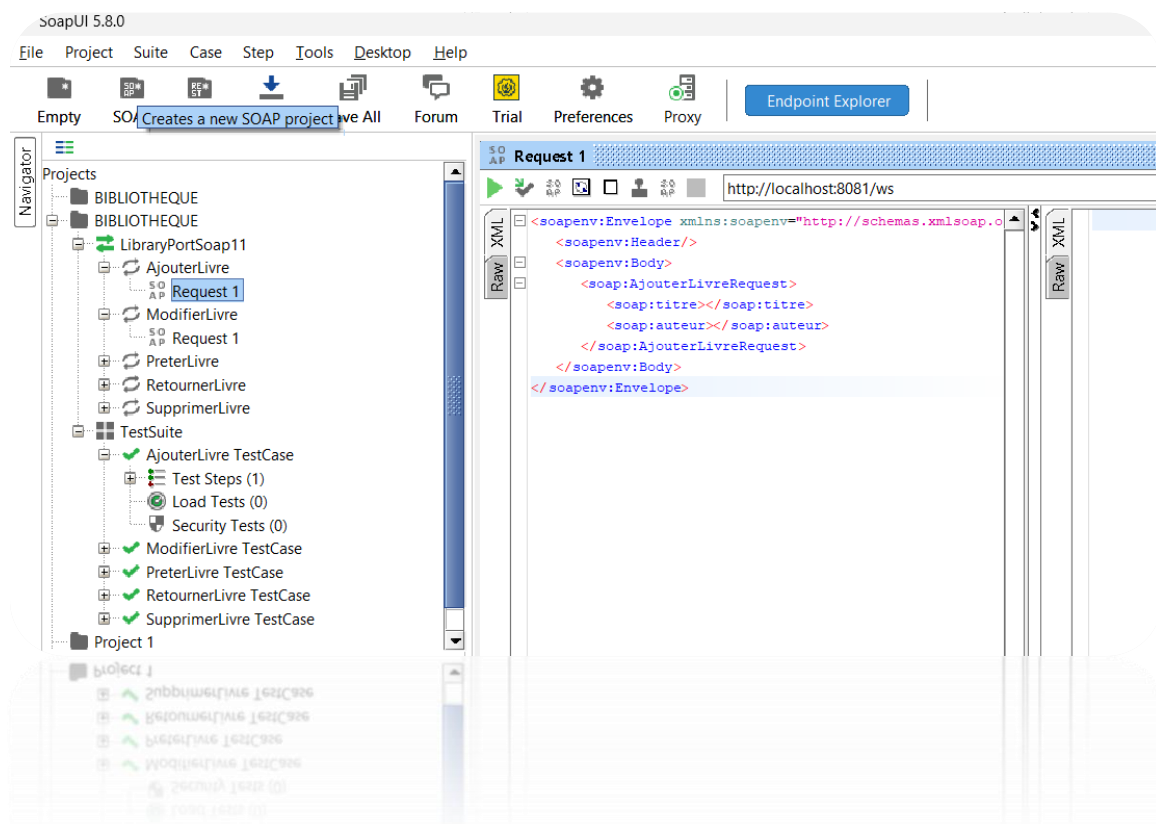
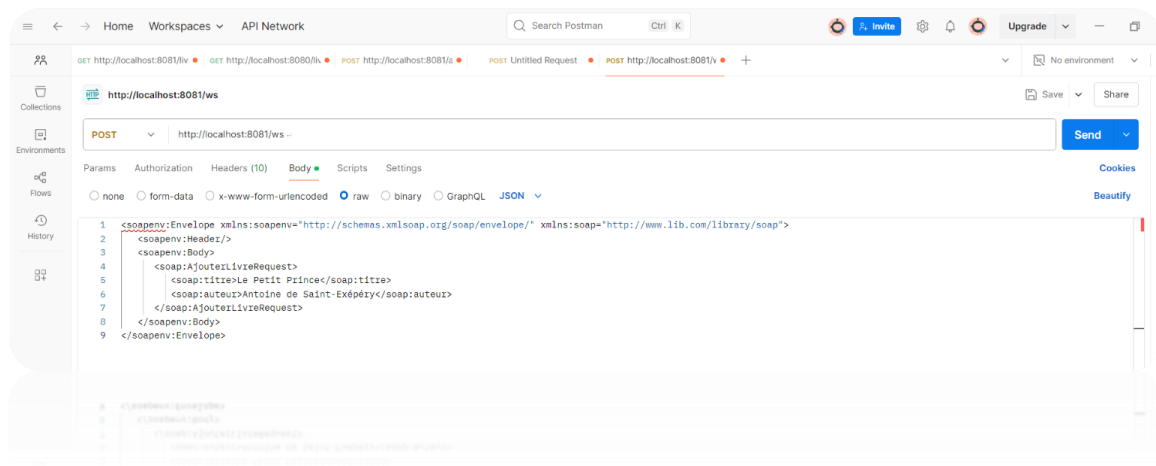


Fig 6: Tests possibles via SoapUI



```

2025-05-02 16:23:21.745 WARN 12988 --- [io-8081-exec-10] o.s.ws.server.EndpointNotFound : No endpoint mapping found for [SaaJSoapMessage {http://
www.lib.com/library/soap}AjouterLivreRequest]
2025-05-02 16:23:23.110 TRACE 12988 --- [nio-8081-exec-1] o.s.ws.server.MessageTracing.received : Received request [<soapenv:Envelope xmlns:soapenv="http
://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://www.lib.com/library/soap">
  <soapenv:Header/>
  <soapenv:Body>
    <soap:AjouterLivreRequest>
      <soap:titre>Le Petit Prince</soap:titre>
      <soap:auteur>Antoine de Saint-Exupéry</soap:auteur>
    </soap:AjouterLivreRequest>
  </soapenv:Body>
</soapenv:Envelope>]
2025-05-02 16:23:23.110 WARN 12988 --- [nio-8081-exec-1] o.s.ws.server.EndpointNotFound : No endpoint mapping found for [SaaJSoapMessage {http://
www.lib.com/library/soap}AjouterLivreRequest]

```

Fig 7: Tests possibles via Postman

5. Fonctionnalités principales

API REST :

- GET /api/livres → liste des livres
- GET /api/livres/{id} → détail d'un livre

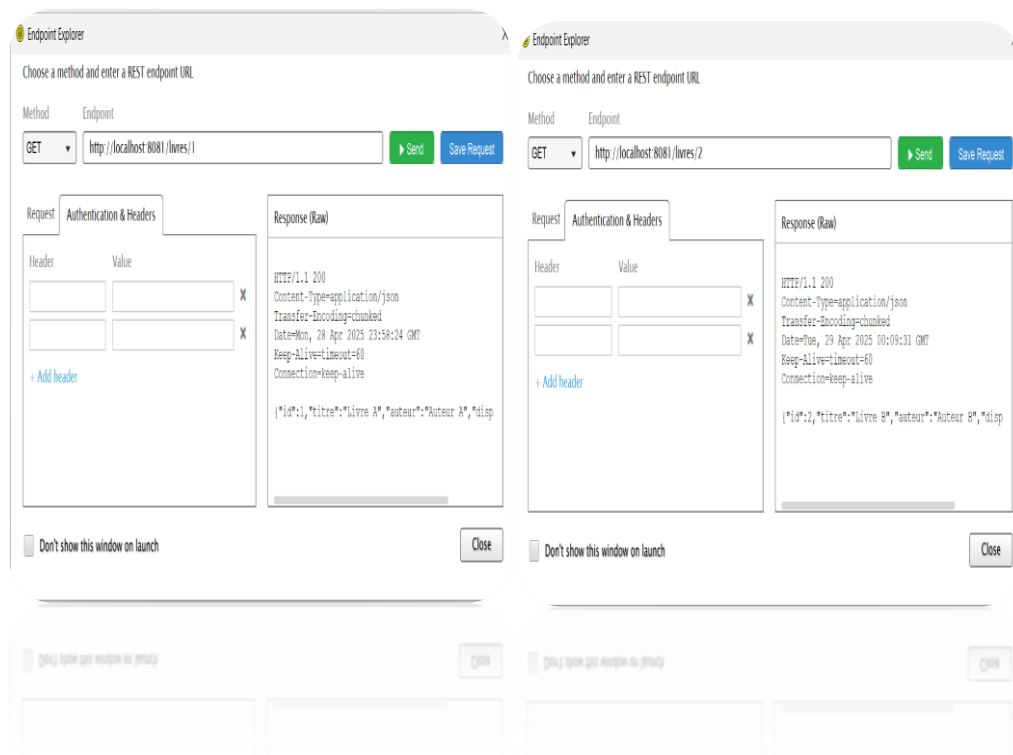


Fig 8: API REST

Service SOAP :

- Ajouter un livre
- Modifier un livre
- Supprimer un livre
- Prêter un livre
- Retourner un livre

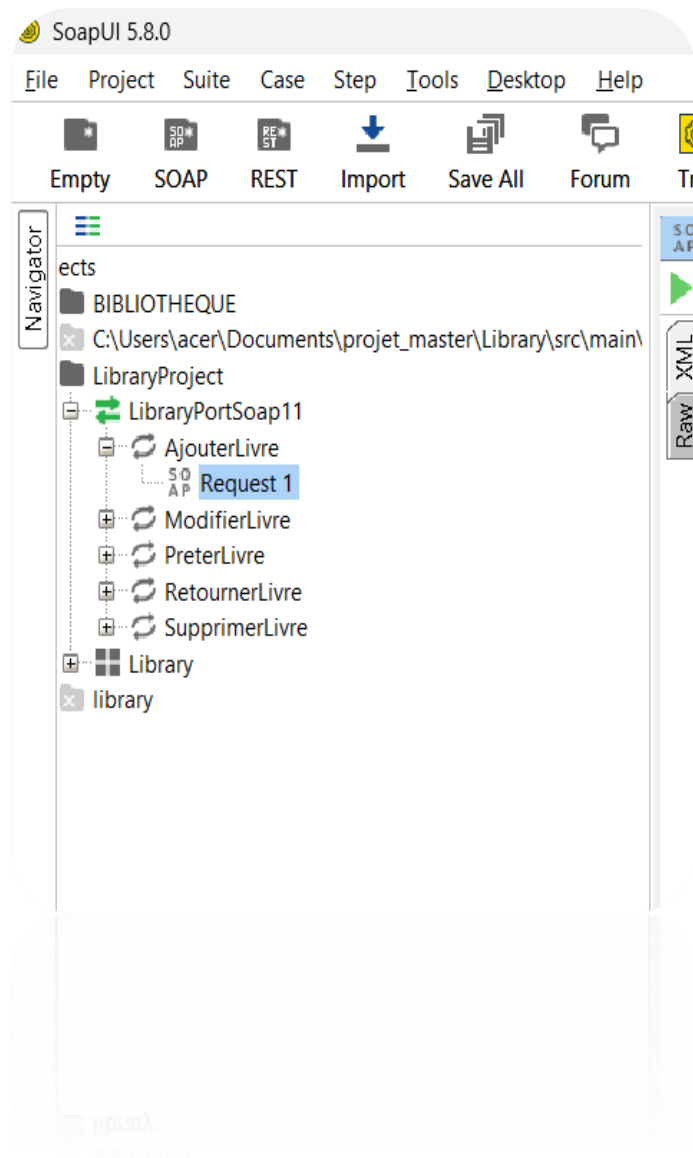


Fig 9 : Service SOAP

6. Tests et démonstration

- SoapUI : importer le fichier WSDL disponible à <http://localhost:8081/ws/livres.wsdl>
- Navigateur : tester les URLs REST et accéder aux consoles/admins

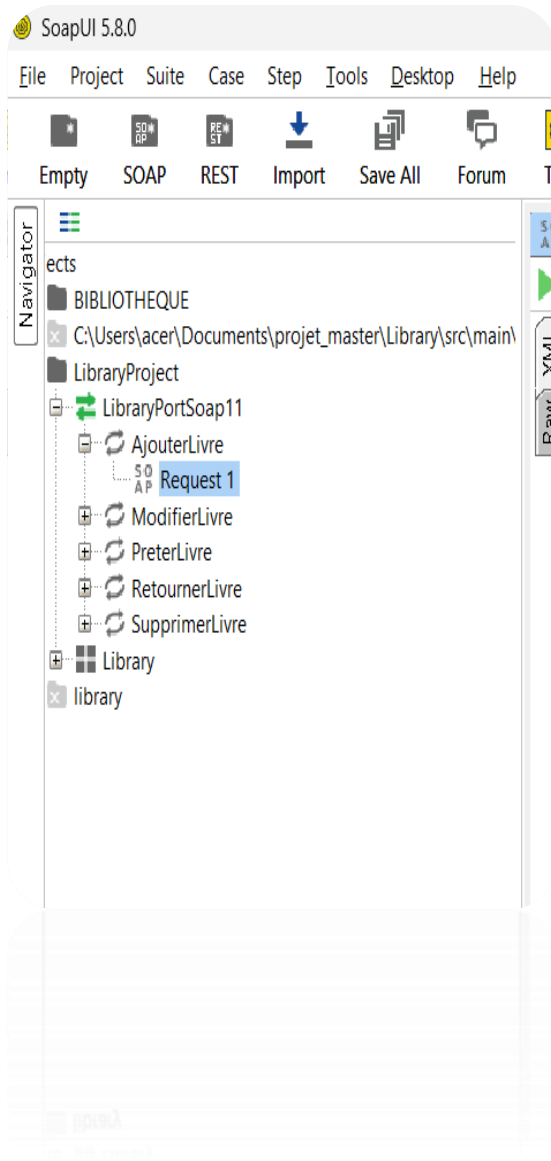


Fig 10: - SoapUI

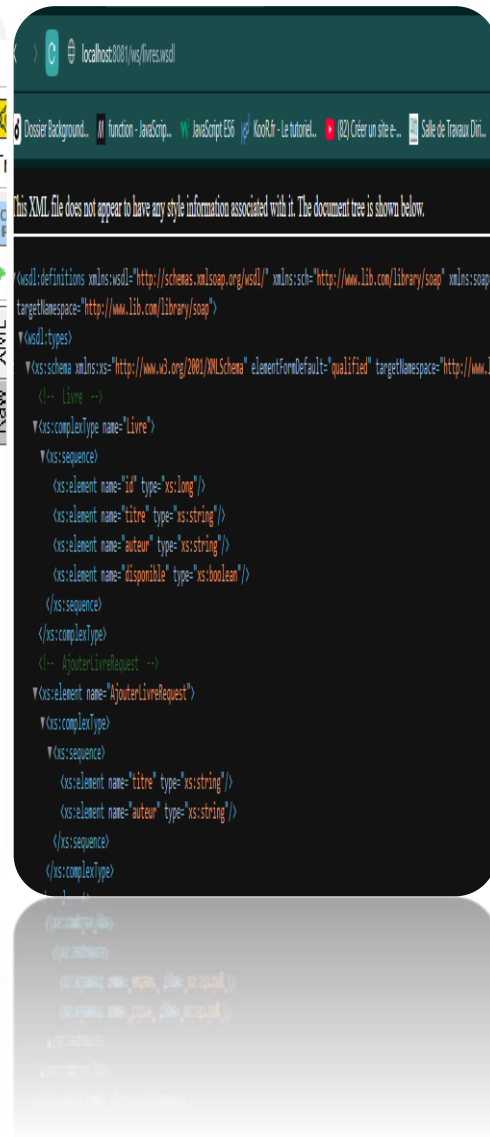


Fig 11: Navigateur

Paramètres H2 Console :

- URL : jdbc:h2:mem:librarydb
- Username : sa
- Password : (laisser vide)
- Driver : org.h2.Driver

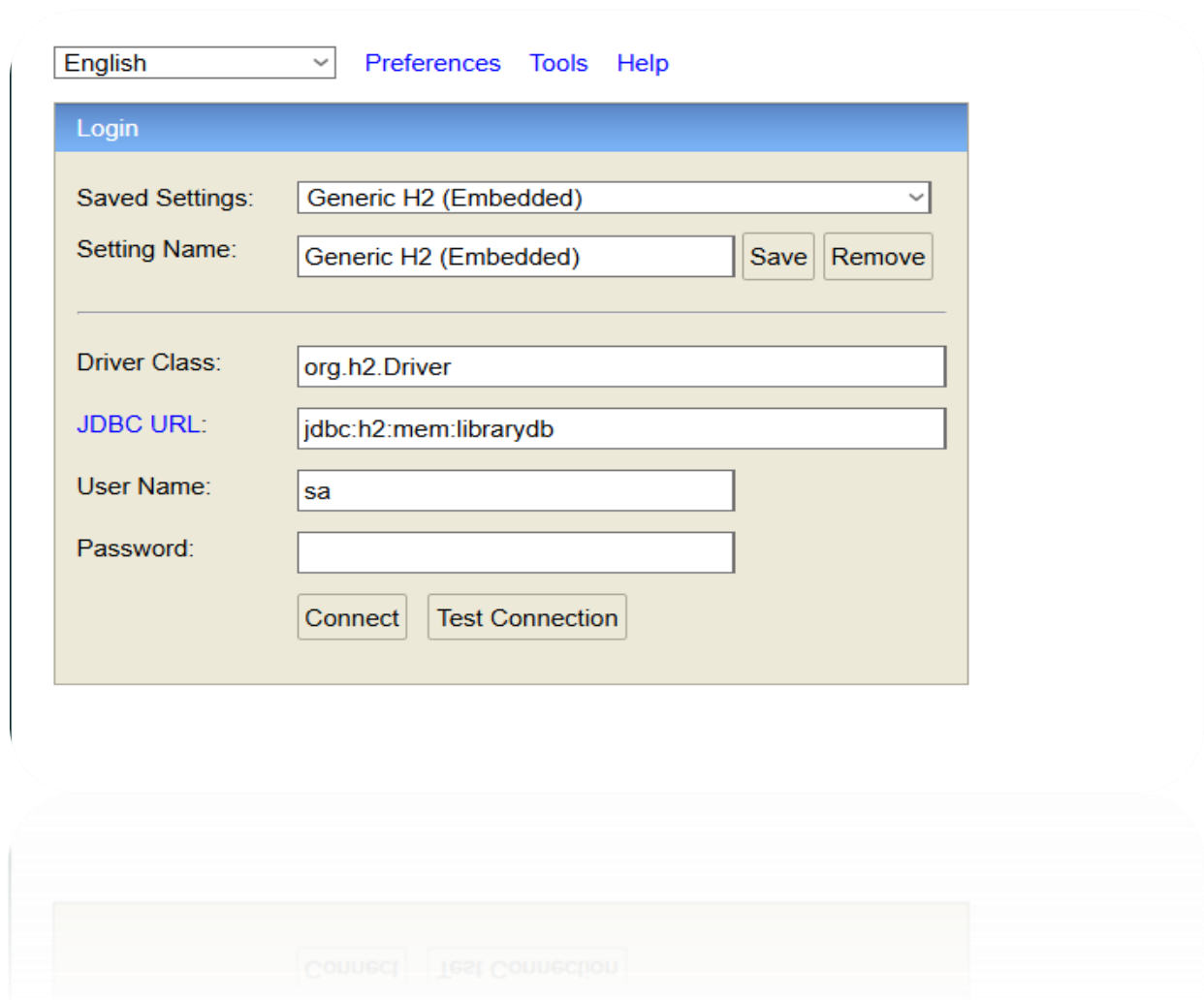


Fig 12 : Paramètres H2 Console

7. Problèmes rencontrés et résolutions

Nous avons rencontré plusieurs défis, notamment :

- Gestion des erreurs
- Validation des données
- Architecture robuste et évolutive

Pour surmonter ces difficultés, nous avons mis en place une bonne documentation et des tests rigoureux afin de réduire les risques de bugs et d'incohérences.

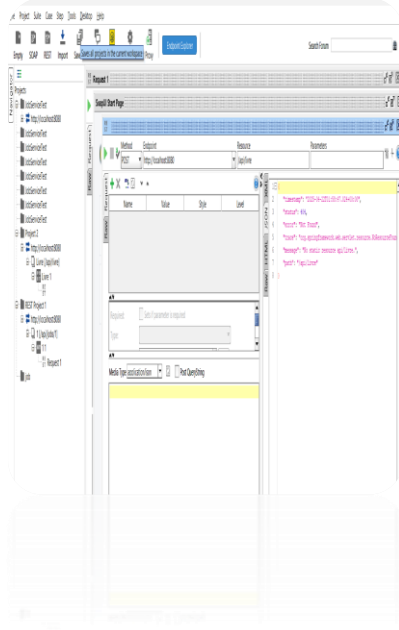


Fig 13: Validation des données

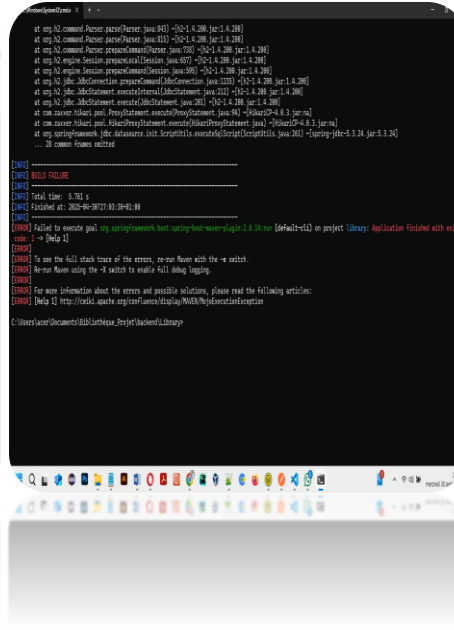


Fig 14: Les risques de bugs

8. Technologies utilisées

Frontend, Angular (TypeScript)

Langages : TypeScript + HTML + CSS

9. Architecture frontend-backend

[Angular Frontend (localhost:4200)]

1

| (HTTP requests)

V

[Spring Boot Backend (localhost:8081)]

1

V

[Base de données (H2, .)]

10. Améliorations futures

- Sécurisation des endpoints
- Ajout d'une interface web (frontend)

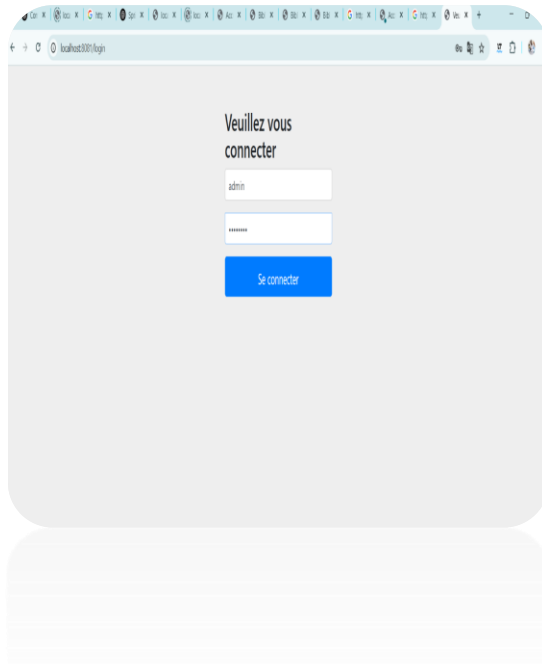


Fig 15: Sécurisation des endpoints



Fig 16 : Ajout d'une interface web (frontend)

11. Conclusion

Ce que nous avons appris :

- Conception et mise en œuvre d'API REST et SOAP
- Intégration de bases de données en mémoire
- Utilisation d'outils comme Postman et SoapUI

12. Annexes

Type	Lien / Description
Console H2	http://localhost:8081/h2-console
REST - Liste des livres	GET → http://localhost:8081/api/livres
REST - Ajouter un livre	POST → http://localhost:8081/api/livres
SOAP - WSDL	http://localhost:8081/ws/livres.wsdl

SECURISATION | <http://localhost:8081/login>

PAGE WEB | <http://localhost:8081/>