Université Sultan Moulay Slimane
Ecole Nationale des Sciences
Appliquées Khouribga

Rapport du TP : Développement d'un Service SOAP

Réaliser par:

BENAOUIJA Mohammed Yassine,
BENERRAGUIG Chafiq,
AIT ITTO Elmahdi

Encadrer par:

Pr. KAROUM Bouchra

1. Introduction

Dans ce projet, nous explorons le développement et le déploiement d'un service **SOAP** (Simple Object Access Protocol) conçu pour gérer des informations de personnes dans une base de données MySQL. SOAP est un protocole utilisé pour échanger des informations structurées dans le cadre de services web. Indépendant des plateformes, ce protocole permet une interopérabilité et est donc idéal pour les applications orientées services.

2. Objectifs

- Créer un service SOAP pour effectuer des opérations CRUD sur une base de données, en
- Gérer les erreurs avec SOAP Faults.
- Développer et configurer le service SOAP.
- Déployer le service sur un serveur.
- Tester le service via un outil comme SoapUI.
- Bonus : Sécuriser le service avec une authentification de base ou WS-Security

3. SOAP : Qu'est-ce que c'est ?

SOAP (Simple Object Access Protocol) est un protocole spécifié pour l'échange d'informations structurées dans les services web. Contrairement à REST, SOAP est un protocole fortement standardisé qui impose des règles pour structurer les messages. Il est adapté aux applications où une communication fiable et sécurisée est essentielle.

Quand et Comment Utiliser SOAP ?

SOAP est particulièrement utile dans les cas suivants :

- Applications d'entreprise nécessitant une sécurité élevée, une fiabilité et une intégrité transactionnelle.
- Interopérabilité inter-plateforme : SOAP peut être utilisé avec différents langages de programmation et protocoles.
- Échange de données complexe : lorsque plusieurs services sont impliqués et que des formats de messages stricts sont nécessaires.

Bénéfices de SOAP

1. **Indépendance de la plateforme** : Les services SOAP peuvent être implémentés sur n'importe quelle plateforme et être accessibles par des clients de différents langages.
2. **Gestion intégrée des erreurs :** Les SOAP Faults fournissent des messages d'erreur structurés, facilitant la gestion des erreurs et l'information des clients sur les problèmes spécifiques.

3. **Extensibilité :** SOAP peut être étendu pour supporter de nouvelles normes, lui permettant d'évoluer au fil du temps avec un impact minimal sur les implémentations existantes.

Limites de SOAP

1. **Complexité :** SOAP peut être plus complexe à configurer et à maintenir que REST, notamment pour des opérations CRUD simples.
2. **Surcharge de performance** : En raison de sa structure de message basée sur XML et des en-têtes additionnels, SOAP peut consommer plus de bande passante et de temps de traitement.
3. **Support limité par les navigateurs :** Contrairement à REST, qui peut être facilement accessible via un navigateur, SOAP nécessite des clients ou frameworks spécifiques.
4. **Protocole de message strict :** Les normes rigides de SOAP peuvent limiter la flexibilité, rendant plus difficile l'implémentation de messageries non standard.

4. Étapes Suivies dans le TP

1. **Mise en Place de l'Environnement :** Installation des outils nécessaires comme JDK, Tomcat, et SoapUI/Postman. Configuration d'un nouveau projet JEE et création de la table Personne dans la base de données.
2. **Création de l'Interface du Service :** Définition de l'interface PersonneService avec des méthodes pour les opérations CRUD.
3. **Implémentation du Service :** Implémentation de l'interface dans PersonneServiceImpl en intégrant la gestion des erreurs avec des SOAP Faults.
4. **Publication et Déploiement du Service :** Déploiement du service, vérification de son accessibilité et validation de la génération du WSDL.
5. **Test du Service :** Utilisation de SoapUI/Postman pour tester toutes les opérations et validation des SOAP Faults en cas d'erreurs.
6. **Sécurisation Optionnelle du Service :** Configuration d'HTTPS et implémentation de WS-Security pour une protection accrue des données

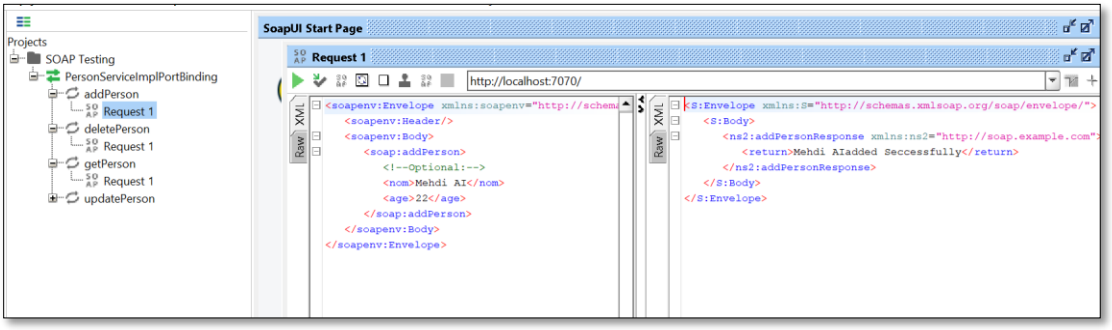
5. Le Code en GitHub :

Github link.

6. Captures d'Écran

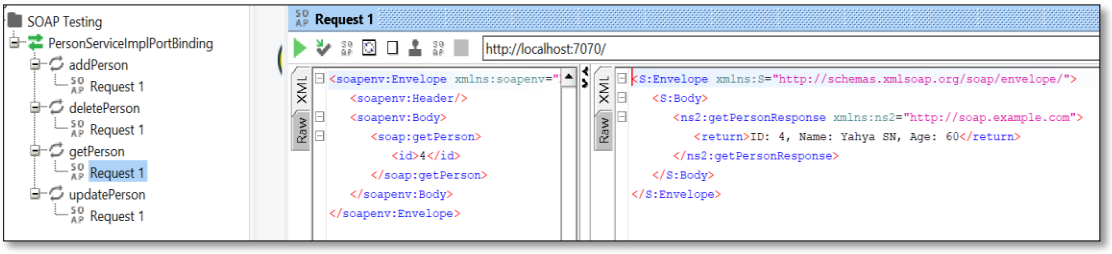
Ces captures d'écran illustrent le test des @WebMethod en utilisant l'outel de test **SOAPUI**

1. **addPerson**

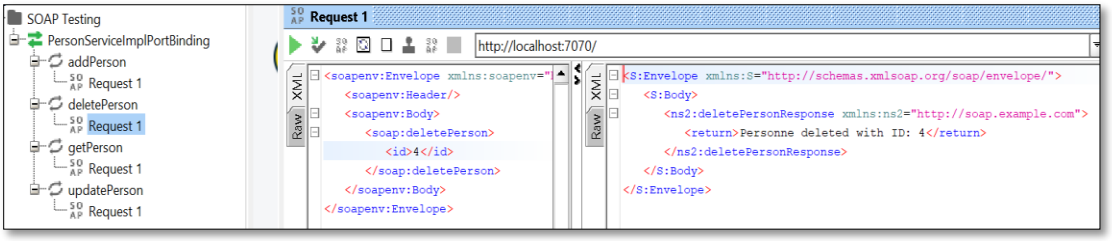


2. **updatePerson**

3. **getPerson**



4. **deletePerson**



---

7. Conclusion

À travers ce TP, nous avons pu développer un service web SOAP entièrement fonctionnel. En mettant en œuvre les opérations CRUD, en gérant les erreurs avec les SOAP Faults, nous avons acquis une compréhension approfondie des capacités et des limites de SOAP dans les architectures orientées services.