

## COMPTE RENDU TP0

Dans ce projet, on a développé un type de service permettant à des programmes de communiquer entre eux : service Web SOAP, testé à l'aide de SOAPUI.

Tout d'abord, un service web permet à un programme de pouvoir appeler à distance un autre, et comment un autre logiciel peut envoyer des requêtes et recevoir des réponses.

Le service qu'on a mis en place fait 3 choses :

- conversion d'un nombre
- calcul d'une somme
- renvoyer les informations les informations d'un étudiant

Ici, SOAPUI joue le rôle du client qui appelle le service, en lui envoyant des requêtes et en affichant les réponses.

On a aussi un fichier WSDL qui est un document qui décrit toutes les opérations disponibles, leurs paramètres et leurs réponses.

- Classe MonServiceWeb

Cette classe contient toutes les opérations que le service web va proposer :

L'annotation `@WebService` indique à Java que cette classe est un service web SOAP et qu'il faudra générer un WSDL et rendre ses méthodes disponibles à distance.

- Méthode conversion

Le but de cette méthode est de prendre un double et de la multiplier par 0.9.

L'annotation `@WebMethod(operationName = "convertir")` indique que cette méthode doit être visible dans le service Web sous le nom de convertir (dans le WSDL et SOAPUI)

Dans SOAPUI, lorsqu'on appellera l'opération convertir avec un nombre quelconque, on recevra le ce nombre\*0.9 en réponse.

- Méthode somme

Cette méthode additionne deux nombres.

`@WebParam(name = "param1")` indique le nom du premier paramètre qui apparaîtra dans le WSDL/SOAP. Cela permet de mieux documenter le service.

- Méthode getEtudiant

Cette méthode permet de renvoyer des informations concernant un étudiant en fonction de son identifiant.

- Classe Etudiant

Cette classe permet de décrire un étudiant avec son identifiant, son nom et sa moyenne.

L'annotation `@XmlRootElement` permet de dire à java que cette classe peut être convertie en XML et envoyée dans une réponse de service web.

L'implémentation de `Serializable` signifie que l'objet Etudiant peut être sérialisé, c'est-à-dire "mis à plat" pour être envoyé ou stocké.

Nous avons 2 types de constructeurs :

- le constructeur vide : est obligatoire pour que le système puisse recréer l'objet automatiquement quand il reçoit du XML
- le constructeur avec paramètres : permet de créer l'étudiant avec les informations nécessaires.

- Le main

Le main démarre le serveur et publie le service web. On a d'abord l'url, qui est l'adresse où le service sera disponible.

Le EndPoint.publish() publie le service web MonServiceWeb à l'adresse indiquée.

Le WSDL est généré automatiquement à partir :

- @WebService
- @WebMethod et @Webparam
- @XmlRootElement

Le WSDL décrit :

- les opérations disponibles : convertir, somme getEtudiant
- les types de messages d'entrée et de sortie
- les types de données utilisés (double, Etudiant)

SOAPUI utilise ce WSDL pour :

- Générer automatiquement des modèles de requêtes
- Envoyer ces requêtes au service
- Afficher les réponses SOAP (en XML)