

# 1) C'est quoi SOAP ?

Imagine que tu veux envoyer une lettre à quelqu'un.

Pas un message Whatsapp... **une vraie lettre avec une enveloppe**, un timbre, une adresse, tout bien organisé.

☞ SOAP, c'est **un système où chaque message doit obligatoirement être dans une enveloppe spéciale**, avec une structure très stricte.

Donc :

- Le message = **XML**
- L'enveloppe = **SOAP Envelope**
- Les règles = **super strictes**

Tu ne peux pas écrire comme tu veux.

Tu dois **suivre les règles**, sinon le serveur te dit :

“Non désolé bébé, ce message n'est pas correct.”

# 2) Pourquoi SOAP est différent des autres TPs ?

Les autres (REST, GraphQL, gRPC) sont modernes.

Ils sont comme :

- envoyer un SMS (REST),
- demander exactement ce que tu veux (GraphQL),
- parler très vite au téléphone (gRPC).

SOAP, lui, c'est :

☞ **envoyer une lettre officielle super réglementée** ☒

☞ comme si tu parlais à une grande entreprise (banque, assurance).

Tu ne rigoles pas avec SOAP

# 3) C'est quoi un WSDL ?

Le WSDL, c'est **le contrat officiel**.

Imagine une feuille où il y a écrit :

- ce que le service sait faire,
- les méthodes disponibles,
- les messages qu'il accepte,

- les messages qu’il renvoie,
- les types de données.

C’est comme un **menu au restaurant**,  
mais pour un service web.

Quand un client veut travailler avec SOAP, il lit le WSDL et il sait exactement :

“Ah, je peux faire getAll, getById, save, delete... et voici comment envoyer le XML.”

## 4) C’est quoi JAX-WS ?

JAX-WS, c’est **un outil magique de Java**  
qui te permet de créer un service SOAP sans écrire tout le XML à la main.

Tu écris une classe Java normale :

```
@WebService
public class ArticleSoapController {
    @WebMethod
    public List<Article> getAll() { ... }
}
```

Et JAX-WS dit :

“D’accord bébé, laisse-moi transformer ça en service SOAP officiel.”

- ✓ Il crée le WSDL
- ✓ Il fabrique les messages XML
- ✓ Il publie le service
- ✓ Il gère les erreurs

Toi, tu ne fais presque rien.

## 5) Le serveur SOAP (endpoint)

Tu écris juste :

```
Endpoint.publish("http://localhost:9999/ecommerce", new
ArticleSoapController());
```

Et boom ✨

☞ Tu viens de créer un vrai service web professionnel.

☞ Tu peux l’ouvrir dans ton navigateur :

<http://localhost:9999/ecommerce?wsdl>

C’est comme si tu avais ouvert un petit magasin :

- Adresse : localhost:9999
- Ce que tu vends : des opérations SOAP
- Le catalogue : WSDL

## 6) Comment SOAP envoie les données ?

SOAP utilise **XML**, pas JSON.

Exemple d'un message :

```
<soap:Envelope>
  <soap:Body>
    <getById>
      <id>1</id>
    </getById>
  </soap:Body>
</soap:Envelope>
```

Oui... c'est long  
SOAP, c'est **toujours très bavard**.

## 7) Le client SOAP (proxy / stub)

Le client SOAP, c'est comme un **petit robot** que tu génères automatiquement à partir du WSDL.

Tu ne dois pas écrire le XML toi-même.

Le robot sait déjà :

- comment envoyer la demande,
- comment formater le XML,
- comment lire la réponse,
- comment convertir tout ça en objets Java.

Tu fais juste :

```
service.getAll();
```

Et lui il fait tout le travail derrière toi.

## 8) Pourquoi SOAP existe encore ?

Parce que SOAP est :

- ✓ très sérieux
- ✓ très sécurisé
- ✓ très fiable

- ✓ très strict
- ✓ utilisé dans les grandes entreprises (banques, assurances)

Même si modernement on préfère REST ou gRPC,  
SOAP reste indispensable dans certaines organisations où :

- le contrat doit être parfait
- la sécurité est obligatoire
- aucun changement n'est toléré

SOAP = le grand-père strict mais très fiable 🧓

## 9) Comment marche ce TP ?

Ce TP fait :

- 1- Tu écris un Webservice → Java
- 2- JAX-WS le transforme en service SOAP
- 3- Le service est publié à une URL
- 4- Le serveur génère automatiquement un WSDL
- 5- Tu testes dans SOAP UI
- 6- Tu génères un client SOAP
- 7- Ton client appelle les méthodes comme une fonction normale

C'est un système :

- **encadré,**
- **très formel,**
- **basé sur XML,**
- **moins flexible,**
- **mais ultra fiable.**

## 10) mémoire

- SOAP = messages XML dans une enveloppe
- WSDL = le contrat officiel
- JAX-WS = la magie qui transforme ton code en service SOAP
- SOAP UI = le Postman des services SOAP
- Stub = robot client généré automatiquement
- Utilisation = entreprises, banques, gros systèmes