

# Rendu TP2 :service web SOAP

**Meryam RHADI**  
**Lamyae KHAIROUN**

**5 décembre 2021**

**Master 1**  
**génie logiciel**  
UE Architectures logicielles Distribuées

**Responsable**  
Abdelhak-Djamel Seriai  
Bachar Rima

## Sommaire

Titre	1
Sommaire	2
1 Objectif :	3
2 Conception	3
2.1 Diagramme pour version non distribuée . . . . .	3
2.2 Diagramme pour version distribuée . . . . .	4
3 Version non distribué :	4
4 Version distribuée	5
4.1 Explication des deux projets crée serveur et client : . . . . .	5
4.1.1 Projet serveur : . . . . .	5
4.1.2 Projet client : . . . . .	6
4.2 Explication de fonctionnement . . . . .	7
5 Read Me	8

## 1 Objectif :

L'objectif de ce TP est de développer en Java une application de réservation d'hôtels en ligne, on répondant au client qui saisie les informations suivantes :Une ville de séjour, une date d'arrivée, une date de départ, un intervalle de prix souhaité, l'application lui retourne une liste d'hôtels qui répondent à ses critères.

L'utilisateur choisira un hôtel de la liste et pour effectuer la réservation l'application lui demande : le nom et prénom de la personne principale à héberger, les informations de la carte de crédit de paiement. Et on l'appuie sur la technologie **SOAP** pour développer cette application.

## 2 Conception

### 2.1 Diagramme pour version non distribuée

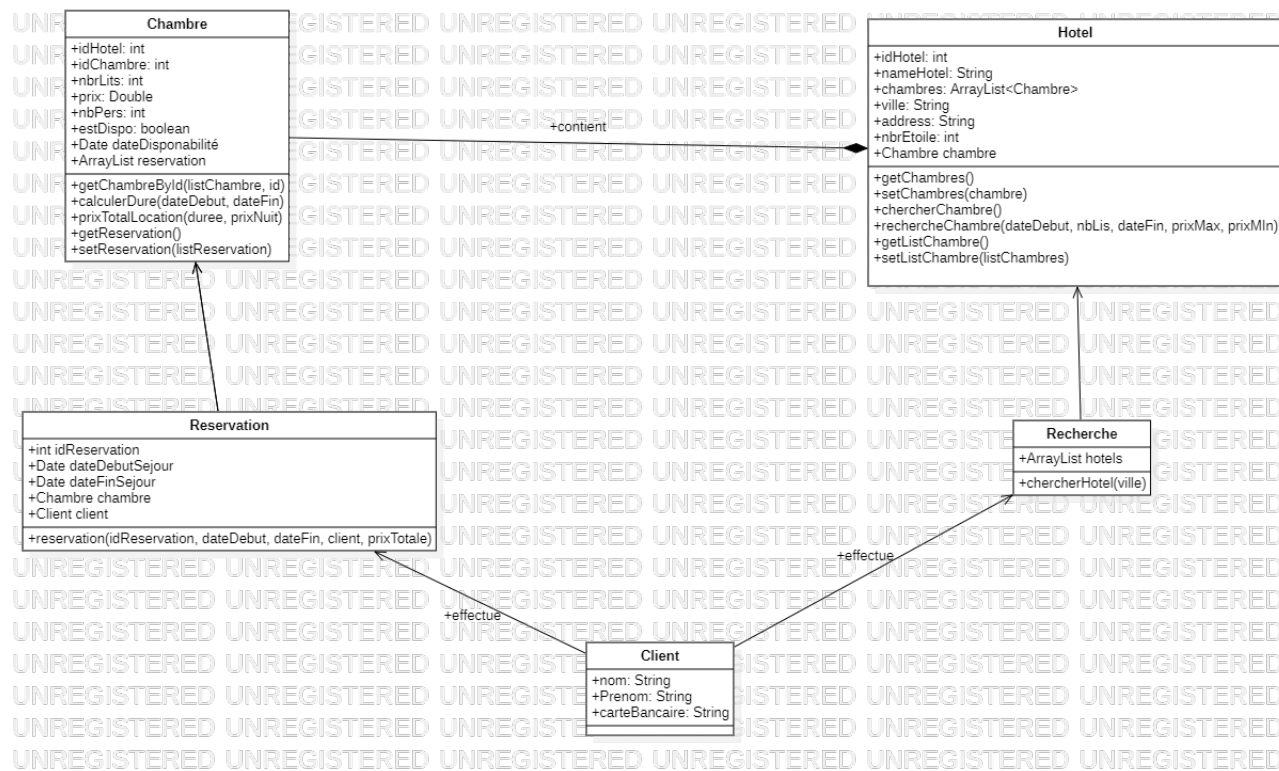


Figure 1. Diagramme de la version non distribuée

Dans cette conception, on a un hôtel qui contient plusieurs chambre, et un client effectue une recherche sur un hôtel pour accéder au chambre, et réserver celle qui correspond à ces critères.

## 2.2 Diagramme pour version distribuée

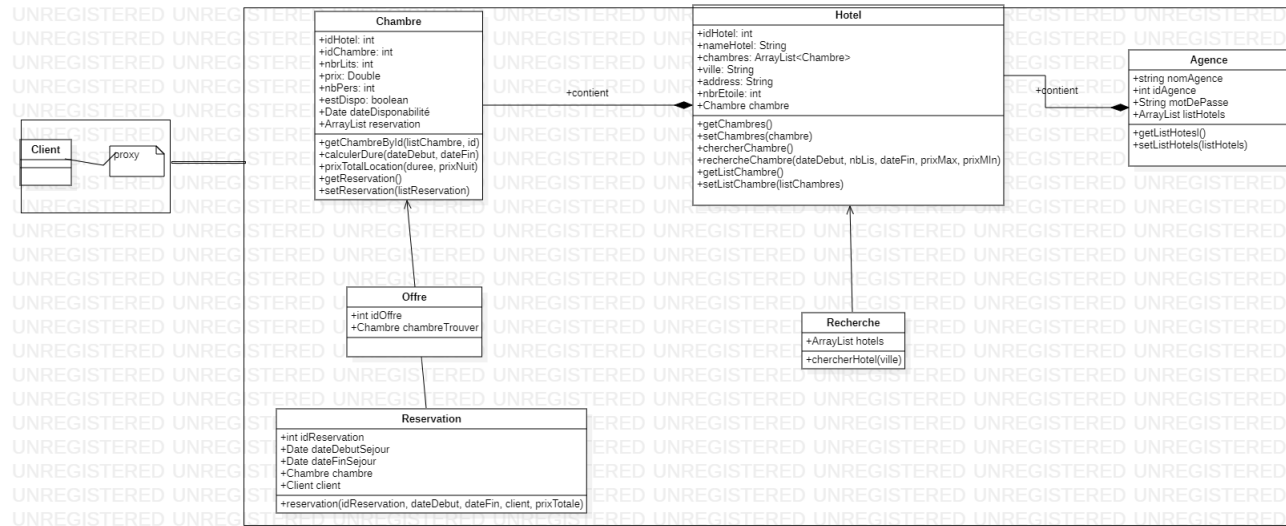


Figure 2. Diagramme de la version distribuée

Dans cette conception on a décidé d'avoir une tel agence qui contient une liste d'hôtels, et cette dernier contient une liste des chambres. La recherche s'effectue sur la liste des chambres d'un hôtel donné, et les offres contient les chambres correspondantes aux critères d'un client. La réservation s'effectue sur une chambre. Tout ce traitement s'effectue dans le projet serveur, et dans l'autre coté le client, on utilise le proxy pour communiquer avec le serveur.

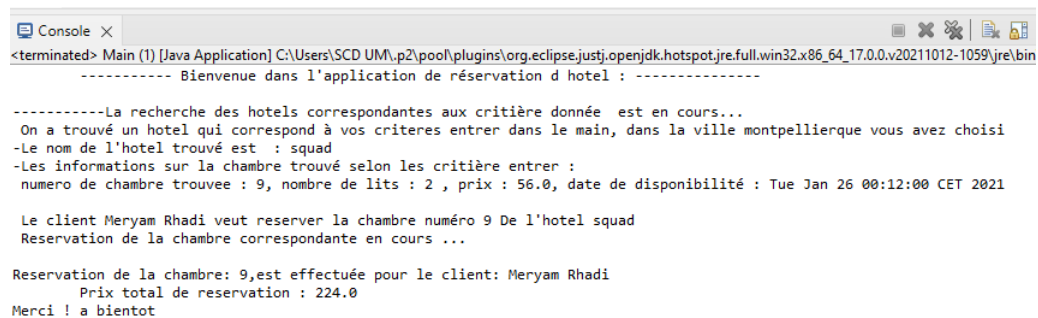
## 3 Version non distribué :

Cette version est simple, il propose les fonctions essentiels pour répondre au requête de l'utilisateur, il pose la logique métier de l'application, pour développer cette version on a implémenté les classes suivante :

- Hôtel : cette classe décrit la structure d'un hôtel en définissant, ( id de l'hôtel, nom, nombre des étoiles, et une liste des chambres), et on a ajouté des méthodes pour le traitements des chambres dans l'hôtel comme la recherche d'une listes des chambres correspondantes à certaines critères données.
- Chambre : cette classe décrit la structure d'une chambre ( son id, id de l'hôtel dont elle appartient, nombre de lits, nombre de personnes à héberger, son prix par nuit, un boolean qui dit si la chambre est disponible ou non, date de disponibilité de la chambre, et une liste

de réservations pour stocker tout les réservations de la chambre en question).

- Client : cette classe définit la structure d'un client(prénom, nom, données de la carte bancaire).
- Recherche : cette classe dispose d'une méthode qui permet de rechercher et retourner une liste des hôtels qui possèdent des chambres correspondantes aux critères proposées par le client.
- Réservation : cette classe contient les champs nécessaire pour une réservation donnée, qui lie un client avec la chambre qu'il a choisie.



```
Console X
<terminated> Main (1) [Java Application] C:\Users\SCD UM\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.0.v20211012-1059\jre\bin
----- Bienvenue dans l'application de réservation d hotel : -----

-----La recherche des hotels correspondantes aux critiere donnée est en cours...
On a trouvé un hotel qui correspond à vos criteres entrer dans le main, dans la ville montpellierque vous avez choisi
-Le nom de l'hotel trouvé est : squad
-Les informations sur la chambre trouvé selon les critiere entrer :
numero de chambre trouvee : 9, nombre de lits : 2 , prix : 56.0, date de disponibilité : Tue Jan 26 00:12:00 CET 2021

Le client Meryam Rhadi veut reserver la chambre numéro 9 De l'hotel squad
Reservation de la chambre correspondante en cours ...

Reservation de la chambre: 9,est effectuée pour le client: Meryam Rhadi
Prix total de reservation : 224.0
Merci ! a bientot
```

**Figure 3.** Affichage de résultat d'exécution de la version non distribuée

Cette capture correspond à l'affichage d'un offre qui est le résultat d'une recherche qu'un client a effectué selon ces critères.

Puis le client réserve la chambre trouvé, dont l'id est égale à 9, nombre de lits est 2, et son prix par nuit est 56.00.

Puis on a effectué cette réservation pour ce client, on lui confirme la réservation et on lui affiche le prix total de la réservation qui est calculé à partir de la durée de son séjour.

## 4 Version distribuée

### 4.1 Explication des deux projets crée serveur et client :

Cette version est distribuée, donc on vas travailler sur deux projet :

#### 4.1.1 Projet serveur :

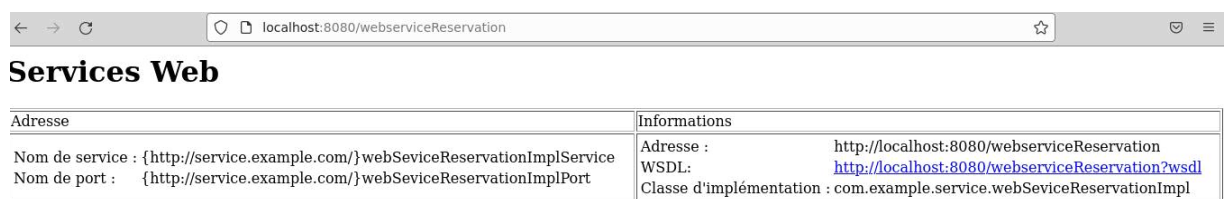
ce projet contient le service web, on a crée deux package, un pour partie Modèle, il contient les classes de basses (Hôtel, Chambre Offre, Agence et Réservation), et autre package (service), qui gère les services web, on a crée une interface pour le premier service web en utilisant la notation **@webservice**, et la notation **@webmethode** pour les signatures

des méthodes qu'on souhaite les rendre distribuée.  
Puis on a implémenté cette interface et ses fonctions distribuées.  
Pour publier notre service web, on a implémenté une classe Publisher qui publie notre service web en utilisant **EndPoint**.  
On a fait la même chose pour le deuxième service web en suivant cette procédure. En fait on a créé deux services web et on a publié leurs wsdl pour qu'il soit consommable par les clients.

## Services Web

Adresse	Informations
Nom de service : {http://service.example.com/}WebserviceImplServiceService Nom de port : {http://service.example.com/}WebserviceImplServicePort	Adresse : <a href="http://localhost:8080/servicewebSOAPHotel">http://localhost:8080/servicewebSOAPHotel</a> WSDL: <a href="http://localhost:8080/servicewebSOAPHotel?wsdl">http://localhost:8080/servicewebSOAPHotel?wsdl</a> Classe d'implémentation : com.example.service.WebserviceImplService

Figure 4. Publication du service web recherche



Adresse	Informations
Nom de service : {http://service.example.com/}webServiceReservationImplService Nom de port : {http://service.example.com/}webServiceReservationImplPort	Adresse : <a href="http://localhost:8080/webserviceReservation">http://localhost:8080/webserviceReservation</a> WSDL: <a href="http://localhost:8080/webserviceReservation?wsdl">http://localhost:8080/webserviceReservation?wsdl</a> Classe d'implémentation : com.example.service.webServiceReservationImpl

Figure 5. Publication du service web réservation

### 4.1.2 Projet client :

on a crée un autre projet coté client, dont les clients puissent consom-  
mer le service web, donc on a utilisé la commande : **wsimport -keep -p**  
**"web.service.client" "http://localhost:8080/servicewebSOAPHotel"** , où  
on donne l'adresse de notre première service web, ce qui nous permet  
d'utiliser les fonctions distribué, après la génération de ces classes on  
copie les classes avec extension **.java** dans un package coté client, et  
on crée une classe Main pour appeler les fonctions distribuée. Dans le  
main on avait besoin du proxy pour pouvoir invoquer ces méthodes.  
On répète la même procédure pour accéder au deuxième service web  
de réservation.

## 4.2 Explication de fonctionnement

Dans la partie client dans la classe main, on a appelé nos services web publiés par le serveur.

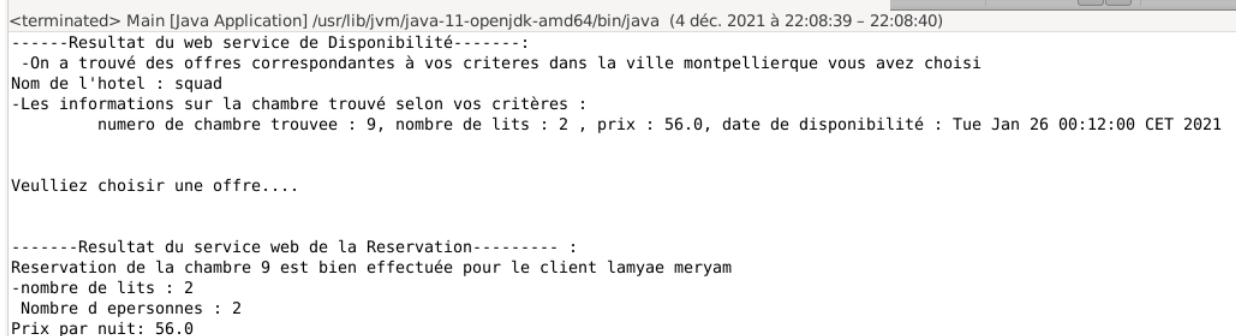
On a considéré qu'on est un client et on va consommer les services web, donc on a entré manuellement les critères qui concernent la recherche de l'hôtel qu'on veut dans le main. Pour ce faire on a appelé la fonction chercherOffre de notre premier service web de Recherche.

Donc on a spécifié l'id et le mot de passe de l'agence pour pouvoir chercher l'agence correspondante et accéder à ses données(une listes des hôtels), la ville de séjours, ainsi que les critères qui concerne la chambre(le prix, le nombre de lits souhaitées, le nombre de personnes à héberger, l'intervalle de prix et finalement la date souhaité pour le début et la fin de séjour), et on ajoute ces dernières comme paramètres de la fonction distribuée.

Le serveur s'occupe de chercher des offres dans les données de l'agence correspondante, une fois il trouve un hôtel dans la ville donnée, il filtre ces hôtels trouvés selon les critères de la chambre souhaité, et finalement il retourne une liste des chambres qui correspond aux clients comme des offres.

Au niveau du retour des méthodes de ce service web, on a choisi de convertir les listes des offres trouvé en type String pour faciliter l'échange. Après avoir afficher au client la liste des offres correspondantes à ses critères, on demande au client de choisir un offre pour effectuer la réservation de la chambre correspondante.

Si la chambre est déjà réservé par un autre client dans la date en question, on lui retourne que la chambre est occupé, sinon on effectue la réservation en l'ajoutant dans la liste des réservation de cette chambre, et on annonce au client que la réservation est bien enregistré.



```
<terminated> Main [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (4 déc. 2021 à 22:08:39 - 22:08:40)
-----Resultat du web service de Disponibilité-----:
-On a trouvé des offres correspondantes à vos criteres dans la ville montpellierque vous avez choisi
Nom de l'hotel : squad
-Les informations sur la chambre trouvé selon vos critères :
    numero de chambre trouvee : 9, nombre de lits : 2 , prix : 56.0, date de disponibilité : Tue Jan 26 00:12:00 CET 2021

Veulliez choisir une offre....

-----Resultat du service web de la Reservation----- :
Reservation de la chambre 9 est bien effectuée pour le client lamyae meryam
-nombre de lits : 2
Nombre d epersonnes : 2
Prix par nuit: 56.0
```

Figure 6. Affichage coté client

Dans cette première capture on a donné la ville de Montpellier, comme critère de l'hôtel , et d'autres critère pour la chambre, notre fonction distribuée nous a retourner une chambre avec ces critères, qui est la chambre numéro 9 de l'hôtel squad à Montpellier, et après on a effectué la réservation de cette chambre pour le client qui a choisi de la réserver.

```
<terminated> Main [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (4 déc. 2021 à 22:09:51 - 22:09:52)
-----Resultat du web service de Disponibilité-----:
-On a trouvé des offres correspondantes à vos criteres dans la ville montpellierque vous avez choisi
Nom de l'hotel : squad
-Les informations sur la chambre trouvé selon vos critères :
    numero de chambre trouvee : 9, nombre de lits : 2 , prix : 56.0, date de disponibilité : Tue Jan 26 00:12:00 CET 2021

Veulliez choisir une offre....

-----Resultat du service web de la Reservation----- :
Chambre non disponible
```

**Figure 7.** Affichage coté client

Quand on essaye de réserver une autre fois la même chambre notre serveur indique que cette dernière n'est pas disponible.

## 5 Read Me

Pour pouvoir exécuter le projet, le serveur doit être lancé, comme ça on peut le consommer cote client et invoquer les méthodes par le client.