

Université de Montpellier

Master Informatique parcours ICO *1ère année*

Module HAI704I Architecture logicielles
distribuées

TP3 REST-Version finale

JIANG Lijuan - BOUALAM Karim - POIRET Valentin

Décembre 2022

1 Diagramme des paquetages

Pour représenter les liens existants entre les différents composants du projet, nous nous sommes appuyés sur le diagramme UML représenté ci-dessous :

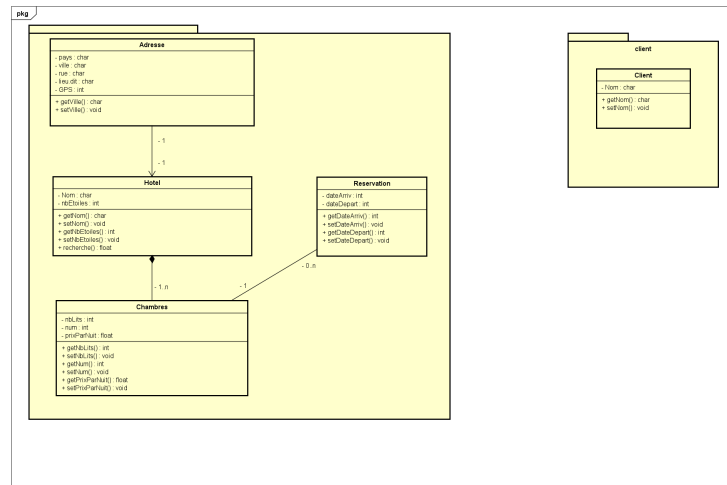


FIGURE 1 – Diagramme des paquetages

2 Conception

2.1 Organisation des Programmes

Pour la réalisation de ce TP, nous avons créé deux projets Spring Boot sur Eclipse que nous avons nommés "hotelREST" et "hotelRESTClient" .

2.2 Projet Web Service

Dans ce projet, nous avons créé cinq packages dans "src/main/java". Le package "model" contient les classes necessaire à la création d'une chambre et la reservation de celles-ci. Deux interfaces "ChambreRepository" et "ReservationRepository" sont créés dans le package "repositories". Dans le package "data" nous avons créé la base de données de l'hôtel, le package "controllers" signifie les web services, qui contient les méthodes "recherche" et "reserver", ces méthodes nous permettent de trouver les chambres libres dans l'hôtel et les réserver. Le package "main" contient la méthode "main" qui nous permet de lancer le serveur.

2.2.1 Lancement du serveur

Pour tester le serveur nous considérons un hotel qui contient une liste de chambre et une methode de recherche qui permet à un client de trouver les chambres libres pour une periode (date arrivée, date de départ) et en fonction du nombre de personnes à logger. Voir photo ci-dessous :

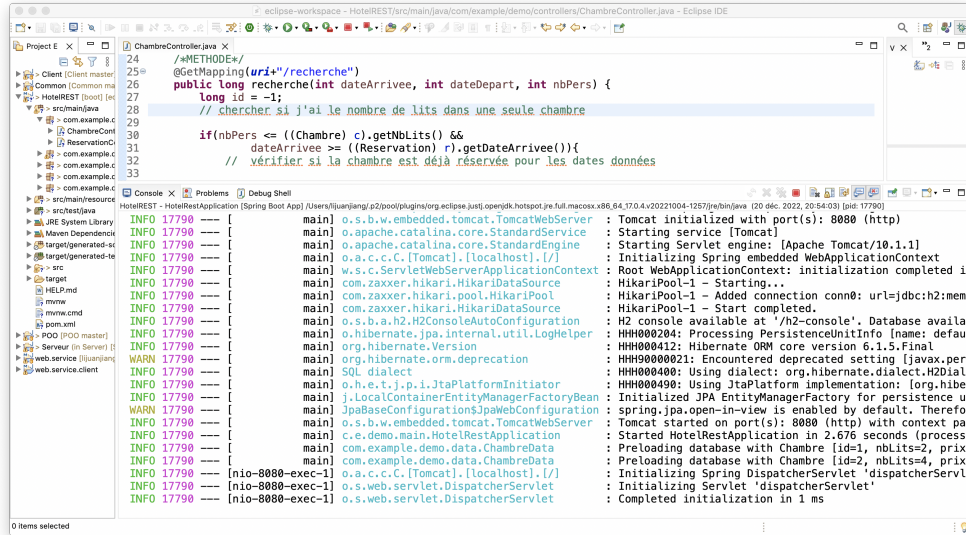


FIGURE 2 – console IDE après lancement du serveur

2.2.2 Base de données

Utilisation de la base de donnée H2 depuis le navigateur web.

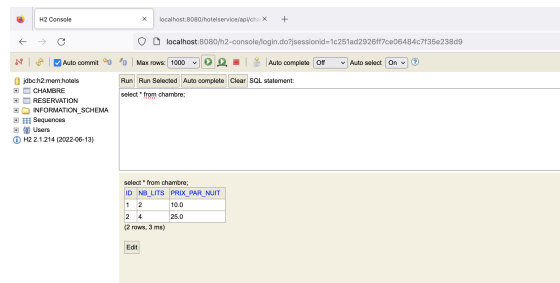


FIGURE 3 – base de données

Utilisation du service web avec affichage de la liste de chambres dans le terminal avec "curl".

```
(base) MacBook-Pro-0:liujuan~ liujuanjiang$ curl -v http://localhost:8888/hotelservice
* Trying 127.0.0.1:8888...
* Connected to localhost (127.0.0.1) port 8888 (#0)
> GET /hotelservice/api/chambres HTTP/1.1
> Host: localhost:8888
> User-Agent: curl/7.78.0
> Accept: */*
<
* Mark bundle as not supporting multiuse
< HTTP/1.1 200
< Content-Type: application/json
< Transfer-Encoding: chunked
< Date: Tue, 20 Dec 2022 19:58:34 GMT
<
* Connection #0 to host localhost left intact
{"id":"1","nbLits":2,"prixParNuit":10.0}, {"id":"2","nbLits":4,"prixParNuit":25.0}](base)
```

FIGURE 4 – utilisation du web service dans le terminal

Utilisation du service web avec affichage de la liste de chambre au format "json pp" (json pretty printer) dans le terminal avec "curl".

```

Connection #0 to host localhost left intact
{"linesize":7822,"id":"D7F9156156140A8B0E08","status":"ok","error":"","not_found","path":"/htmlservice/api/shmehere?jbase=Macbook-Pro-de-l'ujuan-11juanjing?curl=http://localhost:8088/htmlservice1.html"}
Total % Received % Xferd Average Speed Time Time Time Current
                                elapsed remaining bytes/sec complete
0 0 0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0 # Connected to localhost [127.0.0.1 port 8088] (0)
GET /htmlservice/api/shmehere/2 HTTP/1.1
Host: localhost:8088
User-Agent: curl/7.76.0
Accept: */*
# Main handle as not supporting multiuse
# HTTP/1.1 200
Content-type: application/json
Transfer-Encoding: chunked
Date: Tue, 28 Dec 2022 13:57:06 GMT
[14 bytes data]
100 100 0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 12666
# Connection #0 to host localhost left intact
{
  "offset": 2,
  "pageSize": 10,
  "ip": 1
}

```

FIGURE 5 – utilisation du web service dans le terminal

Affichage des information relatives à la chambre numero 1 dans le navigateur.

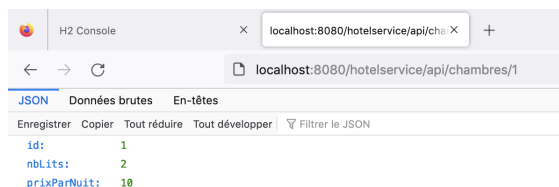


FIGURE 6 – informations sur la chambre N°1

2.3 Projet Web Service Client

Dans ce projet spring boot client nous avons créé quatre packages. Un premier package models qui contient les classes "Chambre" et "Reservation". Un second package appelé client qui servira à créer le proxy. Un troisième package appelé CLI qui permettra aux clients de faire leur recherche de chambre. Et un quatrième package "Main". Ce projet web service client qui nous permettra

de rechercher une chambre à l'aide de la methode de recherche. Voir photo ci-dessous :



```
Console x
HotelRestClient - HotelRestClientApplication [Spring Boot App] [pid: 12128]
2022-12-22T12:58:23.006+01:00 INFO 12128 --- [ main] c.e.d.main.HotelRestClientApplication : Started HotelRestClientAppl
Please provide the URL to the web service to consume:
http://localhost:8080/hotelservice/api
0. Quit.
1. Chambres
2. Chambre ID
3. Rechercher
4. Reserver
```

FIGURE 7 – Affichage CLI du coté Client

2.4 Résultats du projet

Concernant les résultats, nous avons réussi à lancer le serveur, créer la base de données et tester la base de données en Terminal et PostMan, ensuite, nous avons réussi à afficher une liste de chambres et chercher les chambres par Id, finalement, nous avons réussi à lancer le projet coté Client avec CLI. Les difficultés rencontrées dans notre projet consiste aux problèmes du chemin des variables (date arrivée, date départ) dans la méthode "recherche". Nous pourrions encore ajouter la méthode "reserver" dans la classe "ResrvationController".

3 Conclusion

Ce projet nous permet de bien comprendre Web service REST.

ReadMe :

- Ouvrir le projet "HotelREST" dans eclipse et lancer la classe "HotelRestApplication.java"
- Ouvrir le projet "HotelRESTClient" dans eclipse et lancer la classe "HotelRestClientApplication.java"
- Saisir "http ://localhost :8080/hotelservice/api", suivre les indication inscrites en ligne de commande sur le terminal d'eclipse.