esprit Se former autrement HONORIS UNITED UNIVERSITIES	**EXAMEN** **Session Principale**

Module: Architecture Orientée Services (SOA)
Enseignant(s): Equipe Web
Classe : 4éme Année
Documents autorisés : OUI ☐ NON ■ Nombre de pages : 6 + 4(Feuilles de réponse)
Date : 29/10/2024 Heure : 13H00 Durée : 1h30

Exercice1 (7 pts)

1. Que signifie l'intégration ? **(0,75)**
2. A l'aide d'un schéma, expliquer la relation entre le Middleware et l'Architecture Orientée Services (SOA). **(1 pt)**
3. Quel est la différence entre XML et WADL? **(1 pt)**
4. Compléter ce paragraphe par les mots suivants : **(2,5 pts)**

JSON, ressources, méthodes HTTP, léger, XML, SOAP , messages, concrète, abstraite, POST, GET, PUT, DELETE, , XML, Middleware, WADL, Bottom-Up, UDDI , méthodes, URI ,Top-Down ,WSDL , REST , opérations , API, SMTP , SOA , lourd

Les services Web permettent aux systèmes distribués et hétérogènes de communiquer entre eux via un réseau. On distingue deux styles principaux, basé chacun sur un Protocol de communication différent: les services Web étendus (basés sur __(1)_____) et les services Web RESTful (basés sur (2)_____).

Dans les services RESTful, le style d'architecture repose sur l'utilisation de (3) _____ identifiées par des _(4)_____ et des (5) _____ standards.

REST est ___(6)_____ et utilise des formats de données comme __(7)_____ et ___(8)_____. Le document __(9)_____ peut être utilisé pour décrire les services RESTful, bien que l'approche __(10)_____ soit principalement utilisée dans leur développement.

5. Le **JWT (ou JSON Web Token)** est un standard ouvert qui permet la sécurité d'un service web à l'aide d'un jeton (token) signé. **(1,75 pts)**
   - **a.** Expliquer son principe. **(1 pt)**
   - **b.** Enumérer ses composants : (0,75)

Exercice 2 : Etude de cas : Système de gestion des ventes – EasySales (13pts)

Une entreprise souhaite développer une application de gestion des ventes permettant de gérer les **clients**, leurs **commandes**, les **produits** et les **lignes de commande**. Chaque client

peut passer plusieurs commandes, chacune contenant plusieurs produits sous forme de lignes de commande avec détails (quantité, prix….). L'application facilite la gestion des clients, le suivi des commandes et l'inventaire des produits.

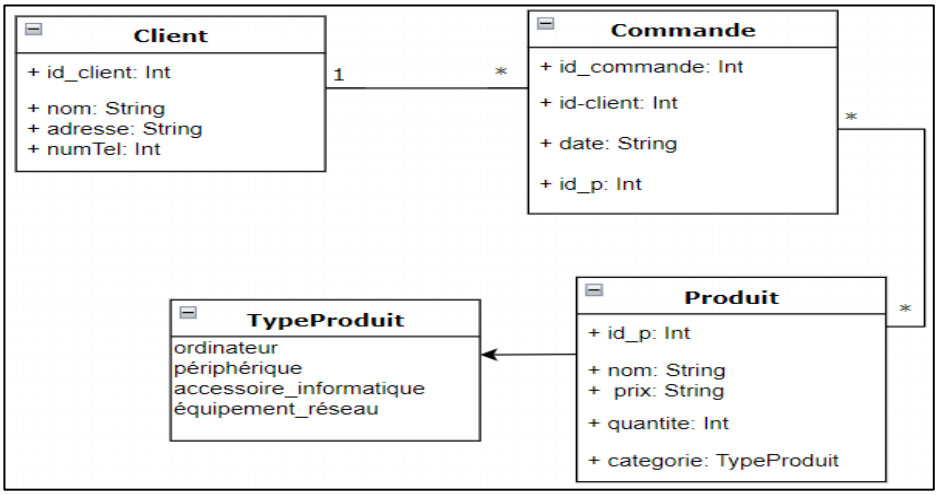

FIGURE 1 : Diagramme de classe

Partie 1 : REST (7pts)

On souhaite développer une application RESTFull qui répond aux besoins de l'entreprise.

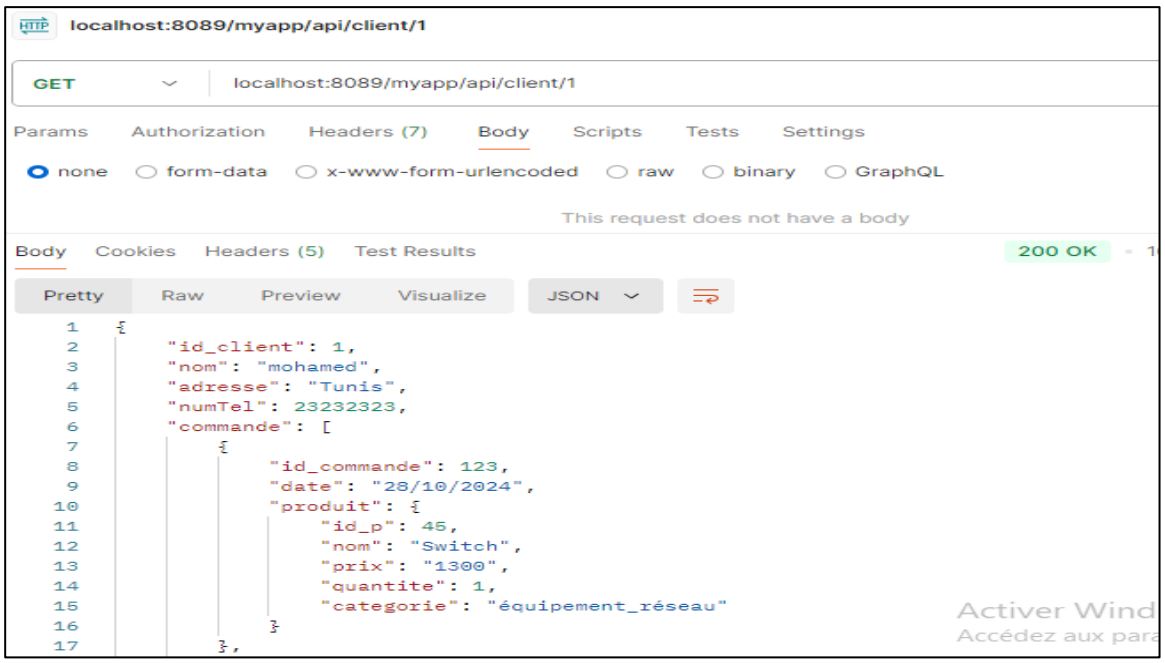

FIGURE 2 : Afficher toutes les commandes du client

1. Soit la requête ci-dessus, permettant de lister toutes les commandes pour un client donné.

a- Terminer la classe « **RestActivator** » ci-dessous. (**0.75pt**)

…….**[1]**(''……..**[2]**'')

```
public class ActivatorRest ……..[3]

}
```

b- Expliquer le rôle de cette classe. (**0.75pt**)

2. Lors des tests des SW (services web) de l'application **EasySales**, une erreur avec le code « **401** » est affichée dans Postman.

   a- Expliquer la cause de cette erreur et proposer une solution pour continuer le test de la méthode **(1,5pt)**

   b- Terminer le code java ci-dessous, de la méthode « **getAllCmdByClient** » relative à la figure 2 **(1.75pt)**

```
……[4]
public class ClientRessources {
……[5]
……[5]
……[6]
public ……[7] getAllCmdByClient(……[8] int id_client) {
        List<Commande> commandesClient = new ArrayList<>();
        for (Commande commande : listeCmds) {
            if (commande.getId_client()==(id_client)) {
                commandesClient.add(commande);
            }
        }
        if(commandesClient.size()!=0)
           return Response.status(Status.OK).entity(commandesClient).build();

           return Response.status(Status.NOT_FOUND).build();
    }
```

   c- On souhaite ajouter une méthode « **clientsByName** », permettant la recherche d'une liste de clients par nom. Pour accéder à la méthode demandée, il faut suivre les étapes de test dans la figure 3.

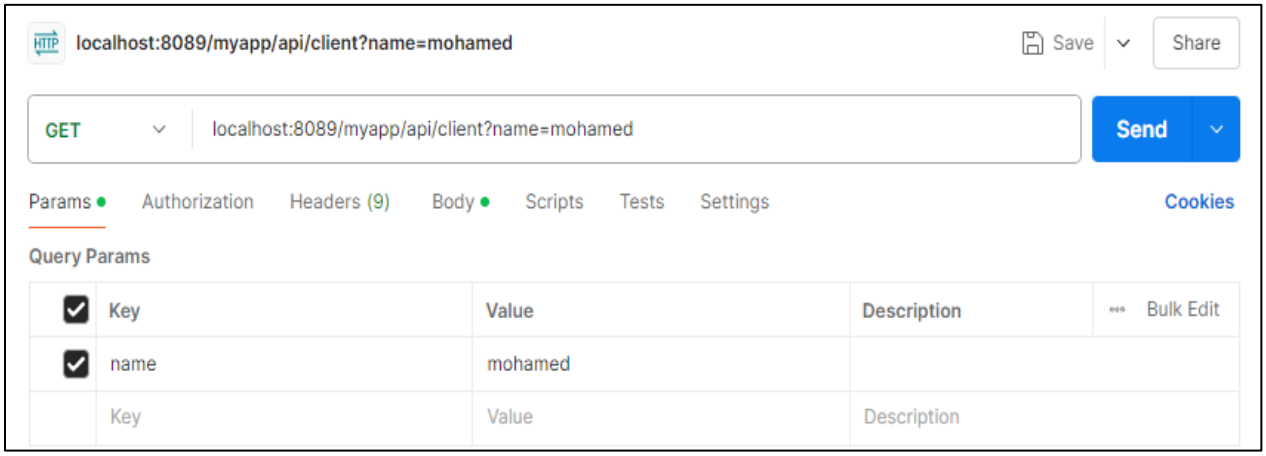

FIGURE 3 : Test de la méthode clientsByName

⇨ Selon la figure 3, on vous demande de compléter le code de la méthode « **clientsByName** ». **(1,25 pts)**

```
……[9]
……[10]
public ……[11] clientsByName……[12] String
name) {
        // Filtrer les clients par nom
        List<Client> clientsTrouves = lstClient.stream()
                .filter(client -> client.getNom().equalsIgnoreCase(name))
                .collect(Collectors.toList());
```

```
        // Vérifier si des clients ont été trouvés
        if (!clientsTrouves.isEmpty()) {
            return  Response.status(Status.OK).entity(clientsTrouves).build();
        }

            return Response.status(Status.NOT_FOUND).build();
    }
```

3. L'entreprise a décidé d'ajouter la méthode « **getAllClients** » pour lister tous les clients. Au cours du test, on a eu l'erreur 500 - Erreur interne du serveur (voir figure 4)
   Expliquer la cause de l'erreur et proposer une solution. **(1pt)**

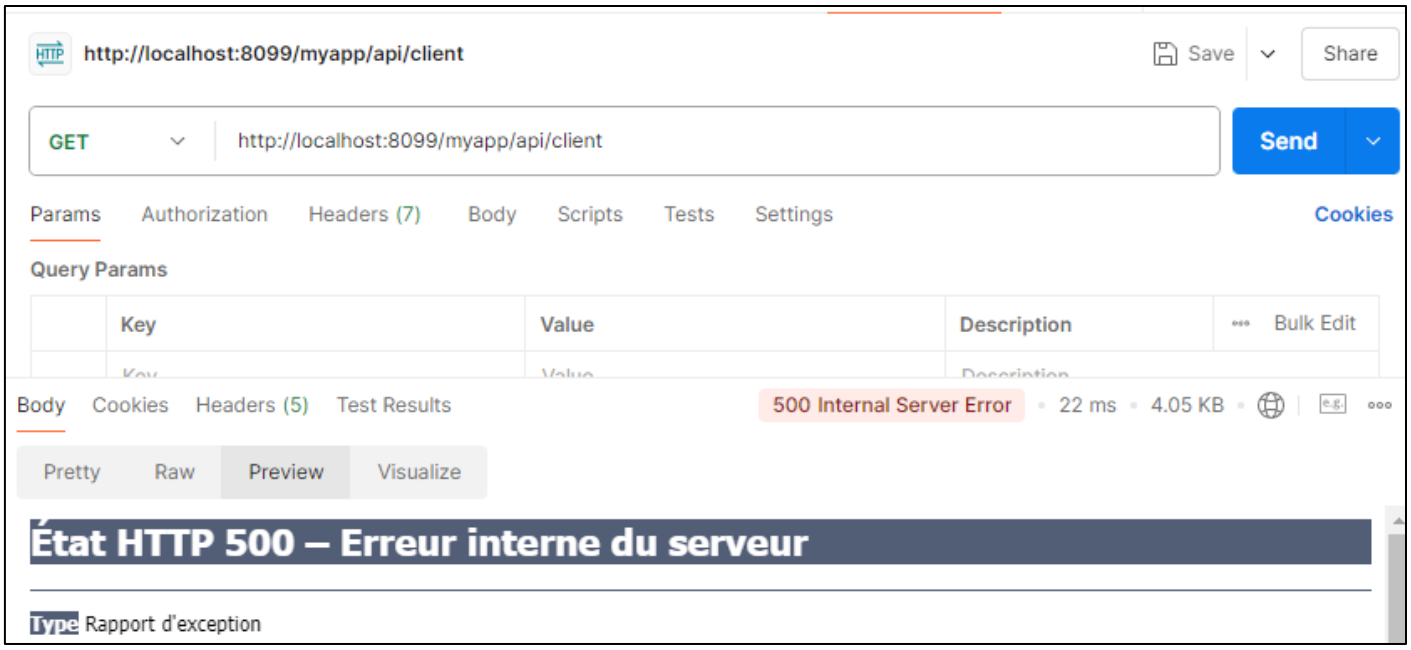

FIGURE 4 : Test de la méthode getAllClients

Partie 2 : GraphQL (6 pts)

On souhaite migrer vers l'API GraphQL

1. Expliquer la différence entre **l'API GraphQL** et **l'API REST**. (**0,75 pt**)
2. Terminer le bloc objet du fichier **schema.graphql** (voir figure **5**) (**1,75 pts**)

```
type Produit {
    id_p: Int
[1].....

}
type Client{
   ic_client: Int
[2].....
}

type Commande{
 [3]..... id_commande:Int
}

enum TypeProduit{
  [4].....

}
```

FIGURE 5 : Fichier schema.graphql

3. On vise définir et tester une méthode qui permet d'ajouter un ou plusieurs produits dans le stock.

L'implémentation de la méthode dans le fichier schema.graphql est comme suit :

```
type  Mutation {
    Addproduct(id_p :Int, nom: String, prix: Float, quantite:Int):[Produit]


}
```

Terminer les informations nécessaires pour tester la méthode sur Postman : (1 pt)

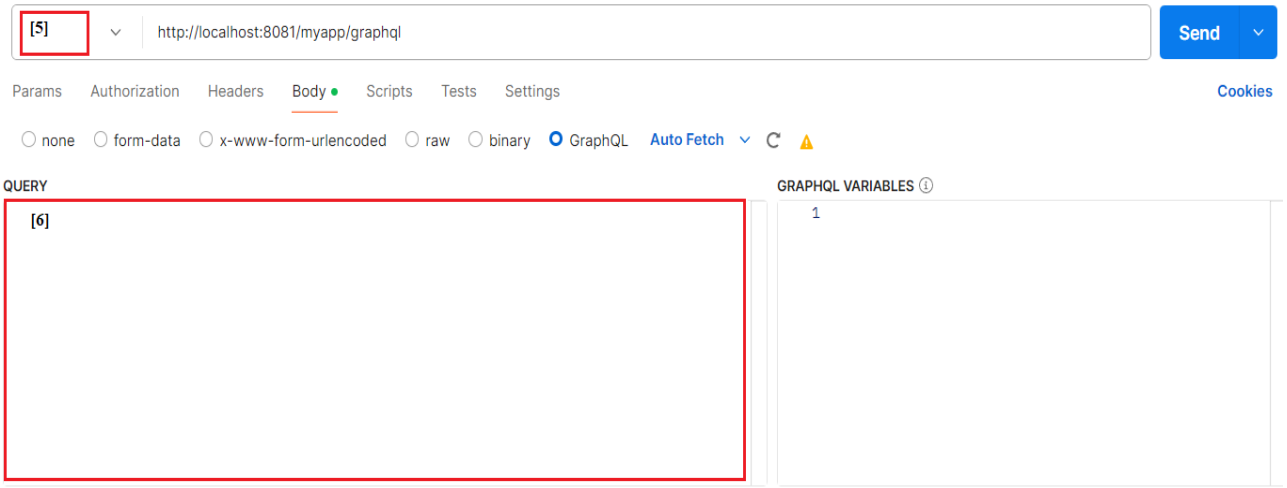

FIGURE 5 : Test de la méthode AddProduct

4. Dans le fichier schema.graphql, ajouter la méthode AddCommande. Sachant que pour chaque commande créée, on souhaite afficher ses détails. (**1 pt**)

5. Chaque client peut modifier la date de sa commande. On souhaite tester la méthode **updateCommande** sur postaman. Mais, le statut indiqué est 400 comme il est mentionné dans la figure 6. (**1,5 pts**)

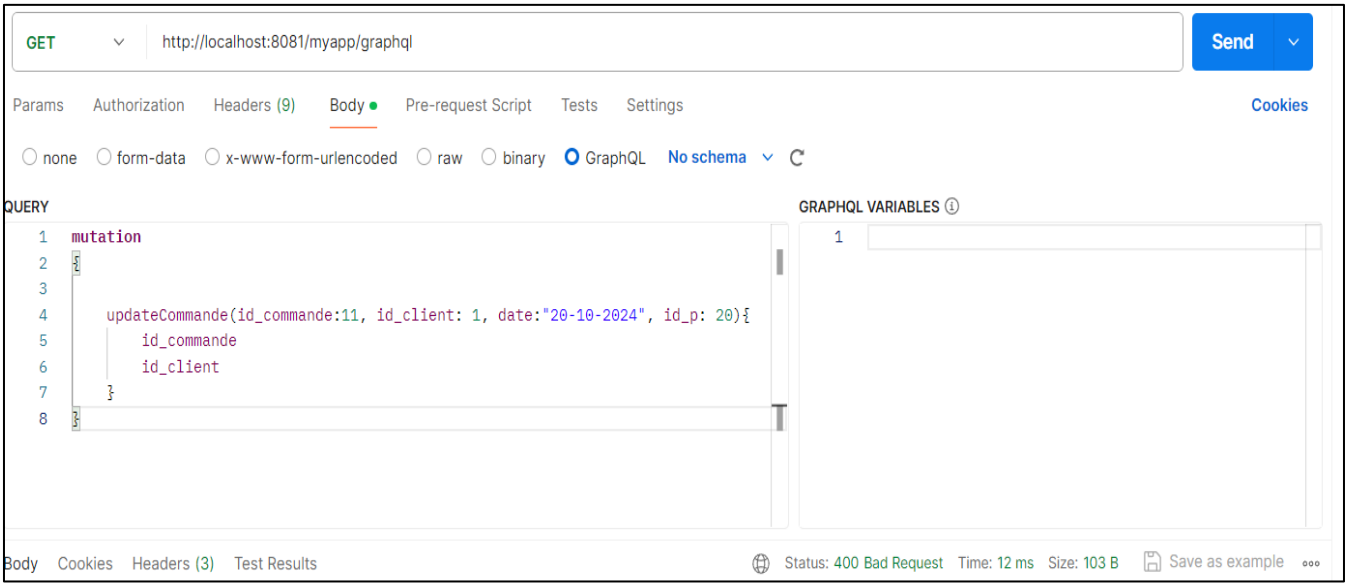

FIGURE 6 : Test de la méthode updateCommande

Expliquer l'origine de cette erreur et proposer une correction afin d'afficher seulement la nouvelle date modifiée.

Bon Travail ☺