2. conception REST

2.1 Lister les collections de ressources accessibles dans cette API

annonces, departement, categorie

Lister les URI pour l'ensemble des collections et pour les ressources de ces collections

GET: /annonces
GET: /annonces/departement
GET: /annonces/categorie

2.2 Indiquer la requête ou la séquence de requêtes nécessaires pour réaliser les opérations suivantes

Lister les annonces
GET: /annonces

Lister les annonces du 54
GET: /annonces/departement/54

Lister les annonces de la catégorie 'voitures'
GET: /annonces/categorie/voitures

Créer une catégorie
POST: /annonces/categorie

Modifier une annonce existante
PUT: /annonces/id_annonce

Créer une annonce, l'associer à une catégorie et un département
PUT: /annonces
PUT: /annonces/id_annonce/categorie
PUT: /annonces/id_annonce/departement

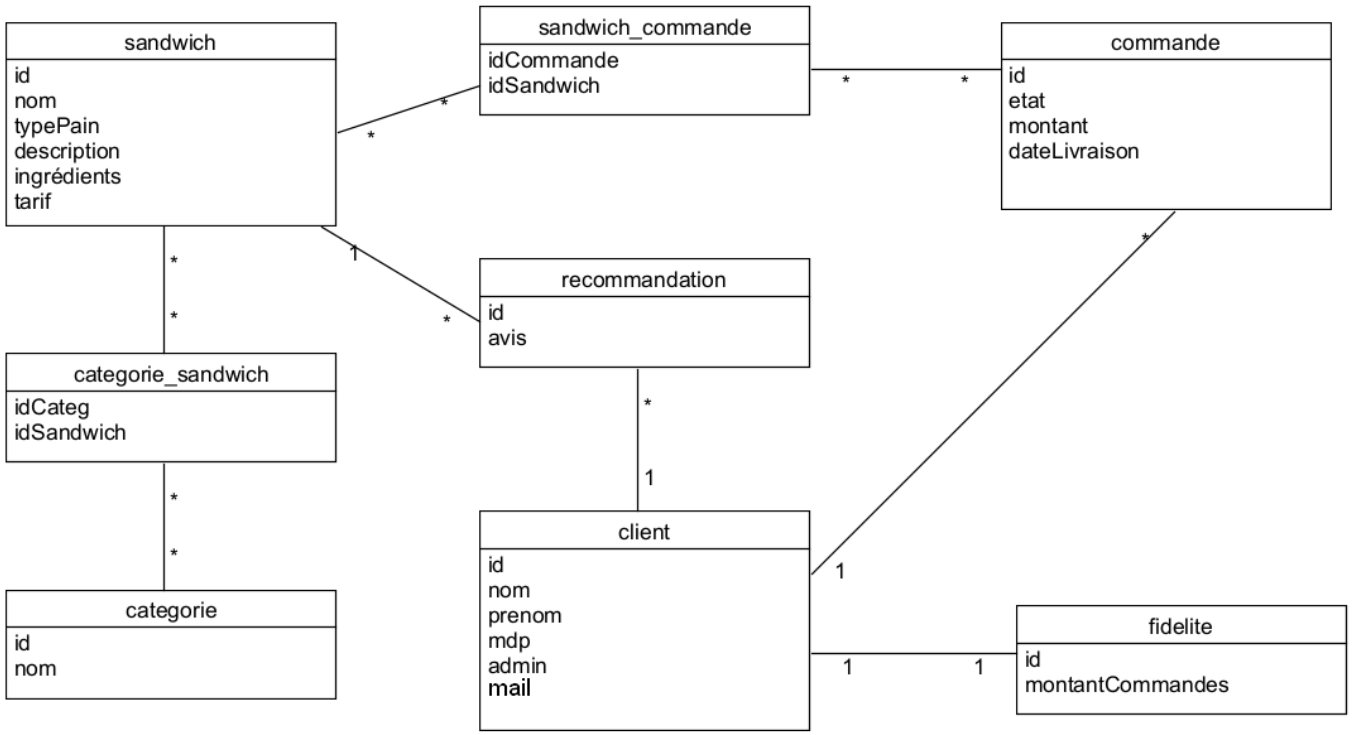
Modifier la catégorie d'une annonce
PATCH: /annonces/id_annonce/categorie

Ajouter une image à une annonce
PUT: /annonces/id_annonce/photo

Supprimer une annonce et toutes les images associées
DELETE: /annonces/id_annonce/photo
DELETE: /annonces/id_annonce

3. conception sur le projet

3.1 UML : Schéma de base de données



3.2 URI : Identification des URI et actions possibles sur chacune d'elles

GET : /sandwiches : Liste les sandwichs du catalogue

GET : /sandwiches/id_sandwich : Récupère un sandwich en particulier

PUT : /commandes : Crée une commande, on y ajoute : une heure et une date, des sandwichs et leur quantités

PATCH : /commandes/id_commande : Modifie sa commande (Payer, modifié l'état)

GET : /commandes/id_commande/etat : Retrouve l'état de la commande

GET : /commandes/id_commande/facture : Récupère la facture d'une commande

GET : /commandes/id_commande/etat/ : Récupère l'état d'une commande

3.3 Scénario

Le client crée une commande :

METHODE : PUT
URI: /commandes
DATA : {"client_name" : "Nom Client", "client_mail" : "nomclient@mail.com", "delivery" : { "date" : "20-20-20", "time" : "12:12:00"}, "items" : [{"uri" : "/sandwiches/c1215515123", "name" : "panini", "q" : 1, "price" : 2.50"}]}
STATUT: 200
HEADER : application/json

Le client modifie la date et l'heure de la commande :

METHODE : PATCH
URI : /commandes/id
DATA : { "type" : "response", "message" : "commande modifiée"}
STATUT : 200
HEADER : application/json

Il effectue le paiement puis interroge le service pour connaître l'état de cette commande :

METHODE : PATCH
URI: /commande/id_commande/etat
DATA: { "type" : "response", "message" : "commande payée"}
STATUT : 201
HEADER : application/json

METHODE : GET
URI : /commandes/id_commande/etat
DATA : { "type" : "response", "état" : "commande payée"}
STATUT : 200
HEADER : application/json