

2. conception REST

Lister les collections de ressources accessibles dans cette API :

- Annonce, Photo, Categorie, Region, Departement

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

- `{{host}}/annonces` ; `{{host}}/photos` ; `{{host}}/categories` ; `{{host}}/regions` ;
`{{host}}/departements`

Lister les annonces :

- `router.route('/annonces')` GET

Lister les annonces du 54 :

- `router.route('annonces/departement/:dep')` GET

Lister les annonces de la catégorie 'voitures' :

- `router.route('annonces/categorie/:categ')` GET

Créer une catégorie :

- `router.route('/categories/create')` POST

Modifier une annonce existante :

- `router.route('/annonces/modified/:id')` PUT

Créer une annonce, l'associer à une catégorie et un département :

- `router.route('/annonces/create?=categorie?=dep')` POST

Modifier la catégorie d'une annonce :

- `router.route('/annonces/modified/:categ')` PUT

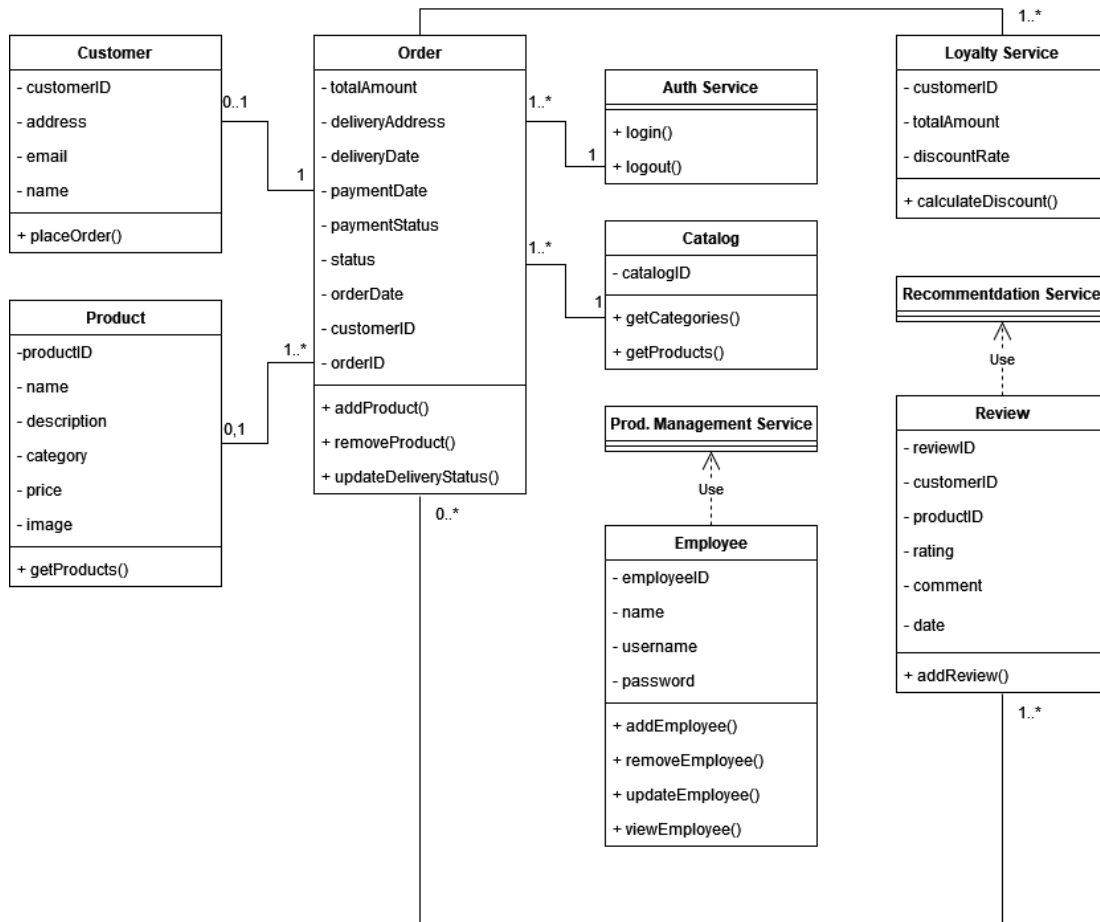
Ajouter une image à une annonce :

- `router.route('/:annonces/add/:images')` POST

Supprimer une annonce et toutes les images associées :

- `router.route('/:annonces/delete/:images')` DELETE

3. conception sur le projet



Entités du modèle UML:

Client : un client est identifié par un identifiant unique et possède des informations telles que son nom, prénom, adresse, numéro de téléphone, et adresse email.

Commande : une commande est identifiée par un identifiant unique et possède des informations telles que la liste des produits commandés, la date et l'heure de la commande, la date de livraison souhaitée, le montant total, et l'état actuel de la commande.

Produit : un produit est identifié par un identifiant unique et possède des informations telles que le nom, la description, le prix, la quantité disponible, et la catégorie à laquelle il appartient.

Paiement : un paiement est identifié par un identifiant unique et possède des informations telles que le montant payé, la date et l'heure du paiement, et le statut (accepté, en attente, refusé).

URI et méthodes HTTP associées à chaque entité:

Client

POST /clients : créer un nouveau client

GET /clients/{id} : récupérer les informations d'un client existant

PUT /clients/{id} : mettre à jour les informations d'un client existant

DELETE /clients/{id} : supprimer un client existant

Commande

POST /commandes : créer une nouvelle commande

GET /commandes/{id} : récupérer les informations d'une commande existante

PUT /commandes/{id} : mettre à jour les informations d'une commande existante

DELETE /commandes/{id} : supprimer une commande existante

Produit

POST /produits : créer un nouveau produit

GET /produits/{id} : récupérer les informations d'un produit existant

PUT /produits/{id} : mettre à jour les informations d'un produit existant

DELETE /produits/{id} : supprimer un produit existant

Paielement

POST /paiements : créer un nouveau paiement

GET /paiements/{id} : récupérer les informations d'un paiement existant

PUT /paiements/{id} : mettre à jour les informations d'un paiement existant

DELETE /paiements/{id} : supprimer un paiement existant

Scénario :

Un client souhaite créer une commande, la modifier, la payer puis suivre son état.

Le client crée une nouvelle commande en envoyant une requête POST à l'URI /commandes, avec les informations de la commande dans le corps de la requête (liste des produits commandés, date et heure de la commande, date de livraison souhaitée, etc.). Le serveur renvoie une réponse avec le statut HTTP 201 Created, contenant l'URI de la nouvelle ressource /commandes/{id}, où {id} est l'identifiant unique de la nouvelle commande créée.