

TD3 : modifier des ressources, conception d'une API

L'objectif du TD est d'implanter des actions de modification de ressource, puis de concevoir les api du projet "le bon sandwich".

1. Mise à jour de ressource

On souhaite programmer une action permettant de mettre à jour une commande au travers de l'api. On utilise la méthode PUT. On ne traite pas le contenu de la commande (items). Seules les données suivantes sont mises à jour : nom du client, mail du client, date et heure de livraison.

Identifier l'URI à utiliser puis programmer l'action :

- les données sont transmises au format json,
- pour des raisons de sécurité (injection), les données sont filtrées avant insertion dans la base,
- en cas de succès, la réponse doit contenir un code de retour 204 NO CONTENT ,
- en cas d'échec, la réponse doit contenir un message d'erreur et un code de retour correspondant à l'erreur (404, 500).

Vous paraît-il judicieux que cette action soit disponible dans l'API de façon ouverte ?

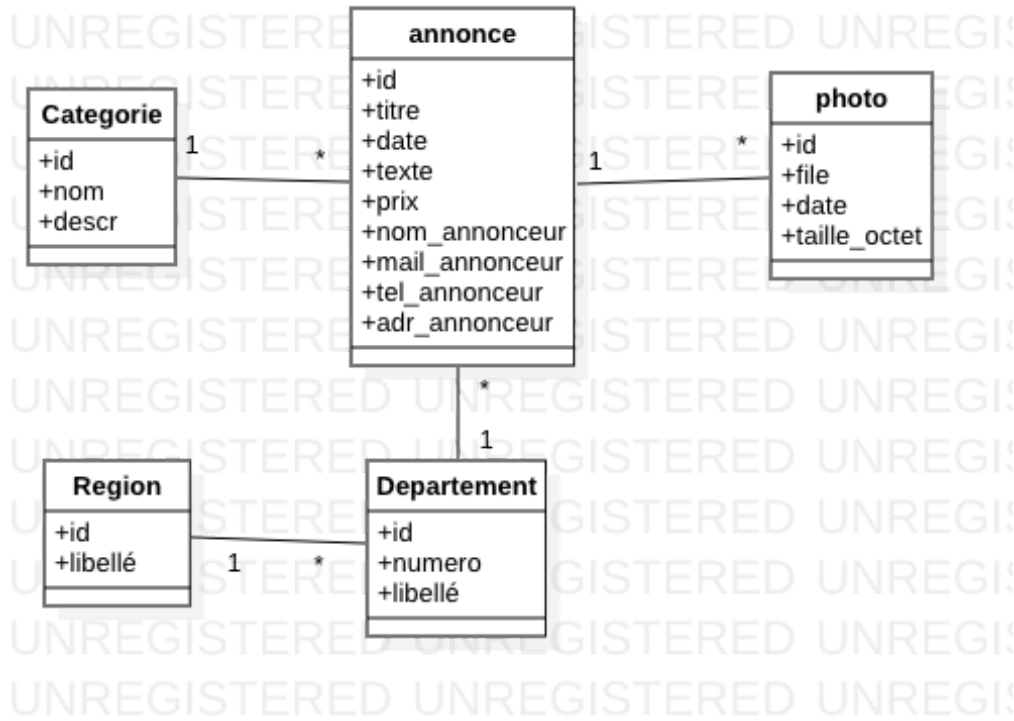
2. Conception REST

On souhaite construire une API RestFul pour un service de publication de petites annonces. Le modèle conceptuel du service est présenté dans le schéma en annexe.

- lister les collections de ressources accessibles dans cette API
- lister les URI pour l'ensemble des collections et pour les ressources de ces collections

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

- lister les annonces,
- lister les annonces du 54,
- lister les annonces de la catégorie 'voitures',
- créer une catégorie,
- modifier une annonce existante,
- créer une annonce, l'associer à une catégorie et un département,
- modifier la catégorie d'une annonce,
- ajouter une image à une annonce,
- supprimer une annonce et toutes les images associées.



3. Conception sur le projet "le bon sandwich"

Étudier le sujet du projet et identifier les bases de données et les applications backend qu'il faudra construire.

Le service de prise de commandes permet à un client de créer une commande, la modifier, la payer puis suivre son état (créée, payée, en préparation, prête, livrée).

- Établir le modèle UML des ressources manipulées par cette API,
- Identifier les URI des différentes ressources, et les actions possibles sur chacune d'elles en leur associant une méthode HTTP.
- Illustrer ces actions en déroulant un scénario dans lequel un client crée une commande, modifie sa date de livraison, effectue le paiement puis interroge le service pour connaître l'état de cette commande ; à chaque étape, expliciter la requête et la réponse Rest (méthode, uri, status, données, headers).