



Service API Design Lab

Auteur: Victor Sallé [salle@polytech.unice.fr]
Date: 8 octobre 2014

Sujet #3 : FedPS

Interface proposée

L'interface que je propose se décompose logiquement en trois services, chacun dédié à une cible utilisateur particulière : client, manager, livreur.

Client

Plusieurs fonctionnalités sont offertes au client :

- réaliser un devis;
- réserver un transport à partir de ce devis;
- réserver un transport de manière directe;
- suivre ses colis.

Pour réaliser un devis, les données fournies n'ont pas besoin d'être aussi précises que lors de la commande d'un transport : par exemple, au lieu de fournir l'adresse complète, le code postal et le pays sont suffisants pour calculer la distance. Aussi, le nom de l'expéditeur et du destinataire ne sont pas nécessaires. Idem pour le nom de la ville qui est calculé automatiquement sur le serveur à partir du code postal.

Le service étant international, il est cependant nécessaire de préciser plusieurs informations en plus des informations de base : la devise, l'unité de taille, ou encore l'unité de poids.

Ainsi, pour réaliser un devis :

— DEVIS

Entrées	Sorties
Adresse de départ : code postal & pays	Id du devis
Adresse de destination : code postal & pays	Coût

Entrées	Sorties
Date de prise en charge	Date estimée de livraison
Mode de transport (classique ou express)	
Devise	
Dimensions	
Unité de taille	
Poids	
Unité de poids	

Ensuite, l'utilisateur peut confirmer son devis. Il passe alors les informations plus importantes, comme son adresse email afin qu'il puisse être joint si nécessaire, et son numéro de carte bleue afin de payer la commande. On ne fournit plus que les informations nécessaires sur l'adresse. Aussi, un devis n'est valable que cinq jours, et la date estimée de livraison peut être recalculée.

— CONFIRMATION DE DEVIS

Entrées	Sorties
Id du devis	Id du colis
Numéro de carte bleue	Date estimée de livraison
Nom de l'expéditeur	
Mail de l'expéditeur	
Nom du destinataire	
Date de prise en charge réelle	
Adresse de départ : numéro et nom de rue	
Adresse d'arrivée : numéro et nom de rue	

Mais l'utilisateur peut également réserver directement le transport de son colis : auquel cas, il fournit toutes les informations précisées ci dessus, et récupère l'identifiant du colis ainsi qu'une date estimée de livraison.

Enfin, il est possible de suivre un ou plusieurs colis : il suffit d'indiquer leur identifiant, et le système retournera pour chacun d'eux un état actuel (en cours de route, bloqué aux douanes...) et une date estimée de livraison.

Concernant les remises personnelles, elles sont calculées automatiquement sur le serveur à partir de l'adresse email fournie par le client lors de la commande.

Manager

Le manager doit suivre des statistiques : il doit pouvoir connaître en direct l'état du réseau, et pouvoir suivre les colis qui sont dans un état particulier. Aussi, il doit pouvoir connaître les colis qui sont en retard.

— SUIVI DE L'ÉTAT DU RÉSEAU

Entrées	Sorties
	Le nombre de colis dans chaque état

— SUIVI DE LES COLIS DANS UN ÉTAT

Entrées	Sorties
Etat à surveiller	Tous les colis qui sont dans cet état

— SUIVI DES COLIS EN RETARD

Entrées	Sorties
	Tous les colis qui sont en retard par rapport à la date estimée de livraison

Livreur

Le livreur requiert deux fonctionnalités : un suivi d'itinéraire, et la possibilité de scanner des colis afin de mettre à jour leur état.

Un livreur peut, soit prendre en charge, soit livrer un colis. Il n'est donc pas nécessaire de préciser le nouvel état du colis lorsque celui-ci est scanné : s'il est en attente de prise en charge, alors c'est une prise en charge et son état passe à "pris en charge", tandis que s'il est en cours d'acheminement, c'est une livraison, et lors du scan son état passe à "livré".

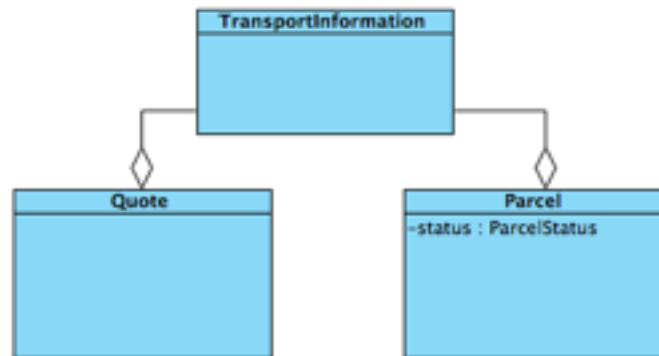
— SUIVI D'ITINÉRAIRE

Entrées	Sorties
Id du livreur	Adresse suivante
	Type d'arrêt (prise en charge ou livraison)
	Id du colis concerné

— MISE À JOUR DE L'ÉTAT D'UN COLIS

Entrées	Sorties
Id du colis (via le scanner)	

Objets métier



Une classe "TransportInformation" qui regroupe toutes les données utiles au transport.

Cela permet de créer facilement un colis à partir d'un devis : il suffit de récupérer son attribut et de se l'attribuer.

Choix d'implémentation

Les services dédiés aux manager et aux livreurs reposent sur une architecture REST. En effet, ils ne demandent que peu, voir aucun argument ; la légèreté de ce type d'architecture est donc appréciable ici.

De plus, on manipule ici des ressources : on accède à des listes de colis, à des adresses, on crée des devis.

Aussi, puisque REST repose sur le protocole HTTP, il est possible de sécuriser les webservices en utilisant ce protocole, ce qui est mieux que de gérer cela au niveau des beans.

Le service dédié aux clients quant à lui repose sur du SOAP. Il n'est pas concevable de passer une liste infinie d'arguments via l'URL.

Concernant les types renvoyés par le serveur, ce ne sont pas les entités manipulés en base. Cela me permet de "cacher" totalement la manière dont est conçue l'application, et de ne fournir que le nécessaire.

Aussi, cela me permet par exemple de donner les dates sous un chiffre (timestamp), même si elles sont sous format objet dans mon modèle : c'est un format universel qui sera bien plus facilement réutilisable ailleurs.