

Service Web : FedPS

1 INTERFACE DES SERVICES WEB

Afin de répondre au besoin défini dans le sujet FedPS. J'ai mis en place 4 service web :

- ManagerWs, pour les managers
- DriverWs, pour les transporteurs
- CustomerPrivate, pour nos clients (qui devront normalement s'authentifier)
- CustomerPublic, pour les clients de nos clients (pas besoin de s'authentifier)

1.1 MANAGERWS

Le service ManagerWs est basé sur le design pattern Document. Il expose une méthode « dispatch ».

Cette méthode permet de traiter trois actions différentes. Le manager peut récupérer le nombre de commande ayant un statut donné, récupérer toutes les commandes en fonction d'un statut, ou encore de donner une réduction sur la prochaine commande d'un utilisateur.

Cette méthode prend en paramètre le type de l'action que l'on souhaite effectuer. Puis en fonction, l'utilisateur devra fournir le statut de la commande s'il souhaite récupérer le nombre de commandes ou les commandes elle-même. Pour la dernière action (l'ajout d'une réduction) le manager doit fournir le nom du client et le pourcentage de réduction souhaité.

1.2 DRIVERWS

Ce service est basé sur le model RPC. Il expose deux méthodes. Une permettant de récupérer la tournée du transporteur et une autre permettant de mettre à jours le statut d'une commande.

La première méthode ne prend pas de paramètre et renvoie un objet Round décrit à la partie 2. La seconde méthode prend la référence du colis et le nouveau statut et informe si l'action s'est bien déroulée.

1.3 CUSTOMERPRIVATE

Ce service permet de gérer la demande de devis et la création de commande. Il est basé sur le model RPC. Pour cela, il expose trois méthodes.

« createQuote » permet de créer un devis et de le récupérer. L'utilisateur doit fournir les informations du colis (Largeur, hauteur, profondeur et poids), le nom et l'adresse de l'expéditeur et du destinataire, ainsi que la date à laquelle le transporteur devra venir chercher le colis.

« updateQuote » permet de mettre à jours la date de récupération du colis. Cela permet de ne pas avoir à renseigner de nouveau toute les informations du devis.

« orderQuote » permet de commander le devis passé en paramètre. La méthode lui retourne les informations du devis ainsi que le numéro de la commande afin de pouvoir suivre son statut.

1.4 CUSTOMERPUBLIC

Le service CustomerPublic utilise le model RPC et expose une méthode permettant de récupérer le statut d'une commande en fournissant uniquement la référence de celle-ci. Ce service permet non seulement à nos clients de connaître le statut de leur commande. Mais également à leur client.

2 OBJETS METIERS

Les différents objets métiers exposés à travers les services web sont les suivant.

La classe Address contient un nom de rue, un code zip, un nom de ville et l'identifiant du pays.

La classe Parcel contient un identifiant, les dimensions du colis, le nom et l'adresse de l'expéditeur et du destinataire, la date de récupération et le statut du colis.

La classe Quote contient un numéro de devis, un objet Parcel, un prix et une date de livraison estimée.

La classe RoundStep décrit une étape de la tournée d'un transporteur. Elle contient quelques informations de la classe Parcel comme l'identifiant du colis, les informations de l'expéditeur et du destinataire ainsi que la date de récupération.

La classe Round contient une liste de RoundStep.

L'énumération ParcelStatus contient les éléments suivant : ARRIVAL_SCAN, DELIVERED, STORAGE, CUSTOMS, IN_TRANSIT

La classe Customer contient le nom du client, son adresse, et ses possibles réductions (permanente ou temporaire).

3 CHOIX DE CONCEPTION

3.1 MANAGERWS

Ce service web utilise le pattern Document. Par exemple, l'utilisateur pourra récupérer le nombre de commande pour les statuts qui l'intéresse en utilisant qu'une seule requête. Ou encore, il lui sera possible d'ajouter des réductions à plusieurs utilisateurs en une seule requête. Pour ces raisons, j'ai décidé d'utiliser ce pattern afin de permettre à l'utilisateur de définir plusieurs actions que le service devra exécuter.

3.2 DRIVERWS

J'ai décidé d'utiliser le model RPC pour ce service web. L'utilisation du model Document ne m'a pas semblé approprié. En effet, le transporteur va effectuer une requête au début de sa tournée puis à chaque étape de celle-ci. Il ne va pas mettre à jours le statut de plusieurs commandes en une requête. Les méthodes proposées peuvent être appelé directement par l'application du transporteur en récupérant uniquement le Wsdl du service web.

3.3 CUSTOMERWS

Concernant les deux services web pour gérer les actions de nos clients. J'ai décidé d'utiliser de nouveau le model RPC afin de permettre l'utilisation de service web plus facilement qu'avec le model Document. J'ai hésité avec le model Document. En effet, celui-ci avait l'avantage de permettre à l'utilisateur de récupérer, par exemple, plusieurs devis en une requête.