

## SOA1

### Intégration : MisterDiscount

---

## **1. Architecture globale**

Trois services sont offerts par notre interface :

- passage d'une commande ;
- suivi d'une commande ;
- abonnement au catalogue.

Ces trois services sont exposés via un protocole SOAP (RPC ici). D'un point de vue métier, nous en avons distingué deux : gestion des commandes et gestion du catalogue. De ce fait, deux interfaces sont proposées : « Order » et « Catalogue ».

La première permet de passer une commande avec un certain nombre d'informations (cf. partie 2) et d'en suivre une à partir de son identifiant.

La seconde permet d'abonner un client existant à partir de son identifiant, d'abonner un nouveau client en spécifiant son adresse, mais également de récupérer la liste des adresses (adresse complète, contenant le nom du client) de tous les clients qui sont abonnés au catalogue.

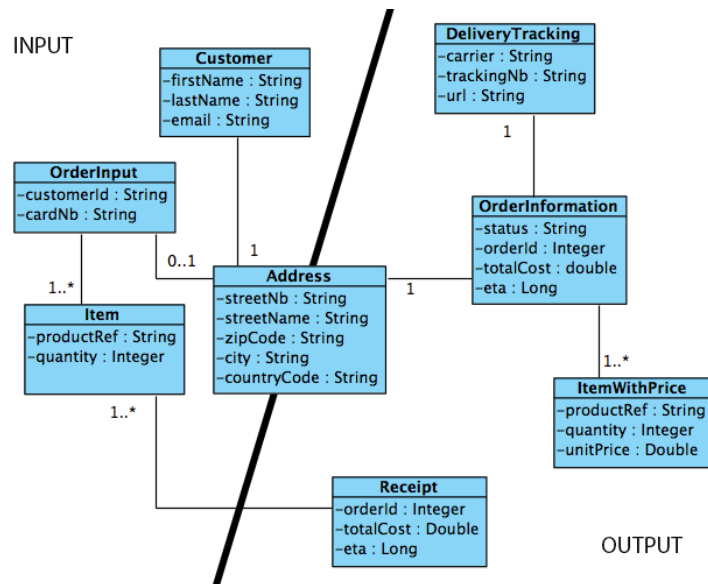
Nous aurions préféré proposer le service de suivi de commande de manière à l'orienter ressources, mais « Mule » n'est à priori pas conçu pour fonctionner comme cela. En effet, il n'est pas possible de préciser la méthode (i.e. « GET », « POST »...) pour un point d'entrée HTTP. Une façon de faire aurait été de filtrer la méthode avec un « Choice » sur « http.method », mais cela pas correct et ne facilite pas la mise à disposition d'un fichier WADL.

## **2. Objets métiers**

Un objet « OrderInput » est utilisé pour passer une commande. Pour passer une commande, un client doit être inscrit. Une adresse peut ou non être spécifiée : si elle est spécifiée, ce sera l'adresse de livraison ; si aucune adresse n'est fournie, la commande sera livrée à l'adresse d'inscription du client contenue en bases de données.

Un client peut s'abonner au catalogue. Il peut être déjà inscrit au service, auquel cas il s'abonne uniquement avec son identifiant, mais peut également être un nouveau client en s'abonnant via un objet « Customer ».

Lorsqu'il commande, le client récupère un reçu qui lui spécifie l'identifiant de sa commande, utile lorsqu'il voudra la suivre, ainsi qu'une date estimée de livraison, un coût total, et une liste des articles commandés.



Après avoir fourni l'identifiant de commande, le client peut récupérer les informations à propos de celle-ci. Il récupère alors un objet « OrderInformation », qui donne des informations sur le statut, la date de livraison estimée (qui peut évoluer), le coût total de sa commande, ainsi que les différents articles qui vont lui être livrés. Il dispose également des informations à propos du transport : le nom du transporteur, le numéro de suivi de colis, ainsi que l'URL lui permettant de suivre directement son colis sur le site du transporteur.

Aussi, des objets de type exception ont été mis en place (i.e. « OrderNotFoundException », « UnknownOperationException »...).

### 3. Flux d'intégration

Précisons d'abord qu'au sein de ces flux, seuls deux services ont pu être intégré : CreditGeneral et FedPS. En effet, les deux autres ne sont apparemment pas fonctionnels.

Le premier flux présent dans le projet est un flux d'initialisation de base de données (en mémoire). Celui-ci n'est pas important, mais il est tout de même nécessaire de le lancer avant d'exécuter les tests sous SoapUI.

Le projet expose deux point d'entrée : l'un pour le service de commande (création et suivi), l'autre pour le service d'abonnement au catalogue.

#### Création d'une commande

Le flux principal est celui de création d'une commande.

Nous supposons que le client doit être inscrit pour passer une commande. La première chose à faire est donc de vérifier que le client existe bien en base de données et chez TeamForce. Si une adresse a été spécifiée, la commande sera livrée à celle-ci, sinon on enrichit le message avec l'adresse par défaut du client.

Ce qu'il faut faire par la suite est de vérifier la disponibilité (tant en référence qu'en quantité) dans les entrepôts de RedWarehouse. S'il manque des objets, alors le message est modifié par la quantité maximale qu'il est possible de commander. Aussi, le message est enrichi avec l'adresse de l'entrepôt de RedWarehouse : cette adresse servira d'adresse de départ du colis pour FedPS. De même, on ajoute au message les informations à propos de chaque produit : poids et taille, dont on calculera le total pour le service de transport.

Maintenant que l'on connaît les objets effectifs de la commande, il faut établir un devis de transport avant de pouvoir effectuer le paiement. Le devis est donc effectué auprès de FedPS. Notons que le numéro de carte bleue fournie au service de paiement est celui de MisterDiscount plutôt que celui du client : le client règle l'ensemble auprès de MisterDiscount, et celui-ci se chargera de régler ses fournisseurs.

Dès lors, nous connaissons les objets, les quantités, et le montant de transport. Il faut donc calculer le total, et le transmettre au CreditGeneral. Si le paiement est validé, on poursuit, sinon une exception est levée.

Arrivé à cette étape, le paiement a été effectué. Il faut donc créer la commande auprès de FedPS (pour qu'il puisse effectuer le transport), de RedWarehouse (pour qu'il puisse gérer ses stocks) et de TeamForce (pour que l'on puisse connaître les commandes de chaque client).

Enfin, on retourne le reçu au client pour qu'il puisse suivre son colis et connaître ses informations de paiement.

### Suivi d'une commande

Pour suivre sa commande, le client fournit son numéro de reçu. La première chose à faire est donc de vérifier l'existence de cette commande auprès de TeamForce. Si la commande existe, on récupère toutes ses informations concernant les articles commandés et l'adresse de livraison.

Lorsque le client suit sa commande, il veut connaître sa date estimée de livraison. A partir du numéro de suivi de commande et du nom du transporteur récupéré à l'étape précédente, nous sommes en mesure d'aller récupérer les informations directement auprès du transporteur.

Nous contactons donc dans le cas présent FedPS, et nous récupérons la date estimée de livraison. Aussi, nous fournissons au client un lien direct de suivi de sa commande sur le site de FedPS.

Enfin, il faut mettre à jour le statut de la commande. Dans la version présentée, le statut est automatiquement à « SENT ».

### Abonnement au catalogue

Dans la version présentée, on ne peut réellement parler de flux d'intégration pour cette partie. En effet, la première chose faite est de vérifier l'existence du client en base de données, puis nous mettons à jour le fait qu'il souhaite recevoir le catalogue.

Idéalement, cette information aurait pu être gérée auprès de TeamForce.

## Récupération des adresses de livraison du catalogue

De même que précédemment, ici on ne fait que lire les informations contenues en base de données (et idéalement chez TeamForce).

C'est une liste d'adresses de livraison qui est reçue en réponse.

## Améliorations envisageables sur le service de catalogue

Comme précisé, il aurait fallu que les informations à propos des produits soient récupérées sous une forme « Google Product Feeds ». Pour ce faire, on aurait pu récupérer tous nos articles auprès de RedWarehouse ; puis, avec un « DataMapper », nous aurions transformé cette liste en un document XML répondant au format demandé.

On aurait alors pu récupérer auprès de TeamForce les adresses des clients abonnés, et, à partir de leur nombre, envoyer la demande auprès d'un service d'imprimerie.

C'est enfin notre service de livraison FedPS qui aurait acheminé les catalogues de l'imprimerie jusqu'aux clients.

Ce flux aurait été intéressant à mettre en place puisque d'une même manière que le flux de passage de commande, il s'agissait vraiment ici d'orchestrer chacun des différents services mis à disposition.

## **4. Modifications des services initiaux**

### FedPS

- Découpage en métiers plutôt qu'en rôle
- Le métier de passage de commandes a vu son interface passer de document à RPC

### TeamForce

- Découpage en métiers plutôt qu'en rôle

### RedWarehouse

- Développement des services

## **5. Implication des membres**

### Swan Jumelle-Dupuy

- Conception des objets métiers

### Hugo Martinez

- Conception des interfaces des services

### Victor Sallé

- Développement des flux d'intégration
- Tests