Exercices

RESTfull Web Services

Réservation de trains

L'exercice consiste à développer un service REST pour la gestion de réservations de trains. Les objectifs sont :

- Développer un service web REST à partir d'une classe Java.
- Utilisation d'un Sub-Resource Locator.
- Mise en place d'un CRUD complet.
- Publier en local.

Description

Le service web REST de cet exercice consiste à créer un système CRUD pour l'interrogation et la réservation de trains. Les ressources manipulées par les services sont donc un train et une réservation. Le service web REST doit pouvoir lister l'ensemble des trains, lister les trains qui satisfont un critère de recherche (ville de départ, ville d'arrivée, jour de départ et un intervalle de temps), de créer, de modifier, de supprimer et de lister une réservation pour un client donné.

Les étapes de réalisation

Étape 1 - Classe Train

Créer la classe Train qui modélise le concept de Train et qui contient un attribut (identifiant fonctionnel d'un train), un attribut (la ville de départ du train), un attribut (la ville d'arrivée du train), un attribut (heure de départ depuis la ville de départ).

Ajouter des modificateurs et des accesseurs sur tous les attributs.

Étape 2 - Classe BookTrainBD

Créer une classe « BookTrainBD » qui conserve la liste des trains pour le service web.

Cela nécessite la définition de méthodes et de variables de classes (static).

Étape 3 - Classe TrainResource

Créer la classe TrainResource permettant l'accès aux services web REST de la ressource Train. Définir comme chemin de ressource racine la valeur /trains (utilisation de l'annotation @Path) puis ajouter trois méthodes qui permettent respectivement de retourner 1) la liste des trains 2) un train en fonction de son identifiant fonctionnel et 3) une recherche de trains par critères passés en paramètres de la requête (ville de départ, ville d'arrivée et heure de départ). Pour cette dernière méthode, le sous-chemin associé est /search. Noter que le format de retour des services est du XML.

Étape 4 - Tests

Testez votre service[1] :

- http://localhost:8080/rest/trains.
- http://localhost:8080/rest/trains/numTrain-TR123.
- http://localhost:8080/rest/trains/search?departure=poitiers&arrival=paris&arrivalhour=1250.

Étape 5 - Classe BookTrain

Créer la classe BookTrain qui modélise le concept de réservation de Train et qui contient un attribut (identifiant fonctionnel de la réservation d'un train), un attribut Train currentTrain (association sur le train de la réservation), un attribut int numberPlaces (nombre de places réservées). Ajouter des modificateurs et des accesseurs sur tous les attributs.

Étape 6 - Classe BookTrainBD

Modifiez la classe « BookTrainBD » pour qu'en plus de la liste des trains, elle conserve aussi la liste des réservations pour le service web.

Étape 7 - Classe BookTrainResource

Créer la classe BookTrainResource permettant l'accès aux services web REST de la ressource BookTrain. Quatre méthodes sont à définir :

1. La première est invoquée pour la création d'une ressource Réservation (méthode POST).
2. La deuxième (méthode GET) est utilisée pour lister l'ensemble des réservations.
3. La troisième getBookTrain(méthode GET) permet de retourner les informations d'une réservation à partir d'un numéro de réservation. Finalement removeBookTrain(méthode DELETE) permet de supprimer une réservation.

Étape 8 - Classe TrainResource

L'accès à la ressource Réservation (via la classe) est obtenu via l'utilisation d'un sub-resource locator. Compléter la classe « TrainResource ».

[1] Le numéro de port dépend du serveur utilisé pour le déploiement. Ici, nous faisons l'hypothèse de Tomcat.