

Sujet de TP 3

Framework JEE

EJB et services Web

Chouki Tibermachine - `chouki.tibermachine@umontpellier.fr`

Objectifs du TP

- ▶ Programmer un client de service Web
- ▶ Déployer une application avec un client de services Web dans un serveur d'application

Exercice

- ▶ L'objectif de cet exercice est de modifier l'application écrite dans les TP EJB précédents pour que la conversion soit effectuée non pas en analysant des fichiers XML accessibles en ligne, mais en invoquant les opérations d'un service Web
- ▶ Nous allons utiliser un service Web REST, qui s'apprête mieux à notre exercice et à ce type de requêtes assez simples (ne nécessitant pas une sauvegarde d'état entre les invocations ou un contexte transactionnel). Par ailleurs, les services REST n'utilisent généralement pas des enveloppes SOAP pour créer les paramètres et les valeurs de retour des opérations fournies par les services. Ils s'appuient dans ce cas uniquement sur les requêtes et réponses HTTP (correspondant aux méthodes HTTP Get, Post, ...). Ceci rend les échanges entre le client et le service Web plus légers et donc plus efficaces (surtout pour la conversion du montant dans toutes les monnaies disponibles, dans notre exercice).
- ▶ Nous allons utiliser le service Web REST de conversion de monnaie documenté et fourni par le site suivant : `http://currencies.apps.grandtrunk.net/`
- ▶ Créer une nouvelle application d'entreprise (JEE) et copier dedans les fichiers du TP 1
- ▶ Modifier le bean session pour obtenir les codes de monnaies et le taux de change :
 - Pour obtenir tous les codes de monnaies, il faudra simplement utiliser l'URL suivant `http://currencies.apps.grandtrunk.net/currencies` de la façon suivante :

```
URL url = new URL("http://currencies.apps.grandtrunk.net/currencies");
BufferedReader in = new BufferedReader(new InputStreamReader(url.openStream()));
```

- Il faudra ensuite parcourir le message (flux, ici) texte retourné par le service en utilisant la variable `in` :

```
String inputLine;
while ((inputLine = in.readLine()) != null) {
    // Dans inputLine, nous avons le code d'une monnaie
    ...
}
in.close();
```

- Pour obtenir le taux de change de l'Euro vers une monnaie cible, nous allons utiliser l'URL suivant `http://currencies.apps.grandtrunk.net/getlatest/<fromcode>/<toencode>` de la façon suivante :

```
url = new URL("http://currencies.apps.grandtrunk.net/getlatest/"
              +"EUR"+"/" +inputLine);
BufferedReader in2 =
    new BufferedReader(new InputStreamReader(url.openStream()));
```

- Récupérer le taux de change dans : `in2.readLine()`
 - Faire attention ici aux monnaies qui sont parfois disponibles dans la liste des monnaies (1er URL), mais dont les taux de change ne sont pas disponibles (2nd URL). Cela retourne une erreur HTTP 400 lors de l'appel à `openStream()` sur l'objet `url`, et ça provoque une exception de type `IOException`. Ajouter ici un `try-catch` pour capturer cette exception et journaliser (loguer) ce code de monnaie. La conversion dans les autres monnaies doit se poursuivre après
- Déployer l'application et la tester sur votre navigateur