

Utilisation des sequences Corba IDL en java

Le langage IDL de Corba autorise l'utilisation de séquences, que l'on peut assimiler à des tableaux de taille variable de n'importe quel type de donnée.

La problématique: Comment utiliser les séquences en particulier lorsqu'elles sont projetées dans le langage cible java?

Exemple de fichier IDL:

```
{mostip}struct MatchStructure {
    string home;
    string visitor;
    short homeGoals;
    short visitorGoals;
};

typedef sequence<MatchStructure> DayOfChampionship;{/mostip}
```

Dans cet exemple ci-dessous, comment manipuler la séquence "DayOfChampionship"?

La réponse passe par un peu de lecture.

si on lit la référence de l'OMG, "IDL to Java Language Mapping, v1.2" (ici <http://www.omg.org/cgi-bin/doc?formal/02-08-05>) on peut lire section 1.10 "Mapping for Sequence" qu'une séquence est tout simplement convertie en un tableau Java ("An IDL sequence is mapped to a Java array with the same name").

Donc cela ne passe pas du tout par les classes générées holder (utiles pour la passage de paramètre en out et inout) ou helper (qui fournit des fonctions statiques pour manipuler le type).

Pour manipuler un type de séquence, ici DayOfChampionship, il faut l'utiliser dans une interface à l'intérieur d'un fichier IDL, en le faisant apparaître dans des prototypes de méthodes.

Pour information il n'est par ailleurs nullement obligatoire de déclarer les types à l'intérieur d'une interface.

Ces prototypes de méthodes seront ensuite convertis en java dans des tableaux du type contenu de la séquence, ici des tableaux de MatchStructure[].

exemple:

```
{mostip}interface Essai {
    DayOfChampionship enTypeRetour();
    string enParametre(in DayOfChampionship a);
}    {/mostip}
```

sera converti en :

```
{mostip} MatchStructure[] enTypeRetour();
String enParametre( MatchStructure[] a);{/mostip}
```

Pour information, après projection (autrement dit conversion) IDL vers java, les méthodes sont définies dans EssaiOperations.java (car j'ai appelé l'interface Essai).

Ensuite le développement est plutôt "classique", où il faut implémenter l'interface Essai en créant une classe qui hérite de EssaiPOA (qui lui-même hérite de EssaiOperations), autrement dit un servant.

Un conseil pour travailler avec des tableaux de taille variable en java: utiliser la classe Vector (voir documentation du JDK ou mon cours de java). Cette classe dispose d'une méthode toArray() qu'on peut utiliser au moment des appels corba.

C'est facile à manipuler, par contre ce n'est pas rapide.