

## INF 4001 – Intergiciels répartis et mobiles

### Axis 2 – Services web POJO

## 1 Installation et environnement

Axis2 est un ensemble de bibliothèques et d'outils Open-Source supportant les services Web en Java. Vous trouverez la documentation relative à Axis2 à l'URL <http://axis.apache.org/axis2/java/core>.

La version 1.7 d'Axis2 permet notamment de fabriquer des clients et des fournisseurs de services web exploitant les descripteurs WSDL (versions 1.1 et 2.0) et le protocole SOAP (versions 1.1 et 1.2). Axis2 fournit également une plate-forme légère d'hébergement des fournisseurs de services web.

Pour pouvoir utiliser Axis 2 en TP, vous devez

- Dézipper l'archive `axis2-1.7.8-bin.zip` du forum dans votre répertoire personnel (ou un sous répertoire)
- Positionner la variable `JAVA_HOME` sur le répertoire contenant le JDK (la version 8 convient)
- Positionner la variable `AXIS2_HOME` sur le répertoire d'installation d'Axis2 que vous venez de créer, typiquement `~/axis2-1.7.8` et faire en sorte que l'ensemble des jar du sous-répertoire `lib` de ce répertoire d'installation soit dans votre `CLASSPATH`. La commande bash suivante permet d'ajouter ces jar à votre `CLASSPATH` :  

```
export CLASSPATH="$CLASSPATH:$(echo $AXIS2_HOME/lib/*.jar | tr ' ' ':')
```

La plate-forme Axis2 est lancée grâce à la commande `$AXIS2_HOME/bin/axis2server.sh`. On lance essentiellement ainsi un serveur HTTP (par défaut sur le port 8080), ce serveur servant de point d'entrée aux clients des services déployés localement dans la plate-forme.

Le déploiement d'un service se fait par copie de l'archive `.aar` contenant ce service dans le répertoire `$AXIS2_HOME/repository/services`. Axis détecte automatiquement les changements opérés sur ce répertoire. L'ajout d'un fichier `.aar` sert donc de déploiement de service et la suppression du fichier fait en sorte de désenregistrer le service.

La liste des services déployés est affichée en envoyant – typiquement via un navigateur web – une requête HTTP à l'URL `http://localhost:8080/axis2/services` (dans le cas où le serveur Axis a été lancé sur la machine locale).

On peut vérifier qu'au lancement de la plate-forme, le service *Version* est bien déjà déployé et qu'il répond à l'opération *getVersion* (aux URL `http://localhost:8080/axis2/services/Version` et `http://localhost:8080/axis2/services/Version/getVersion`).

## 2 Application Web selon l'approche POJO

Les services web et les clients web mis en place dans ce TP suivront l'approche POJO (*Plain Old Java Object*). Le *POJO Guide* de la documentation Axis2 pourra être utilement consulté.

### 2.1 Service web simple

Fabriquez un service web *Bonjour* offrant les opérations correspondant aux signatures de méthodes suivantes :

- `String disBonjourToutLeMonde()` renvoyant une chaîne de caractères « Bonjour tout le monde ! »
- `String disBonjour(String prenom)` prenant en paramètre un prénom (par exemple Jean) et renvoyant une chaîne de caractère du genre « Bonjour Jean ! »

Déployez ce service dans la plate-forme Axis 2 et testez-le via un navigateur Web via les requêtes résultant des URI `http://localhost:8080/axis2/services/Bonjour/disBonjourToutLeMonde` et `http://localhost:8080/axis2/services/Bonjour/disBonjour?prenom=Jean`.

Faites en sorte que votre service soit accessible sur le réseau du Cyberlab. Testez de nouveau votre service, en lançant le serveur et le navigateur sur deux machines distinctes.

## 2.2 Client Java

Modifiez ce service en rajoutant une méthode permettant de fixer la langue favorite (Français, Anglais ou Espagnol) de l'utilisateur. On passera à cette méthode une chaîne de caractères codant le pays (« fr », « en », « es »). Une fois la langue fixée, les appels suivants aux méthodes décrites plus haut renverront des messages dans cette langue.

Écrivez un client qui effectue une série d'appels aux méthodes du service *Bonjour*. Exécutez ce client sur une machine différente de celle sur laquelle est déployé le service.

## 2.3 Paramètre complexe

Modifiez le service web *Bonjour* (et le client correspondant) de sorte que la méthode *disBonjour* ne prenne plus en paramètre une simple chaîne de caractère mais un objet de type *Personne*. La classe *Personne* comportera trois attributs : un nom (chaîne de caractère), un prénom (chaîne de caractère), et un âge (entier) et offrira les accesseurs (getters/setters) adéquats.