



## Tutorial « Web Services » - 1 au 2 février 2006

Disponible en ligne : [http://cdsww.u-strasbg.fr/TP\\_session2.html](http://cdsww.u-strasbg.fr/TP_session2.html)

Avant propos : les expressions « Web Service(s) », « service(s) Web XML », « service(s) Web SOAP » seront considérées comme équivalentes dans ce document

But(s) de ce T.P. :

- . apprendre à installer un serveur Tomcat/Axis et à implémenter son premier « Web Service ».
- . familiariser l'auditeur avec les aspects serveurs des Web Services XML.

**Rappel:** Pour démarrer Tomcat, allez dans le répertoire « ../tomcat/bin » et tapez « ./catalina.sh start », « ./catalina.sh start » permettra de l'arrêter. Pour vérifier que le serveur est maintenant actif, il suffit par exemple de démarrer le navigateur Mozilla et de taper l'URL « <http://localhost:8080/axis/services> ».

### Exercice 1 :

**Action** : Se placer dans le répertoire « ../tomcat/webapps/axis », télécharger le fichier [Addition.java](#) dans ce répertoire, quittez l'édition puis renommez ce fichier en [Addition.jws](#), dans Firefox (ou autre), tapez l'URL : <http://localhost:8080/axis/Addition.jws?wsdl>

Le service apparaît-il lorsque l'on accède à la page <http://localhost:8080/axis/services> qui liste les services déployés ??

### Explications

### Exercice 2 :

**Action** : Rester dans le répertoire « ../tomcat/webapps/axis » et télécharger le fichier [Addition.tar](#), puis taper la commande "tar xvf Addition.tar"

Vous trouverez dans ce répertoire :

. les sous répertoires suivants :

. src

. lib

. build

. les fichiers suivants :

. build.xml

. deploy.wsdd

. undeploy.wsdd

Explications :

Le répertoire « src » comporte les fichiers sources du service à implémenter, « lib » contient les éventuelles bibliothèques nécessaires, « build » est un répertoire accueillant le résultat de la compilation des fichiers du répertoire « src ».

Le fichier « build.xml » est en fait un fichier de type « makefile » qui contient le processus d'implémentation.

Le fichier « deploy.wsdd » contient une description de ce qui doit être déployé.

Le fichier « undeploy.wsdd » est uniquement utile si l'on souhaite désinstaller par la suite le service.

**Action** : pour déployer le fichier, tapez la commande « ant deploy »

Dans votre navigateur, tapez l'URL : <http://localhost:8080/axis/services/Addition?wsdl>

On peut constater que le service apparaît maintenant lorsque l'on accède à la page <http://localhost:8080/axis/services> qui liste les services déployés

Pour le désinstaller, tapez la commande « ant undeploy », une nouvelle consultation de la page <http://localhost:8080/axis/services> permet de constater que le service n'est plus référencé.

Quelques constatations...

### **Exercice 3** :

Idem exercice 2 avec [UCD.tar](#) et [AstroCoo.tar](#)

Commentaires.

### **Compléments** : Cas d'utilisations et remarques

Quelques cas d'utilisations :

- On dispose d'un serveur Web accessible via des cgi : dans ce cas il est intéressant d'offrir un accès via Web Services en se contentant d'encapsuler les services existants dans des Web Services. Les traitements sont toujours effectués de la même façon et le serveur de Web Services n'est qu'un portail d'accès.

- On dispose ou l'on développe des bibliothèques Java et l'on souhaite offrir des services distants exploitant ces développements.

- On dispose de routines écrites en langage C (ou autres s'il est possible de réaliser des appels depuis Java). Il peut s'avérer intéressant de les « recycler » sous forme de Web Services, les langages cibles des utilisateurs pouvant être variés.

A noter qu'il existe une version C++ d'Axis qui est en développement.

### Remarque :

Afin de se mettre en conformité avec les standards actuellement en discussion au sein du groupe de travail IVOA « Grid et Web Services », il convient de consulter régulièrement les pages de ce groupe. A titre d'exemple, le document « VO Standard Interfaces » va passer prochainement à l'état de recommandation et il convient de tenir compte des interfaces qui devront obligatoirement être présentes dans tous les Web Services du VO.

## **Conclusion**

Nous avons constaté lors de ce T.P. qu'il est aisé d'utiliser ou de mettre en œuvre des services Web SOAP.

Au niveau du serveur, le déploiement de services Web SOAP est relativement facile. Par contre il convient d'assurer au niveau du serveur une disponibilité très élevée.

En effet, l'intégration de composants distants dans une application la rend dépendante de la fiabilité et de la disponibilité de ces composants. Fournir ce type de service implique donc de mettre en œuvre diverses techniques afin de tendre vers une disponibilité quasi sans faille (miroirs, surveillance, ...).

## **Références**

- [1] CDS Developer's corner, <http://cdsweb.u-strasbg.fr/devcorner.gml>
- [2] NVO Web Services, <http://voservices.org/webservices.aspx>
- [3] eStar, <http://www.estar.org.uk/>
- [4] Grid and Web Services WG, <http://www.ivoa.net/twiki/bin/view/IVOA/IvoaGridAndWebServices>
- [5] ADASS XIII Tutorial, <http://www.adass.org:8080/meetings/adass2003/events/tutorial>
- [6] SOAP-Lite, <http://search.cpan.org/dist/SOAP-Lite/>
- [7] Java, <http://java.sun.com/>
- [8] Tomcat, <http://jakarta.apache.org/tomcat/>
- [9] Axis, <http://ws.apache.org/axis/>
- [10] Ant, <http://ant.apache.org/>
- [11] DIME, <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/dimeindex.aspx>