

## Tutorial « Standards et Outils de l'Observatoire Virtuel »

### *T.P. « Web Services »*

*Ce document est disponible en ligne : <http://cdsweb.u-strasbg.fr/cdsws/tpWS.html>*

[Télécharger](#) les exemples du T.P.

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

Le but de ce T.P. est de familiariser l'utilisateur avec les services Web XML basés sur le protocole SOAP. Dans une première partie nous aborderons l'aspect Client en intégrant des « Web Services » du VO dans des applications Java et Perl.

L'aspect Serveur sera abordé dans une deuxième partie, l'auditeur apprendra à installer un serveur Tomcat/Axis et à implémenter son premier « Web Service ».

● Ce repère indique les questions ou les actions à mener

La configuration du poste de travail :

Pour installer un serveur de Web Services (Unix, Linux, MacOS X ou Windows) il faut :

- Télécharger et installer
  - une version récente de Java (\*)
  - la dernière version stable de Jakarta Tomcat
  - la dernière version stable d'Axis (\*)
  - recopier le répertoire axis du package Axis dans le répertoire « webapps » de tomcat
  - une version d'Ant (équivalent à un make)
  - ajouter au CLASSPATH les bibliothèques présentes dans le répertoire « ../axis/WEB-INF/lib » (utile lors des déploiements de nouveaux Web Services) (\*)

Ce travail est déjà réalisé sur les postes de travail du tutorial.

Nous utiliserons l'environnement Eclipse.

● Vérifiez qu'un répertoire « tomcat » est présent dans « /home/asov/Tools » puis allez dans le répertoire « ../tomcat » et tapez « ./catalina.sh start ». 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> ». Pour arrêter le serveur il suffit de taper la commande « ./catalina.sh stop ».

Ouvrez une fenêtre du navigateur Mozilla avec l'URL de la page « <http://cdsweb.u-strasbg.fr/devcorner.gml> », rubrique « XML Web Services Corner ». Cela vous sera très utile durant le T.P.

Ainsi configuré, votre poste de travail permet également de créer des clients pour accéder à vos propres Web Services ou à d'autres.

Si vous n'êtes intéressés que par la création de clients (en langage Java), seules les étapes (\*) sont utiles.

### Partie 1 : Comment « consommer » les services Web SOAP du VO ?

#### Exercice 1 : Utilisation du service Sesame du CDS

##### ● Se placer dans le répertoire « ../workspace/Sesame »

Remarque : les Web Services du CDS sont déployés sur des plateformes Linux – Tomcat/Axis

Adresse du service :

<http://cdsws.u-strasbg.fr/axis/services/Sesame>

Ou

<http://vizier.cfa.harvard.edu:8080/axis/services/Sesame>

##### ● Comment obtenir la description de ce service ?

#### Cas d'un client Java

Dans un premier temps il faut créer un « proxy » à partir de la description du service.

Ce proxy est en fait un package Java qui fournit localement les classes Java nécessaires à l'accès au service distant.

##### ● Tapez les commandes suivantes :

```
. java org.apache.axis.wsdl.WSDL2Java http://cdsws.u-strasbg.fr/axis/services/Sesame?wsdl
```

```
. ls
```

Que constatez-vous ?

(si les classes ne sont pas trouvées, allez dans /home/asov et tapez « source .bash\_profile » afin de mettre à jour la variable d'environnement CLASSPATH)

Pour accéder au service Sesame il faut maintenant écrire en quelques lignes un client.

Ceux qui le souhaitent peuvent en quelques lignes écrire un client pour ce service (utilisation de l'interface sesame, paramètres m31 et x).

Dans le cas contraire, un fichier Java « Client.java » existe déjà dans le répertoire courant et il suffit de l'éditer.

##### ● Pour tester :

- Ouvrez le projet Sesame
- Exécuter (menu Run, option Run...)

#### Cas d'un client Perl

Pour créer un client à l'aide du langage Perl il faut :

- Télécharger et installer
  - o une version récente de Perl
  - o la librairie SOAP-Lite

Ce travail est déjà réalisé sur les postes de travail du tutorial.

- A l'aide de l'exemple de client Perl de la rubrique « UCD » de la page « XML Web Services Corner », créez un client pour Sesame.

Le fichier Client.pl constitue une solution possible.

- Pour tester, tapez la commande suivante : perl Sesame.pl m31 x

### **Exercice 2 :** Utilisation du service Spectrum du NVO

Se placer dans « ../workspace/Spectra »

Remarque : les Web Services du NVO sont déployés sur des plateformes Microsoft - .Net

Adresse du service :

[http://voservices.net/spectrum/ws\\_v2\\_0/search.asmx](http://voservices.net/spectrum/ws_v2_0/search.asmx)

Abstract du service :

<http://www.voservices.org/webservices.aspx#wave>

- Comment générer le proxy ?

Une fois le proxy généré (n'oubliez pas de l'ajouter au CLASSPATH) vous pouvez éditer le fichier « Spectra.java » qui donne un exemple de client pour le service Spectrum.

- Pour tester, tapez les commandes suivantes :
  - Ouvrez le projet Spectra
  - Exécutez (menu Run, option Run...)

### **Partie 2 :** Le coté Serveur

*Cas d'un déploiement de type jws : simple mais limité*

### **Exercice 3**

- Se placer dans le répertoire « /home/asov/Tools/tomcat/webapps/axis »  
Editez le fichier Addition.java  
Quittez l'édition puis renommez ce fichier en Addition.jws  
Dans Mozilla, tapez l'URL : <http://localhost:8080/axis/Addition.jws?wsdl>

Quelques constatations...

*Cas d'un déploiement de type wsdd* : permet de déployer des services utilisant des bibliothèques, etc...

#### **Exercice 4**

● Allez dans le répertoire `/home/asov/Tools/tomcat/webapps/axis/Addition`

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 (générée automatiquement) de ce qui doit être déployé.

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

● Pour déployer le fichier, tapez la commande « ant deploy »

Dans Mozilla, 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 suivante :

<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...

Remarque :

Addition est un cas très simple, des cas d'utilisation plus évolués peuvent nécessiter l'utilisation d'objets pour le résultat d'une requête plutôt que des types élémentaires.

Dans ce cas, c'est un mécanisme (transparent pour l'utilisateur) de sérialisation- désérialisation des objets (objet vers XML, XML vers objet) qui est mis en œuvre.

Il est également possible de réaliser des attachements (cf. DIME, utilisable avec .Net et Axis) de fichiers (exemple : une image).

### **Partie 3 : 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 client, l'intégration se fait très naturellement, un Web Service n'étant en définitive qu'un « composant » comme les autres. L'utilisation distante de services est parfaitement masquée.

Au niveau du serveur, le déploiement de services Web SOAP est également 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.asp>