TP Web Services

Daniel Hagimont hagimont@enseeiht.fr

L'objectif de ce TP est l'initiation à l'utilisation des Web services (WS). Nous allons utiliser les WS suivant les 2 modes vus en cours : SOAP et REST.

0) Installations

Je suppose que Eclipse et Java sont installés correctement (comme dans les TP précédents).

Pour installer Tomcat avec Resteasy :

cd

mkdir ws

cd ws

wget https://sd-160040.dedibox.fr/hagimont/software/apache-tomcat-ws.tgz

tar xzf apache-tomcat-ws.tgz

editer votre .bashrc et ajouter

export TOMCAT_HOME=$HOME/ws/apache-tomcat-ws

export PATH=$TOMCAT_HOME/bin:$PATH

Vous pouvez lancer Tomcat depuis un terminal ou le lancer dans Eclipse.

Pour lancer ou arrêter Tomcat depuis un terminal :

startup.sh

shutdown.sh

Pour utiliser Resteasy, inclure tous les jar du répertoire lib de Tomcat.

1) Utilisation d'un WS REST + Json

Nous utilisons les outils Resteasy pour implanter les parties clientes comme serveurs.

Un WS Restful a été implanté. Si vous le deployez localement dans votre Tomcat, il est accessible à l'URL : http://localhost:8080/students-server/rest/

Il propose 2 interfaces :

Méthode **getstudent**, paramètres **firstname** et **lastname** , retourne un **Json**.

Exemple d'appel :

http://localhost:8080/students-server/rest/getstudent?firstname=Alain&lastname=Tchana


```
{
        "firstname":"Alain",
        "lastname":"Tchana",
        "birthdate":"18/12/1984",
        "sex":"male",
        "address":"3 rue Jeff Rouchon",
```

```
	"city":"Toulouse",
	"zip":"31000",
	"country":"France",
	"phone":"0102030405",
	"email":"alain.tchana@enseeiht.fr",
	"ine":"1111111111"
}
```


Méthode **getrecord** paramètre **ine,** retourne un **Json**.

Exemple d'appel :

http://localhost:8080/students-server/rest/getrecord?ine=1111111111


```
{
	"mathematics":"12",
	"middleware":"14",
	"networks":"11",
	"systems":"5",
	"architecture":"16",
	"programming":"18",
	"ine":"1111111111"
}
```


Recréer ce projet (un DynamicWebProject nommé students-server) dans eclipse (vous utilisez le runtime Tomcat installé précédemment, c'est un Tomcat 9.0) et importer les sources java + le web.xml dans WEB-INF.

Installez ce WS REST dans votre Tomcat (export WAR => $TOMCAT-HOME/webapps)

Testez le WS directement avec un navigateur (URLs ci-dessus).

Implantez dans un projet Java un programme client qui fait appel à ce service (en utilisant resteasy, il faut inclure dans votre projet les jars de $TOMCAT-HOME/lib). Un exemple est donné dans les transparents du cours. Il vous suffit de :

- décrire l'interface du service avec les annotations @GET, @Path, @Produces … (attention, dans l'URL du service, ne pas inclure "rest" à la fin)

- créer les Java beans associés aux Json utilisés

- créer et utiliser le proxy Resteasy

2) Création d'un WS REST + Json

Créez un WS REST + Json (nommé marks) qui permet de récupérer à partir du nom-prénom d'un étudiant et d'une matière, la note que cet étudiant a obtenu dans la matière. Ce WS est à la fois client (du précédent) et serveur en fournissant une interface :

Méthode **getmark**, paramètres **firstname, lastname et lecture**, retourne un **Integer**.

Exemple d'appel :

http://localhost:8080/marks/rest/getmark?firstname=Alain&lastname=Tchana&lecture=systems

5

Pour implanter ce WS, vous devez créer un Dynamic Web Project. Vous pouvez vous inspirer du projet qui vous a été donné dans la partie 1). Il vous faudra notamment :

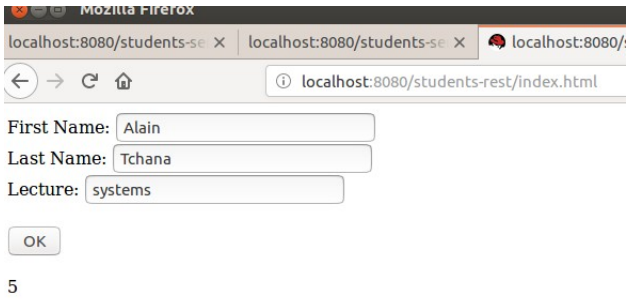
- copier le web.xml présent dans WebContent/WEB-INF

- copier et adapter la classe RespApp qui est référencée depuis web.xml. Cette classe enregistre l'objet appelable avec l'API REST.

- créer la classe implantant votre service et l'annoter.

Vous pouvez installer votre nouveau WS (dans Tomcat) et le tester avec l'URL ci-dessus.

Une autre façon de tester votre WS est d'utiliser une petite page (que nous vous donnons) incluant une séquence de code JavaScript faisant appel à cette API REST + Json. Installez (la copier dans WebContent) et testez avec cette page.



Création et test d'un WS SOAP

Au lieu d'exporter la méthode getmark sous la forme d'une API REST, nous proposons ici de l'exporter sous la forme d'une API SOAP.

Nous repartons d'un projet DynamicWebProject qui inclut une classe et une méthode getMark qui utilise RestEasy (partie cliente) pour faire appel au WS de 1).

Pour créer le WS SOAP associé, sur la classe : click-droit -> Web Services -> Create Web Service (Start Service / No Client)

NB : il faut faire un démarrage (Start Service) du WS dans eclipse pour générer le fichier server-config.wsdd dans WebContent/WEB-INF, sinon ça ne marche pas. Il faut donc arrêter votre Tomcat (shutdown.sh), puis créer le WS dans eclipse (ce qui va démarrer un Tomcat dans eclipse), puis arrêter Tomcat dans eclipse. Vous pouvez ensuite redémarrer votre Tomcat (startup.sh), puis installer le WS SOAP dans votre Tomcat (export WAR).

Ensuite, pour tester votre WS SOAP, sur le fichier WSDL : click droit -> Web Services -> Test with Web Services Explorer