

# PRACTICA WEB SERVICES

## **1 Preparar entorn**

- TOMCAT (Web server)

```
bash
insttomcat
```

- AXIS (Web service framework)

```
wget http://studies.ac.upc.edu/FIB/PXC/lab/bin/axis-1_1.tar.gz
tar xzf axis-1_1.tar.gz
```

- XERCES (packages for parsing and manipulating XML)

```
wget http://studies.ac.upc.edu/FIB/PXC/lab/bin/xerces.jar
mv xerces.jar axis-1_1/webapps/axis/WEB-INF/lib/
cp -R axis-1_1/webapps/axis/ ~/tomcat/webapps/
```

- JAF (JavaBeans Activation Framework)

```
wget http://docencia.ac.upc.edu/FIB/PXC/lab/bin/jaf-1_1-fr.zip
unzip jaf-1_1-fr.zip
cp jaf-1.1/activation.jar ~/tomcat/webapps/axis/WEB-INF/lib/
```

- AXIS.XML

```
mkdir ~/tomcat/conf/Catalina/
mkdir ~/tomcat/conf/Catalina/localhost/
# Crear el axis.xml:
8<-----
<Context path="/axis" docBase="axis" debug="0" reloadable="true" />
----->8
# No fa falta el Context, excepte potser per reloadable
cp axis.xml ~/tomcat/conf/Catalina/localhost/
```

## **2 Arrencar el Tomcat**

- CLASSPATH

```
# Crear el setCLASSPATH.sh:
8<-----
#!/bin/bash
export AXIS_HOME=$(pwd)/axis-1_1
export TOMCAT_HOME=$HOME/tomcat
export CLASSPATH=.:$AXIS_HOME:$AXIS_HOME/lib/axis.jar:$AXIS_HOME/lib/jaxrpc.jar:
$AXIS_HOME/lib/saaj.jar:$AXIS_HOME/lib/commons-logging.jar:$AXIS_HOME/lib/commons-
discovery.jar:$AXIS_HOME/lib/wsdl4j.jar:$TOMCAT_HOME/webapps/axis/WEB-
INF/lib/xerces.jar
----->8
source setCLASSPATH.sh
```

- ARRENCAR EL TOMCAT

```
tomcat start
```

### 3 Provar l'exemple de la calculadora

```
cp axis-1_1/samples/userguide/example2/Calculator.java
~/tomcat/webapps/axis/Calculator.jws
java samples.userguide.example2.CalcClient -p8000 add 2 5
```

### 4 Utilitzar el tcpmon per monitoritzar els missatges dels WebServices

```
java org.apache.axis.utils.tcpmon &
# Posar al tcpmon listen 8080 i target 8000
java samples.userguide.example2.CalcClient -p8080 add 2 5
```

### 5 Provar l'exemple de BotigaWeb

```
# Crearem un fitxer de interfície i un de implementació:
mkdir botigaWeb
# Crear botigaWeb/BotigaWeb.java:
8<-----
package botigaWeb;

public interface BotigaWeb {
    // Mètode que ens dóna el preu d'un producte
    public float veurePreu(String producte);
}
----->8
# Crear botigaWeb/BotigaWebImpl.java:
8<-----
package botigaWeb;

public class BotigaWebImpl{
    float preu = 10.0F;

    public float veurePreu(String producte){
        return preu;
    }
}
----->8
javac botigaWeb/BotigaWeb.java
java org.apache.axis.wsdl.Java2WSDL -o botigaWeb.wsdl -l"http://localhost:
8080/axis/services/botigaWeb" -n urn:botigaWeb -p"botigaWeb" urn:botigaWeb
botigaWeb.BotigaWeb
# Si no tinguéssim el tcpmon escoltant, el port seria 8000 enlloc de 8080
java org.apache.axis.wsdl.WSDL2Java -o . -d Session -s -p botigaWeb.ws
botigaWeb.wsdl

# Editarem el fitxer de SoapBindingImpl per que implementi el nostre servei:
# Editar botigaWeb/ws/BotigaWebSoapBindingImpl.java afegint les línies que
comencen per '+'):
8<-----
package botigaWeb.ws;

+import botigaWeb.BotigaWebImpl;

public class BotigaWebSoapBindingImpl implements botigaWeb.ws.BotigaWeb{
```

```
+   BotigaWebImpl botiga = new BotigaWebImpl();
+   public float veurePreu(java.lang.String in0) throws java.rmi.RemoteException
+   {
+       return botiga.veurePreu(in0);
+   }
}
```

```
----->8
javac -source 1.4 botigaWeb/ws/*.java
jar cvf botigaWeb.jar botigaWeb/*.class botigaWeb/ws/*.class
mv botigaWeb.jar ~/tomcat/webapps/axis/WEB-INF/lib/
# Si no tinguessim el tcpmon escoltant, el port seria 8000 enlloc de 8080
java org.apache.axis.client.AdminClient -p8080 botigaWeb/ws/deploy.wsdd
# Crear un client botigaWeb/BotigaWebTester.java per provar el servei:
8<-----
```

```
package botigaWeb;

public class BotigaWebTester{

    public static void main (String [] args) throws Exception{

        // crear un servei
        botigaWeb.ws.BotigaWebService service = new
            botigaWeb.ws.BotigaWebServiceLocator();

        // usar el servei per a obtenir un stub del servei
        botigaWeb.ws.BotigaWeb botiga = service.getbotigaWeb();

        // cridar el servei
        System.out.println("Preu: " + botiga.veurePreu("aa"));
    }
}
```

```
----->8
javac botigaWeb/BotigaWebTester.java
java botigaWeb.BotigaWebTester
```

**Important:** Si es vol repetir el punt 5 canviant de port, feu una copia de seguretat del fitxer BotigaWebSoapBindingImpl.java a un altre directori i el borreu abans de fer el WSDL2Java. Un cop fet, el torneu a restaurar.

## ***6 Implementar un Webservice que faci lloguer i llistar\_lloguers.***

Un cop fets tots aquests preliminars, ja estem preparats per fer la pràctica: Implementar un Webservice que faci lloguer i llistar\_lloguers.