|  |
| --- |
| Software Engineering of Distributed Systems, KTH |
| Programming Web Services Homework 2 |
| Lab of AXIS, and Web Services |

|  |
| --- |
| Shanbo Li and Sike Huang  2008-2-3 |

## 1. A short description of your activities for installation of the tools (what problems did you face)

**Answer:**

### Installation of Java EE SDK

step 1: go to <http://java.sun.com/javaee/downloads/index.jsp>

step 2: click “Download with JDK”

step 3: select Windows as Platform, English as Language, confirm the license and continue

step 4: click java\_ee\_sdk-5\_04-windows.exe and run

step5: follow the instruction to install Java EE SDK

### 1.2 Installation of Jakarta Tomcat 6

step 1: go to <http://tomcat.apache.org/download-60.cgi>

step 2: click “[Windows Service Installer](http://apache.dataphone.se/tomcat/tomcat-6/v6.0.14/bin/apache-tomcat-6.0.14.exe)” and run the installer

step 3: follow the instruction to install Tomcat 6

**Note:** when it ask to set the JRE path, we set the JDK path. It is very important to make Tomcat run well.

### 1.3 Installation of AXIS 1.4

step 1: go to <http://apache.dataphone.se/ws/axis/1_4/>

step 2: download “[axis-bin-1\_4.zip](http://apache.dataphone.se/ws/axis/1_4/axis-bin-1_4.zip)” and unzip to local disk

step 3: copy %AXIS\_HOME%/webapps/axis to %TOMECAT\_HOME%/webapps

step 4: copy %AXIS\_HOME%/lib/\* to my project lib

## 2. A protocol of compilation of the WSDL service to Java

**Answer:**

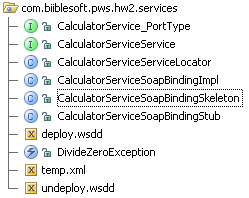
use the follow command to generate java files

|  |
| --- |
| % java org.apache.axis.wsdl.WSDL2Java --server-side --skeletonDeploy true –p com.biiblesoft.com.pws.hw2.services CalculatorService.wsdl |

in which the ***–server-side*** switch indicate that we need a service side Web Service.

***--skeletonDeploy*** true switch will generate a skeleton

The follow files are automatically generated:



Edit ***CalculatorServiceSoapBindingImpl*** and implement the calculator service.

Use ***deploy.wsdd*** to deploy the service.

## 3. Completed code for the implementation of the service

**Answer:**

The core business logic are the following codes:

|  |
| --- |
| public int add(int addend1, int addend2) throws RemoteException {  return addend1 + addend2;  }  public int division(int dividend, int divisor) throws RemoteException, DivideZeroException {  return dividend / divisor;  }  public int multiplication(int multiplicand, int multiplier) throws RemoteException {  return multiplicand \* multiplier;  }  public int subtraction(int minuend, int subtrahend) throws RemoteException {  return minuend - subtrahend;  } |

and the entire source with Intellij Project is in ***PWS\_HW2\_project.rar***

## 4. Protocol of deployment of the service in the axis

**Answer:**

**step 1:**

Package all of the code that we have compiled and copy it into Axis' lib path:

|  |
| --- |
| % cp CalculatorService.jar %TOMCAT\_HOME%/webapps/axis/WEB-INF/lib |

**step 2:**

copy following codes from ***deploy.wsdd*** to ***server-config.wsdd*** which located in

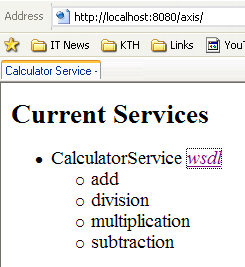
%TOMCAT\_HOME%/webapps/axis/WEB-INF/

|  |
| --- |
| <!-- Services from CalculatorServiceService WSDL service -->  <service name="CalculatorService" provider="java:RPC" style="rpc" use="encoded">  <parameter name="wsdlTargetNamespace" value="http://localhost:8080/axis/services/CalculatorService"/>  <parameter name="wsdlServiceElement" value="CalculatorServiceService"/>  <parameter name="wsdlServicePort" value="CalculatorService"/>  <parameter name="className" value="com.biiblesoft.pws.hw2.services.CalculatorServiceSoapBindingSkeleton"/>  <parameter name="wsdlPortType" value="CalculatorService"/>  <parameter name="typeMappingVersion" value="1.2"/>  <parameter name="allowedMethods" value="\*"/>  </service> |

**step 3:**

start Apache Tomcat

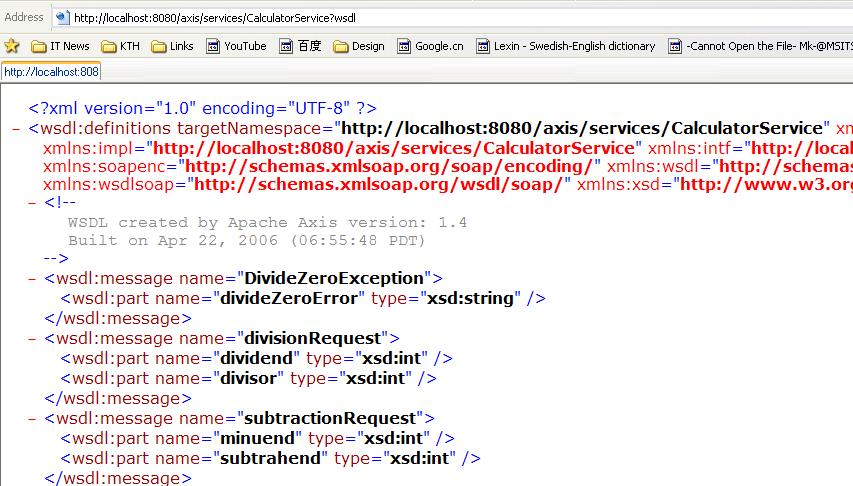
Go to <http://localhost:8080/axis> and see the calculator service there as shown in the following picture:



**step 4:**

Click [*wsdl*](http://localhost:8080/axis/services/CalculatorService?wsdl) and it will link to <http://localhost:8080/axis/services/CalculatorService?wsdl>

that is the location of wsdl of the CalculatorService.



## 4. Texts of incoming and outgoing SOAP messages for invoking "add" operation

**Answer:**

Text of incoming SOAP messages for invoking “add” operation:

|  |
| --- |
| <soapenv:Envelope  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:def="http://DefaultNamespace">  <soapenv:Header/>  <soapenv:Body>  <def:add soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">  <addend1 xsi:type="xsd:int">1124</addend1>  <addend2 xsi:type="xsd:int">810</addend2>  </def:add>  </soapenv:Body>  </soapenv:Envelope> |

Text of incoming SOAP messages for invoking “add” operation:

|  |
| --- |
| <soapenv:Envelope  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  <soapenv:Body>  <ns1:addResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:ns1="http://DefaultNamespace">  <addReturn xsi:type="xsd:int">1934</addReturn>  </ns1:addResponse>  </soapenv:Body>  </soapenv:Envelope> |

## 5. Source code of Java program for invocation of the Calculator service according to required functionality in the Exercise

**Answer:**

please check ***PWS\_HW2\_ServiceTest.rar***

screenshot:

