2G1523
Programming Web Services
Homework 3

Course leader:
Professor Mihhail Matskin
misha@imit.kth.se

Teaching assistant:
Hamid Reza Mizani
hrmizani@imit.kth.se

KTH/ICT/ECS , VT07

Homework #3 - Deadline

- Start date: 2007-02-08
- Due date: 2007-02-18
- Deliverable: Report (See course web)

Homework #3 - Aims

- Aim: To learn the followings:
  - The data structure of UDDI
  - Publish and find UDDI business information using Java
  - Setup a UDDI registry

Homework #3 – UDDI2J

- http://uddi4j.sourceforge.net/
- Download it (Version 2.0.5).
- Add ***uddi4j.jar*** to your class path.
- A Java class library that provides an API to interact with a UDDI registry.
- Enables businesses to:
  - Discover each other.
  - Define how they interact over the internet and share information in a global registry architecture.

Homework #3 – SOAP4J

- Apache SOAP
- http://ws.apache.org/soap/
- Download it (Version 2.3.1).
- Add ***soap.jar*** to your class path.
- An implementation of the SOAP submission to W3C
- Based on the IBM SOAP4J implementation

Homework #3 - JAVAMAIL

- http://java.sun.com/products/javamail
- Download it (Version 1.4).
- Add ***mail.jar*** to your class path.
- Provides a platform-independent and protocol-independent framework to build mail and messaging applications

Homework #3 – JavaBeans Activation Framework

- http://java.sun.com/products/javabeans/jaf/downloads/index.html
- Download it (Version 1.1).
- Add ***activation.jar*** to your class path.
- To determine the type of an arbitrary piece of data, encapsulate access to it, discover the operations available on it, and to instantiate the appropriate bean to perform said operation(s)
- Example: a JPEG image in a browser

Homework #3 – Exercise 1

- Read:
  - **FindBusinessEntity.java**
  - **PublishBusinessEntity.java**
- Textbook: pp327 -- pp334
- http://www.imit.kth.se/courses/2g1523/Homeworks/FindBusinessEntity.java/
- http://www.imit.kth.se/courses/2g1523/Homeworks/PublishBusinessEntity.java/

Homework #3 – Exercise 1
FindBusinessEntity.java

```
String inquiryURL =  "http://uddi.ibm.com/testregistry/inquiryapi";
UDDIProxy uddiProxy = new UDDIProxy();
uddiProxy.setInquiryURL(inquiryURL);
IdentifierBag identifierBag = new IdentifierBag();
Vector keyedReferenceList = new Vector();
keyedReferenceList.add(new KeyedReference("DUNS", "00-111-1111",
"uuid:8609C81E-EE1F-4D5A-B202-3EB13AD01823"));
identifierBag.setKeyedReferenceVector(keyedReferenceList);
CategoryBag categoryBag = new CategoryBag();
keyedReferenceList = new Vector();
keyedReferenceList.add(new KeyedReference(
"Sporting and Athletic Goods Manufacturing", "33992",
"uuid:C0B9FE13-179F-413D-8A5B-5004DB8E5BB2"));
keyedReferenceList.add(new KeyedReference("New York", "US-NY",
"uuid:4E49A8D6-D5A2-4FC2-93A0-0411D8D19E88"));
categoryBag.setKeyedReferenceVector(keyedReferenceList);
```

Homework #3 – Exercise 1
FindBusinessEntity.java

```
Vector names = new Vector();
names.add(new Name("JinghaiSkatesTown"));
BusinessList businessList = uddiProxy.find_business( (Vector) names,
(DiscoveryURLs) null, null, null, (TModelBag) null, (FindQualifiers) null, 10);
BusinessInfos businessInfos = businessList.getBusinessInfos();
for(int i=0; i < businessInfos.size(); i++) {
   String businessKey = ((BusinessInfo) businessInfos.get(i)).getBusinessKey();
   System.out.println("Key for businessEntity " + i + " found: " + businessKey + ".");
   BusinessDetail businessDetail = uddiProxy.get_businessDetail(businessKey);
   Name businessName = (Name) ((BusinessEntity)
   businessDetail.getBusinessEntityVector().elementAt(0)).getNameVector().get(0);
   System.out.println("Name of businessEntity " + i + " found: " +
   businessName.getText() + ".");
}
```

Homework #3 – Exercise 1
PublishBusinessEntity.java

```
String publishURL = "https://uddi.ibm.com/testregistry/publishapi";
UDDIProxy uddiProxy = null;
System.setProperty("java.protocol.handler.pkgs",
"com.ibm.net.ssl.internal.www.protocol");
uddiProxy = new UDDIProxy();
uddiProxy.setPublishURL(publishURL);
BusinessEntity businessEntity = new BusinessEntity();
businessEntity.setBusinessKey("");
businessEntity.setDefaultNameString("JinghaiSkatesTown", "en");
Vector description = new Vector();
description.add(new Description("UDDI businessEntity for
    SkatesTown.", "en"));
businessEntity.setDescriptionVector(description);
```

Homework #3 – Exercise 1
PublishBusinessEntity.java

```
Contact ctoContact = new Contact("Dean Carroll");
ctoContact.setUseType("Technical Information");
ctoContact.setDefaultDescriptionString("CTO for technical information");
Vector phoneList = new Vector();
Phone mainPhone = new Phone("1.212.555.0001");
mainPhone.setUseType("Main Office");
phoneList.add(mainPhone);
ctoContact.setPhoneVector(phoneList);
Vector emailList = new Vector();
Email email = new Email("dean.carroll@SkatesTown.com");
email.setUseType("CTO");
emailList.add(email);
email = new Email("info@SkatesTown.com");
email.setUseType("General Information");
emailList.add(email);
```

Homework #3 – Exercise 1
PublishBusinessEntity.java

```
ctoContact.setEmailVector(emailList);
Vector skatesTownAddress = new Vector();
Address address = new Address();
address.setSortCode("10001");
address.setUseType("Main Office");
Vector addressLineList = new Vector();
addressLineList.add("2001 Skate Services Lane");
addressLineList.add("New York, NY 10001");
addressLineList.add("USA");
address.setAddressLineStrings(addressLineList);
skatesTownAddress.add(address);
ctoContact.setAddressVector(skatesTownAddress);
```

Homework #3 – Exercise 1 PublishBusinessEntity.java

```
Contact salesContact = new Contact("Sandy Smith");
salesContact.setUseType("Sales Information");
salesContact.setDefaultDescriptionString("VP Sales");
phoneList = new Vector();
phoneList.add(mainPhone);
Phone mobilePhone = new Phone("1.212.555.8888");
mobilePhone.setUseType("Mobile");
phoneList.add(mobilePhone);
salesContact.setPhoneVector(phoneList);
emailList = new Vector();
email = new Email("sandy.smith@SkatesTown.com");
email.setUseType("VP Sales");
emailList.add(email);
email = new Email("sales@SkatesTown.com");
email.setUseType("Sales Information");
emailList.add(email);
salesContact.setEmailVector(emailList);
salesContact.setAddressVector(skatesTownAddress);
```

Homework #3 – Exercise 1
PublishBusinessEntity.java

```
Contacts contacts = new Contacts();
contacts.add(ctoContact);
contacts.add(salesContact);
businessEntity.setContacts(contacts);
AuthToken authToken = uddiProxy.get_authToken(userid, password);
Vector businessEntityList = new Vector();
businessEntityList.add(businessEntity);
BusinessDetail businessDetail =
uddiProxy.save_business(authToken.getAuthInfoString(),
    businessEntityList);
String businessKey = ((BusinessEntity)
    businessDetail.getBusinessEntityVector().elementAt(0)).getBusinessKe
    y();
System.out.println("Published businessEntity key: " + businessKey + ".");
```

Homework #3 – Exercise 2

- Setup a private UDDI registry

- "http://www.devx.com/Java/Article/21390/"

Homework #3 – Exercise 2

- Step 1 of 5: Install these tools:
  - Java 2 SDK (http://java.sun.com/j2se/index.jsp)
  - J2EE (http://java.sun.com/javaee/index.jsp)
  - Web server (http://tomcat.apache.org/index.html)
  - SOAP (http://ws.apache.org/axis/)
  - Database (http://www.mysql.com/)
  - UDDI (http://ws.apache.org/juddi/)

Homework #3 – Exercise 2

- Step 2 of 5: Setup the web application:
  - Copy the UDDI web application directory to your application server directory.
  - "C:\juddi-0.9rc4\webapp\juddi" → "C:\Program Files\Apache Software Foundation\Tomcat5.5\webapps\juddi"

Homework #3 – Exercise 2

- Step 3 of 5: Setup the database:
  - After installing, create UDDI databases and configure the parameters.
  - There are SQL scripts which do it automatically.
  - “C:\juddi-0.9rc4\sql\”.
  - Choose the right script for your database system.
  - MySQL: “C:\juddi-0.9rc4\sql\mysql\create_database.sql”

Homework #3 – Exercise 2

- Step 4 of 5: Create publisher information:
  - In order to be able to publish, the publisher needs a username and password in the database.
  - To insert this information there is another SQL script.
  - MySQL: "C:\juddi-0.9rc4\sql\mysql\insert_publishers.sql"
  - Modify this script, enter your information and then run it to insert publisher information into the database.

Homework #3 – Exercise 2

- Step 5 of 5: Start the web server:
  - After starting the web server (or restarting it if it was running before copying UDDI web application folder) the registry is ready to use.
  - You can test it by opening the address “http://<host>:<port>/juddi” in a browser.
  - If you use default settings for Tomcat this address will be: “http://localhost:8080/juddi”.

Homework #3 – Exercise 3

- Modify **PublishBusinessEntity.java** to Publish the service you built in homework 2.
- Compile **PublishBusinessEntity.java**
- Execute: **java PublishBusinessEntity username password**
- Delete the 30th line:
  - **java.security.Security.addProvider(new com.ibm.jsse.JSSEProvider());**
  - Otherwise you will get error!

Homework #3 – Exercise 4

- Modify **FindBusinessEntity.java** to find the service you published.
- Compile **FindBusinessEntity.java**
- Execute **FindBusinessEntity**
- Print out the result.

Homework #3 – Exercise 5

- Based on **FindBusinessEntity.java**, write a new program to locate all business entities that have name "SkatesTown“.

Homework #3 – Report

- A short description of your activities for installation of the tools (what problems did you face).
- A short description of your activities for installation the UDDI registry.
- A protocol of compilation and execution of **FindBusinessEntity.java.**
- Source code for finding all business entities that have name "SkatesTown”.
- Results of running **PublishBusinessEntity.java** and publishing you CalculatorService from homework 2 in the UDDI registry.

Homework #3 - Delivery

- Send your report by e-mail to both misha@imit.kth.se and hrmizani@imit.kth.se.
- Deadline: 2007-02-18
- See course web for more information.

GOOD LUCK!