

# Accessing MySQL from Java

---

## Contents

1. Introduction .....	2
2. Prerequisites .....	2
3. Setup .....	2
3.1. Get Apache Axis2 .....	2
3.2. Populate MySQL Database.....	2
3.3. Set up ODBC Access .....	3
4. Discovery.....	4
4.1. Web Service Creation using SOA Gateway .....	4
4.2. Accessing the WSDL .....	6
5. Accessing Web Service with Java .....	8
5.1. Initial Setup .....	8
5.2. Building the generated Code .....	9
6. Writing the Code.....	10
6.1. Simple Java Program .....	10
6.2. Example using Java Swing .....	12
6.3. Building the Code.....	20
6.4. Running the code .....	20
7. Conclusion.....	23
8. Appendix .....	23
SimpleTest.java .....	23
RisarisBankDemo.java.....	23

## 1. Introduction

In this tutorial we will show you how to build a Java application to access MySQL via the SOA Gateway.

## 2. Prerequisites

It is assumed that you are running the 3 components, MySQL, Java and the SOA Gateway on Windows.

It is assumed you already have a SOA Gateway server and Control Centre installed. See [here](#) for more info about installing the SOA Gateway.

## 3. Setup

To build and run Java application, you will need a Java-compatible IDE. There are many available to download, such as NetBeans, JCreator, and IntelliJ. For the purposes of this tutorial, we are going to use Eclipse. The main reason for this is that you are already using Eclipse to run the SOA Gateway control centre, all you need to do is open the Java perspective, by clicking Window, Open Perspective, Java.

### 3.1. Get Apache Axis2

Apache Axis2 is an open source web service framework. It can be used to generate Java classes from a WSDL file. We can then use these classes to invoke our SOA Gateway web service.

Download Axis2 from [here](#). You should choose the “Standard Binary Distribution”. Save this file to a well known location, and extract. For example save to C:\Axis2\ and extract in this folder.

You will also need a MySQL database. Again, the Open Source version (known as the *MySQL Community Server*) can be freely downloaded from the MySQL website. See [this link](#) for download, and [here](#) to step you through the installation and configuration.

### 3.2. Populate MySQL Database

Now that you’ve got MySQL installed and configured, you will need to populate it with some demo data. For this we use the RisarisBank sample. This is available [here](#).

Save this file to “C:\Temp\RisarisBank.sql”

- Connect to the MySQL Server using the **mysql** command.

E.g `shell> mysql -u root -p`

This command connects to the server using the MySQL `root` account to make sure that you'll have permission to create the `RisarisBank` database. The `-p` option tells **mysql** to prompt you for the `root` password. Enter the password when prompted. (Remember

that the MySQL `root` account is not the same as the operating system `root` account and probably will have a different password.)

- Create the `RisarisBank` database.

```
mysql> CREATE DATABASE RisarisBank;
```

```
mysql> use RisarisBank;
```

- Load the contents of `RisarisBank.sql` into the `RisarisBank` database. E.g.

```
mysql> SOURCE c:\Temp\RisarisBank.sql
```

- After the `SOURCE` command finishes, you can view your new tables.

```
mysql> SHOW TABLES;
```

```
mysql> DESCRIBE CustomerInformation;
```

```
mysql> DESCRIBE Branch;
```

*etc ...*

### 3.3.Set up ODBC Access

The final thing to do with your MySQL Database is to set up an ODBC DSN which will be used by the SOA Gateway to access this database.

Click Start, Control Panel, Administrative Tools, Data Sources (ODBC)

From the resulting screen, choose the "System DSN" Tab.

Click Add

From the list of data source drivers, select "MySQL ODBC 3.51 Driver".

*If you do not see this driver in the list, you need to install the MySQL Connector. See [here](#) for more information. We recommend installing v3.51.*

Click Finish, and a window will appear allowing you to enter the DSN information. Add the following:

Data Source Name: `RisarisBank`

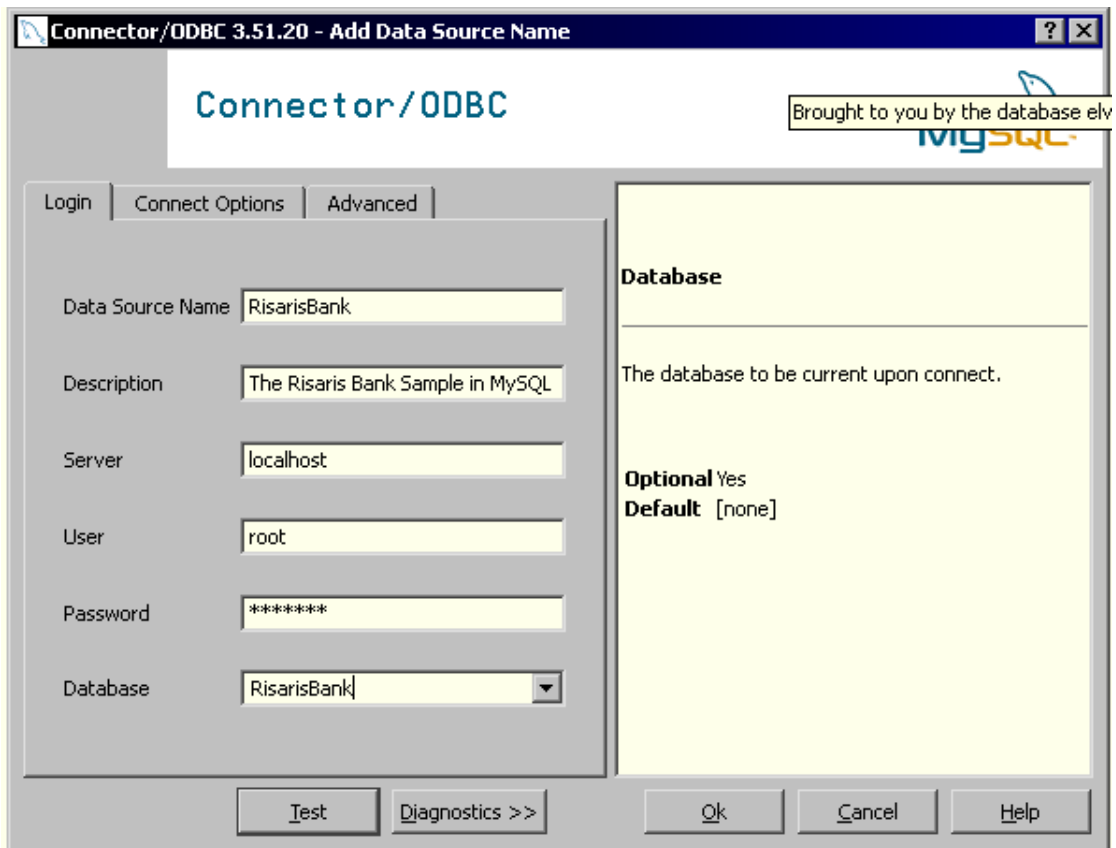
Description: `The Risaris Bank Sample in MySQL`

Server: `localhost`

User: `root`

Password: `*** your MySQL root password ***`

Database: `RisarisBank (select from the drop down list)`



All other options can be left as-is. Click OK.

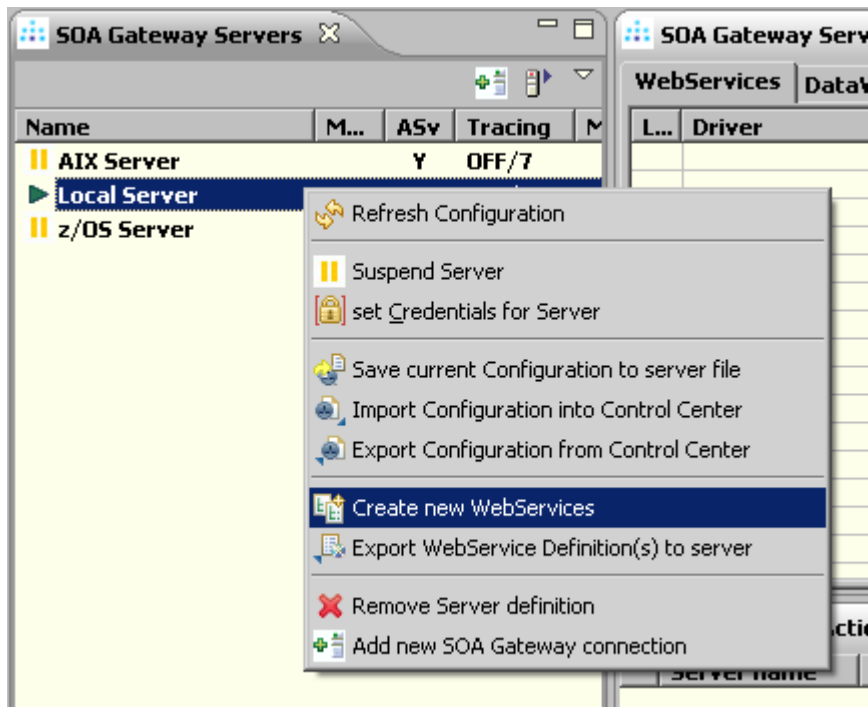
## 4. Discovery

At this stage you've got a Java IDE, and a MySQL database with some sample data in it. In this section we'll show you how to create web services from each of the MySQL tables. These web services can be used by the Java language (and many others) to give you direct real-time access to your MySQL Data.

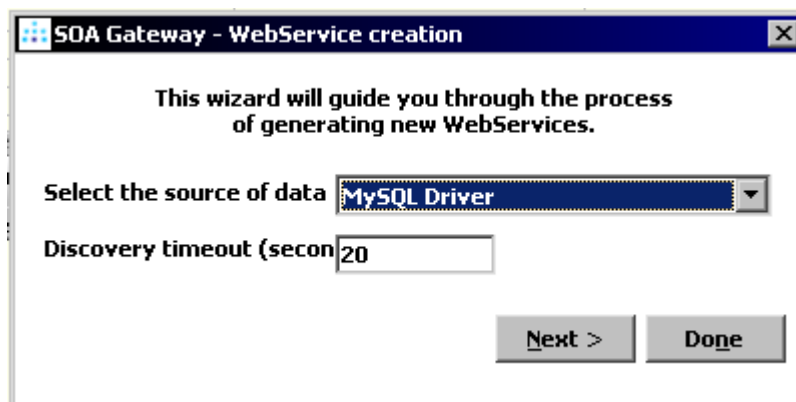
### 4.1. Web Service Creation using SOA Gateway

Start your SOA Gateway Control Centre. See [here](#) for an introduction to the Control Centre.

In your servers view, right click the entry which represents your local SOA Gateway Server. Select "Create New Web Services".

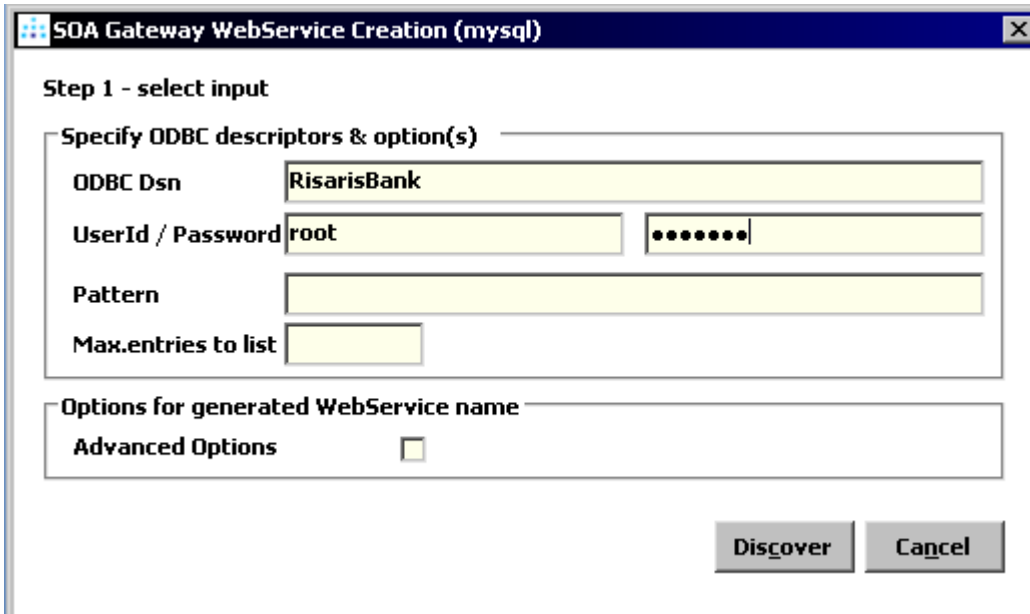


From the next dialog, choose “MySQL Driver”. If you do not see have a MySQL Driver in the list, see how to create one [here](#).



Click Next.

The next screen gives you the ability to add information about your DSN



Enter the above information and click Discover.

The wizard will display all the tables it finds at this (RisarisBank) DSN.

Click "Select All", and click "Import".

The wizard will create web services from each one of these tables.

The screenshot shows the "SOA Gateway Servers" configuration window. The "WebServices" tab is active, displaying a table with columns: Name, M..., ASv, Mod, Driver, WebService, DataSource Id, and DataView. The table lists 8 web services, each created from a MySQL driver and linked to a specific table in the RisarisBank DSN.

Name	M...	ASv	Mod	Driver	WebService	DataSource Id	DataView
AIX Server	Y		MySQL	MySQL Driver	accountsmovements	odbcDsn=RisarisBank, tableName=accountsmovements	accountsmovements
DMZ	Y		MySQL	MySQL Driver	audit	odbcDsn=RisarisBank, tableName=audit	audit
dublin dev	Y		MySQL	MySQL Driver	branch	odbcDsn=RisarisBank, tableName=branch	branch
jk server	Y		MySQL	MySQL Driver	currentaccount	odbcDsn=RisarisBank, tableName=currentaccount	currentaccount
jom server	Y		MySQL	MySQL Driver	customeraccountxref	odbcDsn=RisarisBank, tableName=customeraccountxref	customeraccountxref
lxbre server	Y		MySQL	MySQL Driver	customerinformation	odbcDsn=RisarisBank, tableName=customerinformation	customerinformation
PCRJW9	Y		MySQL	MySQL Driver	depositaccount	odbcDsn=RisarisBank, tableName=depositaccount	depositaccount
risaris.com server	Y		MySQL	MySQL Driver	tellertable	odbcDsn=RisarisBank, tableName=tellertable	tellertable
vse	Y						
z/OS Server	Y						
z/vse	Y						

You've just created 8 Web Services from your 8 MySQL Tables!

## 4.2. Accessing the WSDL

Web Service Description Language (WSDL) is a standard, XML-based language that is used to describe a Web Service.

For each of the 8 web services you've created in the previous section, the SOA Gateway provides you with a WSDL to describe the Web Service. The WSDL itself is usually interpreted by a web service client, such as Java, but it is useful to know where to find the WSDL for each of your Web Services.

As WSDL is XML-based, it will open in your browser of choice. To see the WSDL for one of your Risar Bank web services, do the following in your SOA Gateway Control Centre:

- Click on the web service you are interested in, for example the “branch” web service.
- The properties for this web service should appear in your [Properties View](#). If you do not see the Properties view, select Window -> Show View -> Other -> General -> Properties and click OK.
- In the properties view, there is a link to your WSDL. Click it to open the WSDL in a browser.

The screenshot displays the SOA Gateway Server Configuration interface. The top window, titled "SOA Gateway Server Configuration - Local Server", shows a table of web services. The "branch" service is highlighted. Below this, the "SOA Gateway Action Log" shows messages such as "ODBC discovery completed, 8 WebService(s) generated" and "Configuration autosaved due to published WebService modification(s)". The bottom window, titled "Properties", shows the "WebService properties" for the "branch" service. A green arrow points to the "WSDL URL" field, which contains the link <http://localhost:56000/branch?WSDL>. Other fields include "Name", "DataView", "Driver" (MySQL), "Read-only" (unchecked), "SBCS-Codepage", "MBCS-Codepage", "odbcDsn" (RisarisBank), "schemaName", and "tableName" (branch).

Mod	Driver	WebService	DataSource Id
MySQL	MySQL Driver	accountsmovements	odbcDsn=RisariskBank, tableN
MySQL	MySQL Driver	audit	odbcDsn=RisariskBank, tableN
MySQL	MySQL Driver	branch	odbcDsn=RisariskBank, tableN
MySQL	MySQL Driver	currentaccount	odbcDsn=RisariskBank, tableN
MySQL	MySQL Driver	customeraccountxref	odbcDsn=RisariskBank, tableN
MySQL	MySQL Driver	customerinformation	odbcDsn=RisariskBank, tableN
MySQL	MySQL Driver	depositaccount	odbcDsn=RisariskBank, tableN
MySQL	MySQL Driver	tellertable	odbcDsn=RisariskBank, tableN

SOA Gateway Action Log

Server name	Message
Local Server	ODBC discovery completed, 8 WebService(s) generated
Local Server	Configuration autosaved due to published WebService modification(s)

Properties

WebService properties

Resource

Name

DataView

Driver: MySQL

Read-only:

WSDL URL is: <http://localhost:56000/branch?WSDL>

WebService Identification and options

odbcDsn: RisarisBank

schemaName

tableName: branch

You can view the WSDL for the other web services by clicking the link from their properties view.

This WSDL is the starting point for using Web Services, and can be used time and again by different web service clients.

## 5. Accessing Web Service with Java

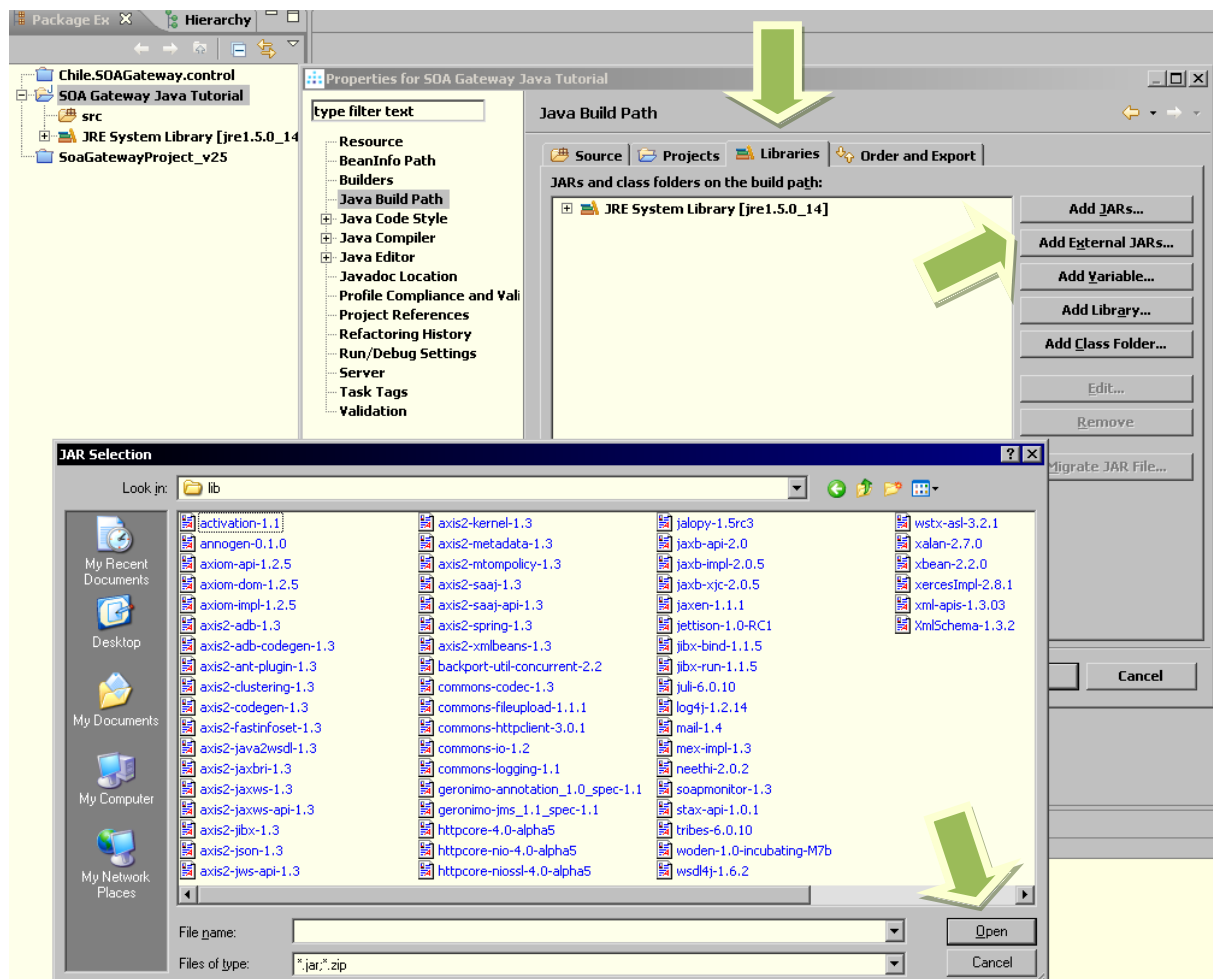
Java is an object-orientated programming language developed by Sun Microsystems. Its syntax is typically based around C++, but has fewer low-level APIs. Java programs are usually compiled into byte-code which can be run on any machines which run a Java Virtual Machine.

### 5.1. Initial Setup

In your Eclipse IDE, open a Java perspective. Select Window, Open Perspective, Java.

Create a new java project, by selecting File, New, Java Project. Name the project "SOA Gateway Java Tutorial". Click Finish.

Click File, Properties to view your project properties. Select "Java Build Path", the "Libraries" tab, and click "Add External Jars". From the pop-up that appears, traverse the "lib" folder in the Axis2 distribution you downloaded earlier. Select **all** these jar files, and click "Open".



Click OK.

Open a DOS box, and change to the Axis2 bin directory. For example

```
C:\documents and settings\brian> cd \
```

```
C:\>cd Axis2\axis2-1.3\bin
```

We want to use 2 of the Web Services we've created, the customerinformation and the currentaccount web services.

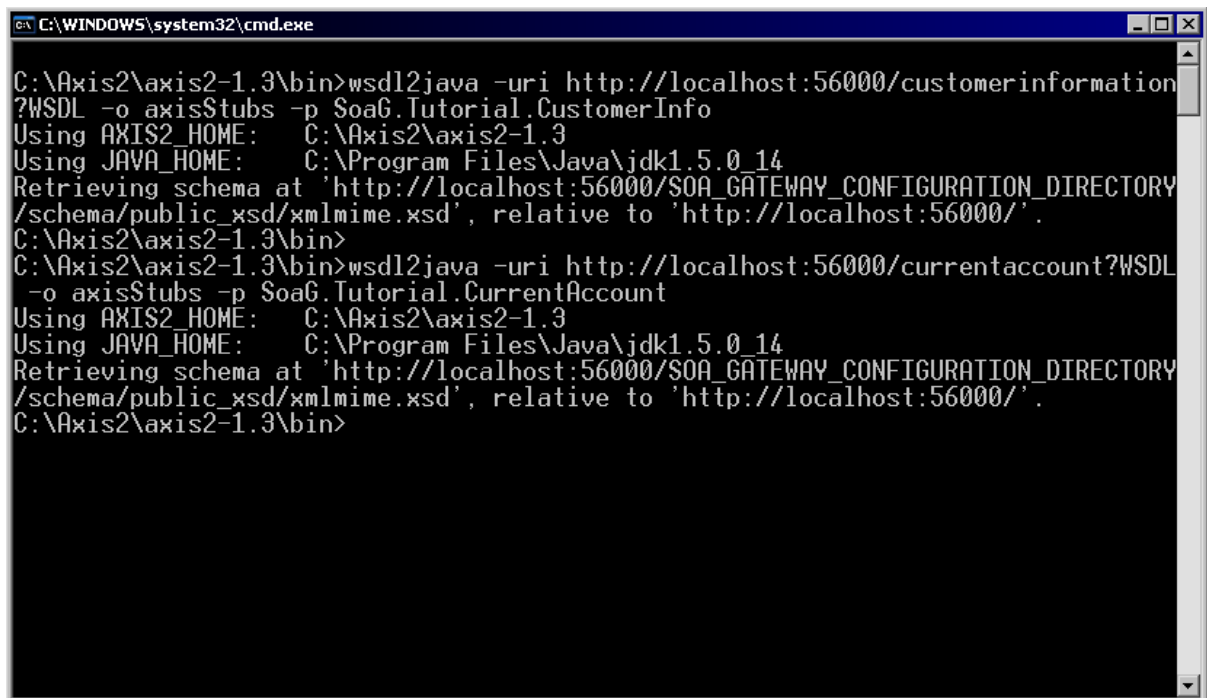
Using the "wsdl2Java" program, Axis2 will generate Java code from the WSDL which we can use to call the SOA Gateway web services.

Run the following commands

```
wsdl2java -uri http://localhost:56000/customerinformation?WSDL -o axisStubs -p  
SoaG.Tutorial.CustomerInfo
```

```
wsdl2java -uri http://localhost:56000/currentaccount?WSDL -o axisStubs -p  
SoaG.Tutorial.CurrentAccount
```

For example:



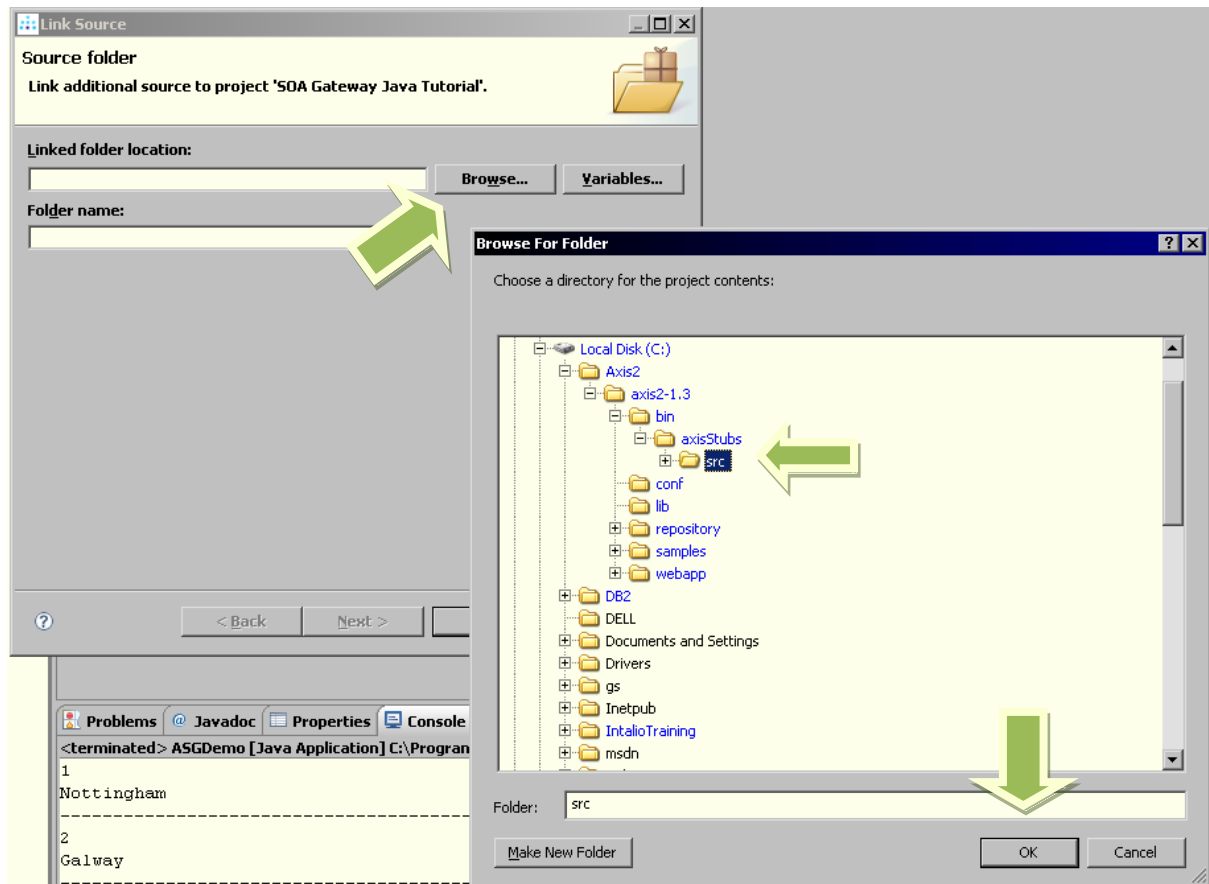
```
C:\WINDOWS\system32\cmd.exe
C:\Axis2\axis2-1.3\bin>wsdl2java -uri http://localhost:56000/customerinformation?WSDL -o axisStubs -p SoaG.Tutorial.CustomerInfo
Using AXIS2_HOME: C:\Axis2\axis2-1.3
Using JAVA_HOME: C:\Program Files\Java\jdk1.5.0_14
Retrieving schema at 'http://localhost:56000/soa_gateway_configuration_directory/schema/public_xsd/xmlmime.xsd', relative to 'http://localhost:56000/'.
C:\Axis2\axis2-1.3\bin>
C:\Axis2\axis2-1.3\bin>wsdl2java -uri http://localhost:56000/currentaccount?WSDL -o axisStubs -p SoaG.Tutorial.CurrentAccount
Using AXIS2_HOME: C:\Axis2\axis2-1.3
Using JAVA_HOME: C:\Program Files\Java\jdk1.5.0_14
Retrieving schema at 'http://localhost:56000/soa_gateway_configuration_directory/schema/public_xsd/xmlmime.xsd', relative to 'http://localhost:56000/'.
C:\Axis2\axis2-1.3\bin>
```

## 5.2. Building the generated Code

We will use the Eclipse IDE to build the generated code for us. To do this we need to add the "axisStubs" folder to our "SOA Gateway Java Tutorial" Project.

To do this, right click the "SOA Gateway Java Tutorial" project. Select "Build Path" and "Link Source". Browse to the location of the "axisStubs". Select the "src" folder and click OK.

For example:



After clicking OK, in the “Folder Name” field, enter AxisStubs. Click Finish.

## 6. Writing the Code

### 6.1. Simple Java Program

In this section we’ll create a simple java program that calls one of our SOA Gateway web services. This is intended to give you a brief introduction on how to call 1 web service once.

Under the SOA Gateway Java Tutorial project, right click the “src” folder. Select “New”, “File”, and name your file “SimpleTest.java”.

The first thing we want to do is import the generated Axis stubs. Add the following to SimpleTest.java

```
import SoaG.Tutorial.CustomerInfo.*;
```

```
import SoaG.Tutorial.CustomerInfo.CustomerInformationRootServiceStub.*;
```

Hit Ctrl+S to save the file. The IDE will automatically build the file, and display errors and warnings in the “Problems” view.

Now add the class definition and the main method.

```
public class SimpleTest {
```

```

public static void main(String[] args) {

}
}

```

Inside the main method, create the security credentials required to access the server. This should be enclosed within a try/catch block.

N.B. Note the user name of `root` and password of `letmein` below. Change to those of your MySQL setup.

```

try{
    Security sec = new
        CustomerinformationRootServiceStub.Security();

    CustomerinformationRootServiceStub.UsernameToken_type0 token = new
        CustomerinformationRootServiceStub.UsernameToken_type0();
    token.setUsername("root");
    token.setPassword("letmein");
    sec.setUsernameToken(token);
}
catch(Exception e){
    e.printStackTrace();
}

```

Within the try block, create a new stub object which will be used to access the web service.

```

CustomerinformationRootServiceStub stub =

    new CustomerinformationRootServiceStub();

```

Next setup a key to hold the query information for the web service. In this case, we are going to ask the SOA Gateway web service to list all records in the Customer Information table. This is indicated by specifying CustomerNumber as `"*"`.

```

CustomerinformationGroupKeyType key =
    new CustomerinformationGroupKeyType();

key.setCustomerNumber("*");

```

We put this key information into a listKey object, which is the type of variable our web service expects.

```

CustomerinformationGroupListElement listKey =
    new CustomerinformationGroupListElement();

listKey.setCustomerinformationGroupListElement(key);

```

Next we setup a variable to hold the results of the web service query.

```

CustomerinformationRootElement results = null;

```

Call the Web Service!

```

results = stub.list(listKey, sec, null);

```

Finally we will process the output, and print the results

```

CustomerinformationRootType root =
results.getCustomerinformationRootElement().getCustomerinformationRoot();

for( int i = 0;
    i != root.getCustomerinformationGroup().length ;
    i++){

CustomerinformationGroupType[] customer =
root.getCustomerinformationGroup();
System.out.print(customer[i].getCustomerNumber() + " ");
System.out.print(customer[i].getFirstName() + " ");
System.out.print(customer[i].getSurname() + " ");
System.out.print(customer[i].getAddressLine1() + " ");
System.out.print(customer[i].getAddressLine2() + " ");
System.out.print(customer[i].getCity() + " ");
System.out.print(customer[i].getPostcode() + " ");
System.out.println(customer[i].getDateOfBirth() + " ");
}

```

The full source code is available in the appendix at the end of this document.

The final thing to do is to run your program. Again this can be done from within the Eclipse IDE. Firstly, right-click the “SimpleTest.java” file. Then select “Run As”, “Java Application”.

The results of the program will appear in the “Console” view.

E.g.

```

<terminated> SimpleTest (1) [Java Application] C:\Program Files\Java\jre1.5.0_14\bin\javaw.exe (2 Apr 2008 12:32:52)
45 Mary Cohen 199 Letsby Avenue Alvaston Preston PR9 3HB 14/02/1979
46 Frederick Rubin 13 Hillcott Road Alvaston Gloucester GL8 4KA 10/07/1988
47 Marianne Senko 72 Openshaw Rise Marehay Manchester MA0 4ZT 14/08/1977
48 John Alexander 8 Upper Pavement Uttoxeter Bath BA2 4NA 19/09/1972
49 Gillian Schoen 51 Low Street Borrowash Birmingham B05 8FT 06/11/1972
50 Michael Wilson 31 Nottingham Road Sandiacre Nottingham NTO 800 19/02/1985
51 Henry Weinberg 72 Harmer Drive Castle Donington Birmingham B04 4WW 08/09/1971
52 James Childs 14 Anderson Street Quorn Exeter EX3 0AQ 17/06/1982
53 Angela Ramamoorthy 7327 Allan Avenue Codnor Preston PR8 1JS 01/10/1982
54 Pauline Goldberg 14 Siddals Road Kilburn Nottingham NT9 2NW 09/05/1970
55 Alan Trowbridge 113 Derby Road Aston Lane Newcastle upon Tyne MV3 2KK 18/08/1972
56 Wilfred Pargitter 13 Kings Road Chaddesden Bath BA2 8UU 09/02/1988
57 Mirabelle Wilson 732 Osmaston Road Ilkeston Gloucester GL9 3HH 23/07/1981
58 Bernard Stapleton 28 Hands Road Hulland Ward Birmingham B&3 3PP 27/06/1979
59 Geoffrey Fullerton The Lodge Littleover Nottingham NT3 IR8 22/11/1978
60 Martin Suddes 12 Pastures View Chaddesden Exeter EX3 3KK 29/04/1986
61 Mary Dodgson 99 Derby Road Stapleford Manchester M03 5FUY 23/07/1971
62 John Gillespie 59 Burton Road Codnor Gloucester GL3 4NM 28/03/1983

```

## 6.2.Example using Java Swing

In this section we’ll create a more complex example which makes use of Java’s Graphical components, called “swing”.

Right-click the “src” folder in your SOA Gateway Java Tutorial project. Select, “New”, “File”, and enter “RisarisBankDemo.java”. Click Finish.

We want the user of this program to enter a customer ID, and this will generate a call to the SOA Gateway web service. The web service will return the details for one or more customers, and display in a list. The user can then select one of these customers, and request the Current Account details for that customer. This will generate a web service call to request the current account information. The resultant current Account balance and current overdraft limit will then be displayed to the user.

The following code sets up a basic template that we can start working on. See the comments in the code for more information.

```
/**
 * Import Java APIs
 */
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;

import SoaG.Tutorial.CurrentAccount.*;
import SoaG.Tutorial.CurrentAccount.CurrentaccountRootServiceStub.*;

import SoaG.Tutorial.CustomerInfo.*;
import SoaG.Tutorial.CustomerInfo.CustomerinformationRootServiceStub.*;

public class RisarisBankDemo implements ActionListener,
ListSelectionListener {

    private static JFrame frame;
    private static JPanel panel;

    /**
     * Set up the main frame
     */
    public static void createAndShowGUI() {
        RisarisBankDemo demo = new RisarisBankDemo();

        frame = new JFrame("Risaris Demo");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        frame.setContentPane( demo.createPanel() );

        frame.pack();
        frame.setVisible(true);
        return;
    }

    /**
     * Create and fill the main panel
     */
    public JPanel createPanel(){
        panel = new JPanel();
        return panel;
    }

    /**
     * Handle action events
     */
}
```

```

public void actionPerformed( ActionEvent e ){
    return;
}

/**
 * handle value changed events (row select)
 */
public void valueChanged(ListSelectionEvent e) {
    return;
}

/**
 * Main method
 */
public static void main(String[] args) {
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}
}

```

Now we expand on the “createPanel()” method. We add the required components to allow the user to enter information, and the components that will be used to display the results.

```

private static JTextField custNumText;
private static JButton searchButton;
private static JButton getAccountButton;
private static JTextField accountBalanceText;
private static JTextField accountOverdraftText;
private DefaultTableModel model;

. . .

/**
 * Create and fill the main panel
 */
public JPanel createPanel(){
    panel = new JPanel();

    panel.setLayout( new BorderLayout( panel, BorderLayout.Y_AXIS));

    //
    // Customer Search panel
    // A label, text field, and button in a row.
    //
    JPanel searchPanel = new JPanel();
    searchPanel.setLayout( new GridLayout( 1, 4 ));

    JLabel label1 = new JLabel("Customer Number");
    searchPanel.add( label1);

    JLabel dummy1 = new JLabel();
    searchPanel.add(dummy1);
}

```

```

custNumText = new JTextField();
custNumText.setText("");
searchPanel.add(custNumText);

searchButton = new JButton();
searchButton.setText("Search");
searchButton.addActionListener(this);
searchPanel.add(searchButton);

//
// Dummy Panel 1
//
JPanel dummyPanel1 = new JPanel();
dummyPanel1.setLayout(new GridLayout(1, 1));

JLabel dummy2 = new JLabel(" ");
dummyPanel1.add(dummy2);

//
// List panel
// This component will display the results of the customer lookup.
//
JPanel listPanel = new JPanel();
listPanel.setLayout(new BorderLayout(listPanel, BorderLayout.Y_AXIS));

String columnNames[] = { "Customer Number", "First Name", "Surname",
"Address1", "Address2", "City", "Zip", "DOB" };

model = new DefaultTableModel();
JTable table = new JTable(model);

for(int i=0; i !=columnNames.length; i++){
    model.addColumn( columnNames[i] );
}

table.setPreferredSize(new Dimension(700, 500));
table.setRowSelectionAllowed(true);
table.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);

table.getSelectionModel().addListSelectionListener(this);

JScrollPane scrollPane = new JScrollPane(table);

listPanel.add(scrollPane);

//
// Dummy Panel 2
//
JPanel dummyPanel2 = new JPanel();
dummyPanel2.setLayout(new GridLayout(1, 1));

JLabel dummy3 = new JLabel(" ");
dummyPanel2.add(dummy3);

//
// Get Current Account Info panel
// This panel will allow the user to lookup the account details for
// currently selected customer.
//
JPanel currentAcPanel = new JPanel();
currentAcPanel.setLayout(new GridLayout(3, 4));

```

```

    getAccountButton = new JButton();
    getAccountButton.setText("Get Account Details");
    getAccountButton.addActionListener(this);
    currentAcPanel.add( getAccountButton);

    JLabel dummy4 = new JLabel();
    JLabel dummy5 = new JLabel();
    JLabel dummy6 = new JLabel();
    currentAcPanel.add(dummy4);
    currentAcPanel.add(dummy5);
    currentAcPanel.add(dummy6);

    JLabel label2 = new JLabel("Current Account Balance");
    currentAcPanel.add( label2);

    JLabel dummy7 = new JLabel();
    currentAcPanel.add(dummy7);

    accountBalanceText = new JTextField();
    accountBalanceText.setEditable(false);
    currentAcPanel.add( accountBalanceText);

    JLabel dummy8 = new JLabel();
    currentAcPanel.add(dummy8);

    JLabel label3 = new JLabel("Current Account Overdraft");
    currentAcPanel.add(label3);

    JLabel dummy9 = new JLabel();
    currentAcPanel.add(dummy9);

    accountOverdraftText = new JTextField();
    accountOverdraftText.setEditable(false);
    currentAcPanel.add( accountOverdraftText );

    JLabel dummy10 = new JLabel();
    currentAcPanel.add(dummy10);

    //
    // Dummy Panel 3
    //
    JPanel dummyPanel3 = new JPanel();
    dummyPanel3.setLayout( new GridLayout( 1, 1));

    JLabel dummy11 = new JLabel(" ");
    dummyPanel3.add( dummy11);

    panel.add(searchPanel);
    panel.add(dummyPanel1);
    panel.add(listPanel);
    panel.add(dummyPanel2);
    panel.add( currentAcPanel);
    panel.add(dummyPanel3);

    return panel;
}

```

The createPanel method requires 2 event handlers. The first handler is used with the “Search” and “Get Account Details” button-press. The second handler is used to set the selected row in the list of customers.

```
private String selectedID = "";

. . .

/**
 * Handle action events (button press)
 */
public void actionPerformed( ActionEvent e ){
    if( e.getActionCommand().toString().equals("Search")){
        searchCustomer();
    }
    else if ( e.getActionCommand().toString().equals( "Get Account
Details")){
        getAccountDetails();
    }

    return;
}

/**
 * handle value changed events (row select)
 */
public void valueChanged(ListSelectionEvent e) {

    if (e.getValueIsAdjusting()) {
        return;
    }

    ListSelectionModel lsm = (ListSelectionModel)e.getSource();
    selectedID = (String)model.getValueAt(lsm.getMaxSelectionIndex(), 0);
}
```

The “searchCustomer” and “getAccountDetails” methods will actually make the web service calls to the SOA Gateway to retrieve the necessary information. Both methods are very similar to the “SimpleTest.java” program we implemented earlier. Note the user name of `root` and password of `letmein` below. Change to those of your MySQL setup.

The code is:

```
private String userName = "root";
private String password = "letmein";

. . .

/**
 * Take the customer number, and call the SOA Gateway web
 * service to get the customer information.
 * Add the customer information to the list table;
 */
public void searchCustomer() {
    //
    // Set up the security credentials
    //
```

```

        CustomerinformationRootServiceStub.Security sec = new
CustomerinformationRootServiceStub.Security();
        CustomerinformationRootServiceStub.UsernameToken_type0 token =
new CustomerinformationRootServiceStub.UsernameToken_type0();
        token.setUsername(userName);
        token.setPassword(password);
        sec.setUsernameToken(token);

        try{
            CustomerinformationRootServiceStub stub = new
CustomerinformationRootServiceStub();

            //
            // Set up the key based on the current value in the text box.
            //
            CustomerinformationGroupKeyType key = new
CustomerinformationGroupKeyType();
            key.setCustomerNumber(custNumText.getText());

            CustomerinformationGroupListElement listKey = new
CustomerinformationGroupListElement();
            listKey.setCustomerinformationGroupListElement(key);

            //
            // Will hold the results of the web service
            //
            CustomerinformationRootElement results;

            // make the call!
            results = stub.list(listKey, sec, null);

            //
            // Display the results by adding to the table
            //
            CustomerinformationRootType root =
results.getCustomerinformationRootElement().getCustomerinformationRoot();

            for( int i = 0;
                i != root.getCustomerinformationGroup().length ;
                i++){

                CustomerinformationGroupType[] customer =
root.getCustomerinformationGroup();

                Object [] thisRow = new Object[]{
customer[i].getCustomerNumber(), customer[i].getFirstName(),
customer[i].getSurname(),

                customer[i].getAddressLine1(), customer[i].getAddressLine2(),
customer[i].getCity(),
customer[i].getPostcode(),
customer[i].getDateOfBirth() };

                model.addRow( thisRow );
            }
        }
        catch(Exception e ){
            JOptionPane.showMessageDialog( null, e.getMessage(),
"Error" , JOptionPane.ERROR_MESSAGE);
            e.printStackTrace();
        }
    }

```

```

}

/**
 * Get the account details for the current selected customer
 */
private void getAccountDetails () {
    //
    // Set up the security credentials
    //
    CurrentaccountRootServiceStub.Security sec = new
CurrentaccountRootServiceStub.Security ();
    CurrentaccountRootServiceStub.UsernameToken_type0 token = new
CurrentaccountRootServiceStub.UsernameToken_type0 ();
    token.setUsername (userName);
    token.setPassword (password);
    sec.setUsernameToken (token);

    try {
        CurrentaccountRootServiceStub stub = new
CurrentaccountRootServiceStub ();

        //
        // Set up the key based on the current value in the text
box.
        //
        CurrentaccountGroupKeyType key = new
CurrentaccountGroupKeyType ();
        key.setCustomerNumber (selectedID);
        key.setAccountNumber ("");

        CurrentaccountGroupListElement listKey = new
CurrentaccountGroupListElement ();
        listKey.setCurrentaccountGroupListElement (key);

        //
        // Will hold the results of the web service
        //
        CurrentaccountRootElement results;

        // make the call!
        results = stub.list (listKey, sec, null);

        //
        // fill out the balance and overdraft text boxes
        CurrentaccountRootType root =
results.getCurrentaccountRootElement ().getCurrentaccountRoot ();

        CurrentaccountGroupType [] customer =
root.getCurrentaccountGroup ();

        if ( customer != null ) {
            Float bal =
                Float.valueOf (customer [0].getBalance ())
                    / 100;
            Float od =
                Float.valueOf (customer [0].getOverdraftLimit ())
                    / 100;

            accountBalanceText.setText ( bal.toString () );
            accountOverdraftText.setText (od.toString () );
        }
    }
}

```

```

        else {
            accountBalanceText.setText("0");
            accountOverdraftText.setText("0");
        }
    }
    catch (Exception e) {
        JOptionPane.showMessageDialog( null, e.getMessage(),
"Error" , JOptionPane.ERROR_MESSAGE);
        e.printStackTrace();
    }
}

```

That's it! The program in full is available in the appendix at the end of this document.

### 6.3. Building the Code

Hopefully, all is well, but in the case of errors try the following:

- Obviously misspellings are often the cause of compilation errors. Ensure that all object names, and variable names are spelt correctly.
- When copying from PDF files, the double-quote characters can sometimes appear as double-backticks, which will cause a build error. Ensure double-quotes are used.
- The Eclipse IDE has a neat trick where you can type the first few letters of an object, and by hitting Control + Space, it will bring up the suggested object names. For names starting with "cu", such as "Customerinformation" or "Currentaccount", try typing this and hitting Control + Space. It may give you an indication of a misspelt object name.
- Similarly, if you need the name of a member of an object, type the object name, followed by a dot (".") and Control+Space. The list of available proposals should appear.

### 6.4. Running the code

To run your program, right-click "RisarBankDemo.java" and select "Run As", and "Java Application".

The program will appear on the screen

**Rissaris Demo**

Customer Number  **Search**

Customer Nu...	First Name	Surname	Address1	Address2	City	Zip	DOB
----------------	------------	---------	----------	----------	------	-----	-----

**Get Account Details**

Current Account Balance

Current Account Overdraft

In the Customer Number text box, enter \* and hit Search.

The program will access the SOA Gateway web service to list customer records from the MySQL database. The list of customers will be displayed.

E.g.

Risaris Demo

Customer Number  \*

Customer Nu...	First Name	Surname	Address1	Address2	City	Zip	DOB
1	Casper	Ankergren	34 Green Str...	Crosses Street	Mansfield	MA5 9AJ	30/05/1978
2	Peter	Richardson	West Gate Av...	Frankfield	Manchester	MA9 4AA	18/07/1948
3	Donald	Smith	32 Parsons ...	Fulham	London	SW4 3JA	05/09/1988
4	Lillian	Marques	1 Father Del...	Bordesley	Birminham	BO4 4LW	14/01/1979
5	Arthur	Holman	12 Padbury ...	Griffin Close	Newcastle u...	NW8 2KK	16/05/1975
6	Bartholomew	McCarthy	22 Hillview ...	Griffin Close	York	SW9 1GH	24/03/1974
7	Neville	Lee	72 Thoresby ...	Sinfin	Leeds	LS9 2DS	21/03/1980
8	Brian	Colman	14 Gilliver Ga...	West Field	Birmingham	BA9 3KK	20/04/1983
9	Ian	Burton	144 Allestree...	Lime Kiln Lane	York	YR0 2NT	06/09/1971
10	Grenville	Dekker	22 Uttoxter N...	Chaddesden	Leeds	LS4 9WB	21/10/1974
11	Jane	Freeman	3 Portland St...	Lichfield	Gloucester	GL4 8KD	26/10/1980
12	Dale	Rolling	The Hunters ...	Branston	Preston	PR4 4HG	06/12/1989
13	Arnold	Corrigan	14 Anderson ...	Allestree	Derby	DE6 8FT	10/06/1970
14	Alfred	Summerscale	3701 S. Geor...	Melbourne	Bath	BA6 8UU	07/08/1982
15	Peter	Spiegel	11663 Chart...	Foremark	Manchester	M98 4GT	29/12/1971
16	John	Sviridov	3319 Rosem...	Brailsford	Leeds	LS8 5QQ	12/06/1984
17	Sarah	Hall	4 Denmark L...	East Bridgford	Portsmouth	PO9 7QN	12/02/1983
18	Elspeth	Jones	12375 W. Oh...	Risley	Birmingham	B93 1IO	28/08/1986
19	Nigel	Wyllis	14 Borough ...	Ockbrook	Leeds	LS8 2WL	29/04/1981
20	Beatrice	Smith	The Willows	Risley	Derby	DE6 7DT	29/11/1985
21	Suzanne	King	3 Devonshire...	Allenton	Portsmouth	PO3 4JE	30/08/1981
22	Malcolm	Wallace	The Old Rect...	Ilkeston	Bath	BA4 9WH	05/07/1981
23	Fiona	Grobe	New Farm	East Leake	Derby	DE2 4KW	14/06/1978
24	Lesley	Wells	23 Charlotte ...	Oakham	Newcastle u...	NW6 3NJ	06/05/1982
25	Harry	Chu	26 Edinburg...	Kegworth	Preston	PR4 3KN	30/09/1978
26	Simon	Jones	24 Main Street	Draycott	Manchester	MK3 4MA	02/10/1971
27	Eileen	Smith	14 Grange Av...	Burton-On-Tr...	Portsmouth	PO3 4NA	18/09/1970
28	Helen	Ferrari	15 New Street	Borrowwash	Leeds	LS4 2WN	01/04/1970
29	Stephen	Romerio	22 Normanto...	Draycott	Birmingham	B93 3YL	01/03/1974
30	Helen	Dilworth	99 Derby Road	Mickleover	Bath	BA3 4QZ	04/05/1976
31	Denise	Birkhof	73 High Street	Columbus	Lincoln	LN3 9AA	08/01/1981

Current Account Balance

Current Account Overdraft

Choose a row from the list, and hit "Get Account Details". The Current Account web service will be called to retrieve the customers account information from MySQL.

E.g.

15	Peter	Spiegel	11663 Chart...	Foremark	Manchester	M98 4GT	29/12/1971
16	John	Sviridov	3319 Rosem...	Brailsford	Leeds	LS8 5QQ	12/06/1984
17	Sarah	Hall	4 Denmark L...	East Bridgford	Portsmouth	PO9 7QN	12/02/1983
18	Elsbeth	Jones	12375 W. Oh...	Risley	Birmingham	B93 1IO	28/08/1986
19	Nigel	Wyllis	14 Borough ...	Ockbrook	Leeds	LS8 2WL	29/04/1981
20	Beatrice	Smith	The Willows	Risley	Derby	DE6 7DT	29/11/1985
21	Suzanne	King	3 Devonshire...	Allenton	Portsmouth	PO3 4JE	30/08/1981
22	Malcolm	Wallace	The Old Rect...	Ilkeston	Bath	BA4 9WH	05/07/1981
23	Fiona	Grobe	New Farm	East Leake	Derby	DE2 4KW	14/06/1978
24	Lesley	Wells	23 Charlotte ...	Oakham	Newcastle u...	NW6 3NJ	06/05/1982
25	Harry	Chu	26 Edinburg...	Kegworth	Preston	PR4 3KN	30/09/1978
26	Simon	Jones	24 Main Street	Draycott	Manchester	MK3 4MA	02/10/1971
27	Eileen	Smith	14 Grange Av...	Burton-On-Tr...	Portsmouth	PO3 4NA	18/09/1970
28	Helen	Ferrari	15 New Street	Borrowash	Leeds	LS4 2WN	01/04/1970
29	Stephen	Romerio	22 Normanto...	Draycott	Birmingham	B93 3YL	01/03/1974
30	Helen	Dilworth	99 Derby Road	Mickleover	Bath	BA3 4QZ	04/05/1976
31	Denise	Birkhof	73 High Street	Columbus	Lincoln	LN3 9AA	08/01/1981

<input type="button" value="Get Account Details"/>	
<b>Current Account Balance</b>	8504.79
<b>Current Account Overdraft</b>	1000.0

If you hit problems, you may wish to debug your code by adding breakpoints in your code. See the IDE documentation for further information.

## 7. Conclusion

This tutorial shows how to access MySQL from Java using the SOA Gateway. As you can see, you have built a powerful application that uses Web Services to retrieve information in real-time.

## 8. Appendix

### SimpleTest.java

[Code available here](#)

### RisarisBankDemo.java

[Code available here](#)