

Accessing MySQL from PHP

Contents

1. Introduction	2
2. Prerequisites	2
3. Setup	2
3.1. PHP Configuration.....	2
3.2. Apache configuration.....	3
3.3. Populate MySQL Database.....	3
3.4. Set up ODBC Access	4
4. Discovery.....	5
4.1. Web Service Creation using SOA Gateway	5
4.2. Accessing the WSDL	7
5. Accessing Web Service with PHP	9
5.1. Initial Setup	9
5.2. The Form	10
5.3 Code	11
5.3 Query the database.....	11
6. Conclusion.....	13
7. Appendix	13

1. Introduction

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

2. Prerequisites

It is assumed that you are running the 3 components, MySQL, PHP 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

Download the latest Windows binaries for PHP v5.x from the [PHP downloads page](#), install it according to the install.txt document in the PHP distribution library.

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.

You will also require a version of Apache installed and it is here that the PHP script is stored. See [Apache Downloads Site](#) for the binaries and documentation explaining how to install Apache for Windows.

3.1.PHP Configuration

Enable the PHP SOAP Extension. The soap extension is shipped as part of PHP 5, but needs to be enabled explicitly.

- If you downloaded the PHP source code and compiled it for your platform, you will probably need to rebuild, because ext/soap is not enabled by default. Repeat your previous build process, adding the `--enable-soap` option to the configure command.
- If you downloaded pre-compiled platform binaries, they have ext/soap compiled in, but not loaded, so you will need to update your PHP configuration to do so. Edit your php.ini (you might have to rename or copy it from php.ini-recommended), and look for the Dynamic Extensions section. Here you will need to add a line that causes the extension to be loaded automatically:

```
extension=php_soap.dll
```

If you have not previously loaded any optional extensions, you may also have to set the `extension_dir` directive to point to the directory containing the extension libraries, including `php_soap`, for example:

```
extension_dir="C:/php/ext/" (use forward slashes even on Windows)
```

Do not try to put directory information in the extension directive; use `extension_dir` if necessary.

3.2. Apache configuration

To enable Apache and PHP to interact you need to add your PHP script in a location known to Apache (see 5.1 for more on PHP scripting).

Go to the conf directory of Apache (e.g. C:\Program Files\Apache Group\Apache2\conf). Open file httpd.conf and find the entry for DocumentRoot:

e.g.

```
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "C:/Program Files/Apache Group/Apache2/htdocs"
```

In this case C:/Program Files/Apache Group/Apache2/htdocs is where any PHP scripts must be placed.

3.3. 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 ...

Depending on your MySQL settings table names may appear all in lowercase e.g. customerinformation or CustomerInformation.

3.4.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

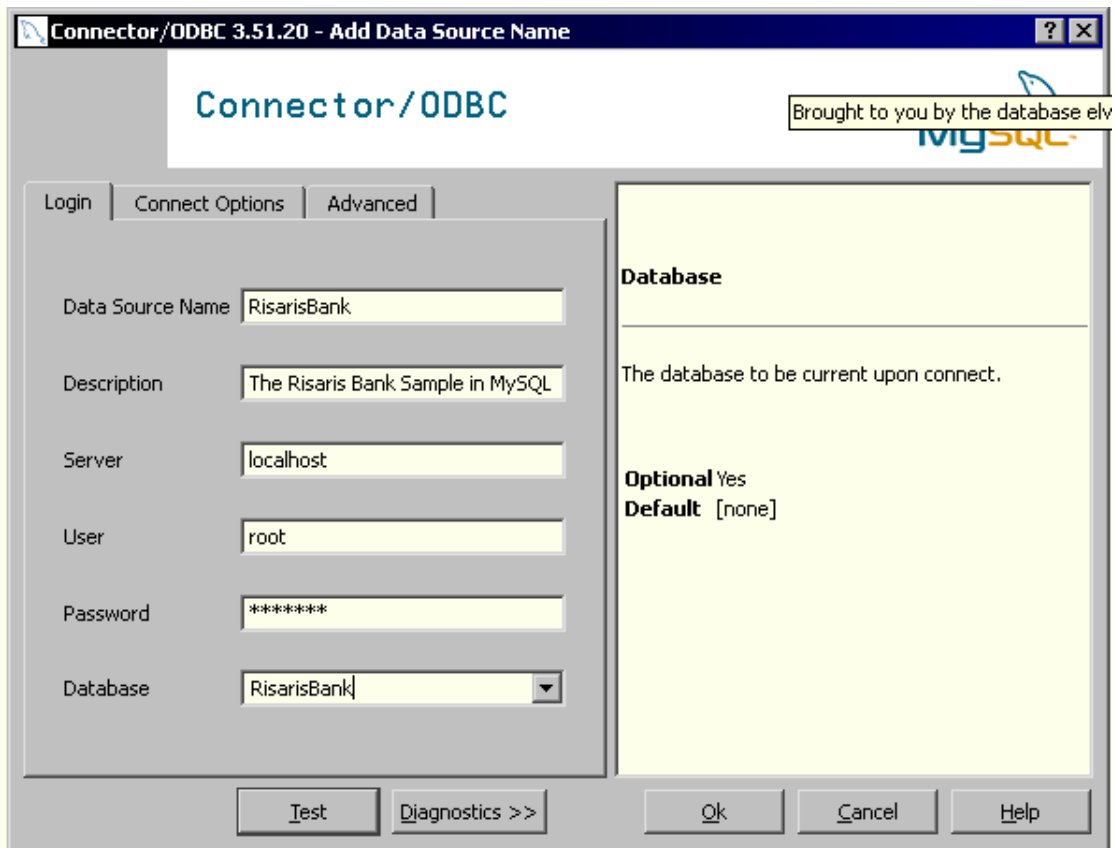
Description: The Risaris Bank Sample in MySQL

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.

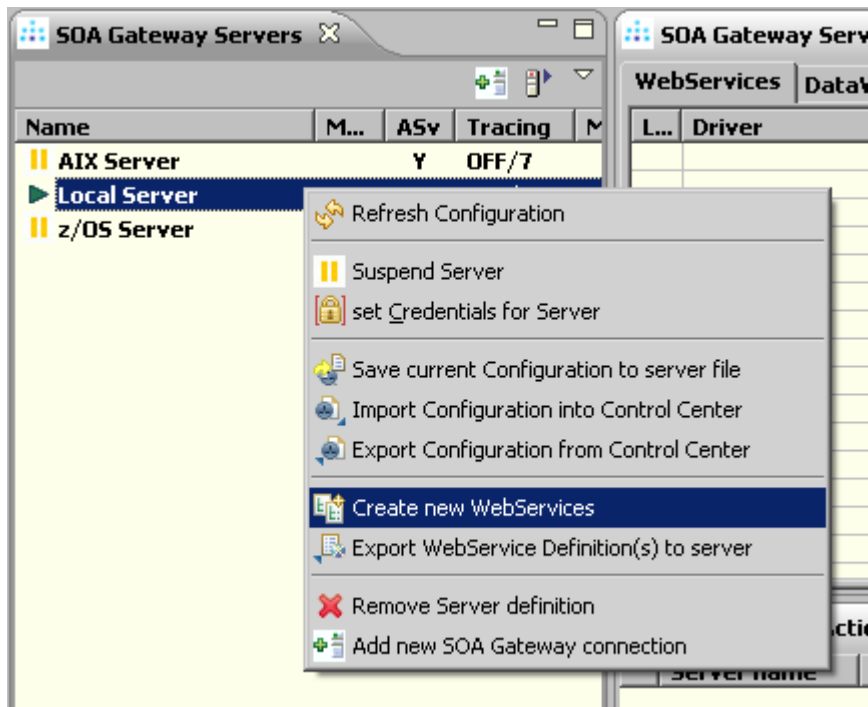
4. Discovery

At this stage you've got PHP installed, 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 PHP language (and many others) to give you direct real-time access to your MySQL Data.

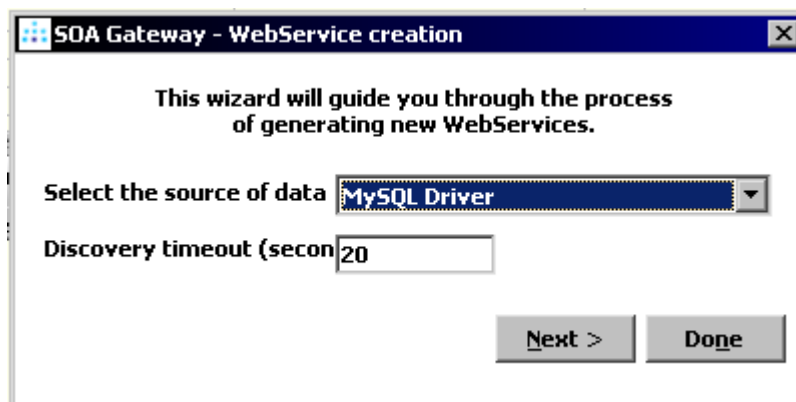
4.1. Web Service Creation using SOA Gateway

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".

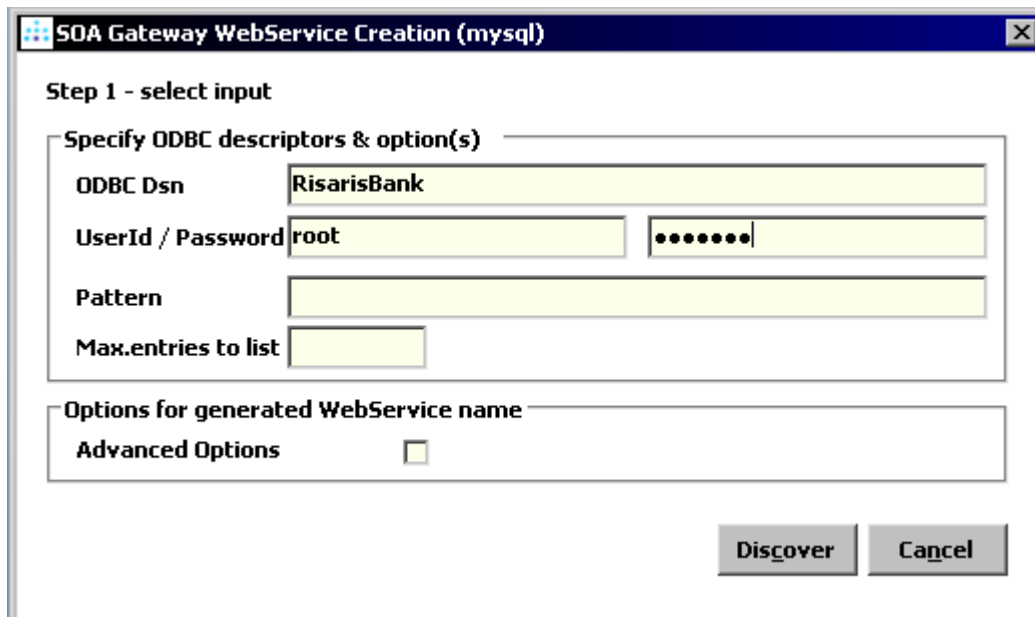


From the next dialog, choose “MySQL Driver”. If you do not see have a MySQL Driver in the list, see how to create one [here](#).



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.

SOA Gateway Servers			SOA Gateway Server Configuration - Local Server				
			WebServices DataViews / XSDs / XSLs				
Name	M...	ASv	Mod	Driver	WebService	DataSource Id	DataView
AIX Server		Y		MySQL Driver	accountsmovements	odbcDsn=RisarisBank, tableName=accountsmovements	accountsmovements
DMZ		Y		MySQL Driver	audit	odbcDsn=RisarisBank, tableName=audit	audit
dublin dev		Y		MySQL Driver	branch	odbcDsn=RisarisBank, tableName=branch	branch
jk server		Y		MySQL Driver	currentaccount	odbcDsn=RisarisBank, tableName=currentaccount	currentaccount
jk server linux		Y		MySQL Driver	customeraccountxref	odbcDsn=RisarisBank, tableName=customeraccountxref	customeraccountxref
jom server		Y		MySQL Driver	customerinformation	odbcDsn=RisarisBank, tableName=customerinformation	customerinformation
Local Server		Y		MySQL Driver	depositaccount	odbcDsn=RisarisBank, tableName=depositaccount	depositaccount
lxbre server		Y		MySQL Driver	tellertable	odbcDsn=RisarisBank, tableName=tellertable	tellertable
PCRJW9		Y		MySQL Driver			
risaris.com server		Y		MySQL Driver			
vse		Y		MySQL Driver			
z/OS Server		Y		MySQL Driver			
z/vse		Y		MySQL Driver			

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 PHP, 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 Risaris 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 for a local server. It is divided into three main sections:

- WebServices:** A table listing various web services. The 'branch' service is highlighted.
- SOA Gateway Action Log:** Shows messages such as 'discovery completed, 8 WebService(s) generated' and 'autosaved due to published WebService modification(s)'.
- Properties:** The 'WebService properties' view for the 'branch' service. It includes fields for Name, DataView, Driver, Read-only, SBCS-Codepage, and MBCS-Codepage. The WSDL URL is displayed as <http://localhost:56000/branch?WSDL>. Below this, the 'WebService Identification and options' section shows fields for jdbcDsn (RisarisBank), schemaName, and tableName (branch).

A green arrow points from the 'branch' service in the WebServices table to the 'branch' service in the Properties view.

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

Local Server | Message

Local Server | discovery completed, 8 WebService(s) generated

Local Server | autosaved due to published WebService modification(s)

Local Server |

WebService properties

Resource | WebService

Name | branch

DataView | branch

Driver | MySQL Driver | SBCS-Codepage |

Read-only | | MBCS-Codepage |

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

WebService Identification and options

jdbcDsn | 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 PHP

PHP: Hypertext Preprocessor is a popular server-side scripting language for creating dynamic Web content. PHP is available as source code or as pre-compiled binaries for major platforms, including most Linux™ distributions, Windows®, Mac OS X, iSeries™.

We will use a PHP script to access our new Risaris Bank Web Services via the WSDL. Those familiar with PHP will probably want to change the script provided but it can be run with minimum changes described below.

5.1. Initial Setup

PHP can be written using any editor such as Wordpad, Notepad, etc and saved with a .php extension. The resultant file must then be placed in the Apache documentation folder (e.g. C:\Program Files\Apache Group\Apache2\htdocs). Check your Apache httpd.conf file for the DocumentRoot entry:

```
#  
# DocumentRoot: The directory out of which you will serve your  
# documents. By default, all requests are taken from this directory, but  
# symbolic links and aliases may be used to point to other locations.  
#  
DocumentRoot "C:/Program Files/Apache Group/Apache2/htdocs"
```

Appendix A shows the full script used in this tutorial. Copy the full contents of this into a file called mysql_tutorial.php.

The script can then be invoked via a browser such as MS Internet Explorer which is used in this tutorial. Note the URL of http://localhost/mysql_tutorial.php


```

<?php echo $msg; ?><br><br>
<input type="submit" value="List" name="submit">
<input type="submit" value="Get" name="submit">
<input type="submit" value="Add" name="submit">
<input type="submit" value="Update" name="submit">
<input type="submit" value="Delete" name="submit">
<input type="submit" value="Reset" name="submit">
</form>

```

5.3 Code

SoapClient

Note the URL for accessing the WSDL. Due to an as yet unresolved bug with Vista we have to use 127.0.0.1 instead of localhost. Note also the service name of customerinformation. Depending on your MySQL settings this may need to change to CustomerInformation. See section 4.2 for checking the generated WSDL. This name is repeated throughout the code.

```

try {
    $soapclient = new SoapClient (
        "http://127.0.0.1:56000/customerinformation?WSDL");
    } catch (SoapFault $soapfault) {
        echo "<pre>"; print_r($soapfault); echo "</pre>";
        return;
    }
}

```

Username/Password

Depending on your existing MySQL setup you may need to modify the login details. Below we are logging in as user root with password letmein. Change as necessary for each instance.

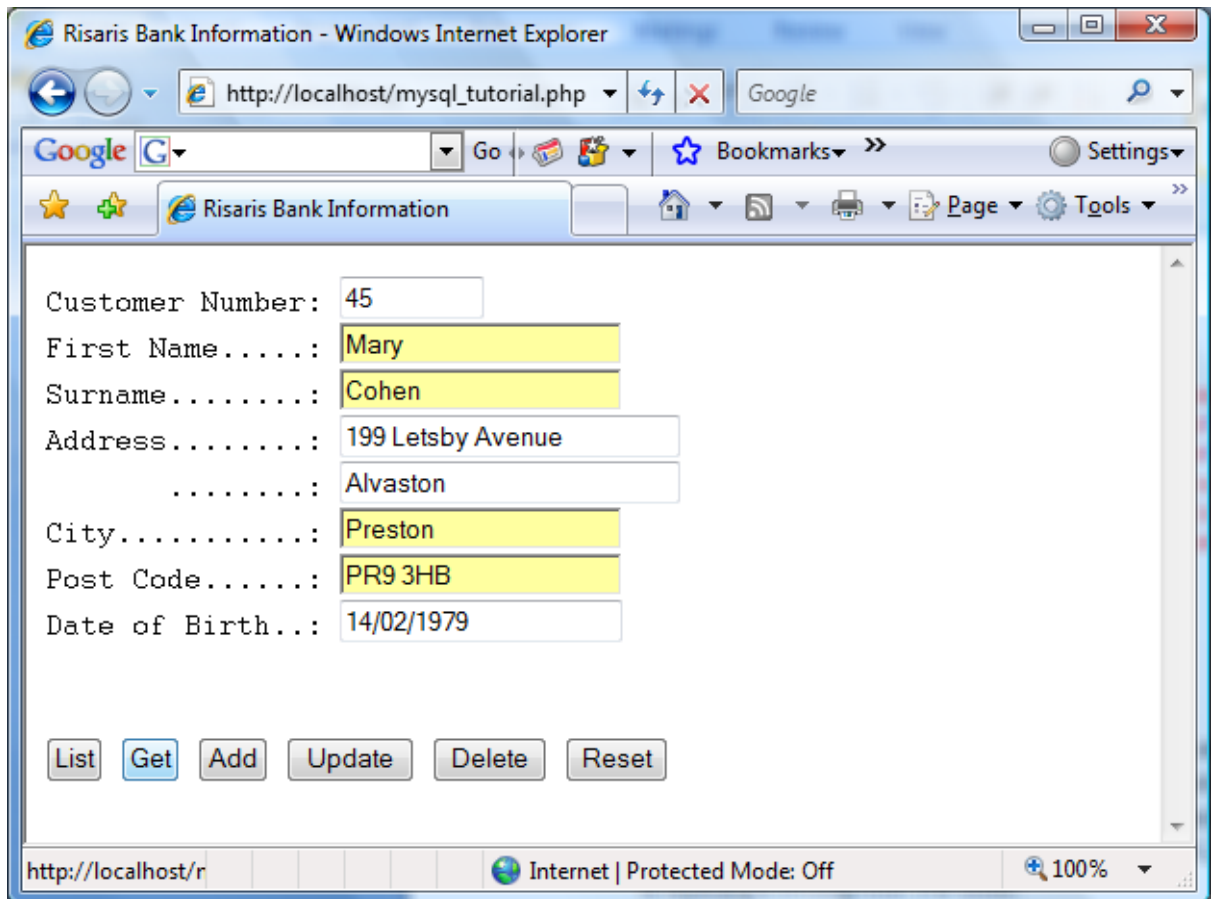
```

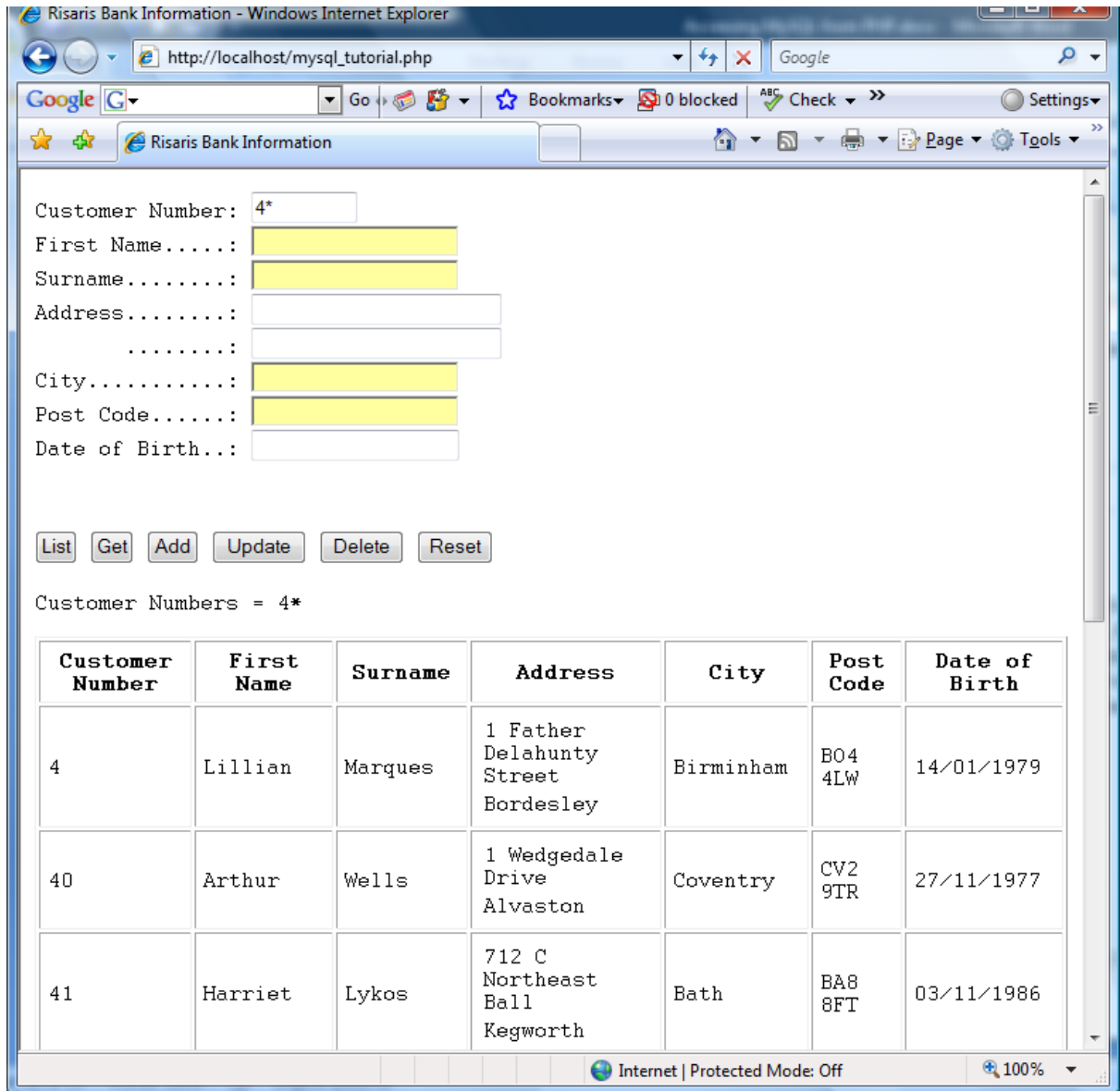
$getkey = array('CustomerNumber' => $Cnumber);
$sec = new securityHeader ( 'root', 'letmein' );
$securityHeader = new SoapHeader( "http://www.risaris.com/security",
    "Security" , $sec );
    $Mysqlresponse = $soapclient->__soapCall( 'get',
array($getkey), null , $securityHeader, $output_headers);

```

5.3 Query the database

Each of the buttons, apart from Reset, issues a database query via the SOA Gateway. In the example below 45 was entered as the Customer Number and Get selected. Note that Get, Add, Update and Delete require the Customer Number as the key. The List request on the other hand can accept wildcard entries for Customer Number e.g. 4* which will return all customers with numbers beginning with 4. As there can be several relevant entries the results are displayed beneath the buttons as shown below.





6. Conclusion

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

7. Appendix

Copy the all the code below into a file and name it mysql_tutorial.php. See 5.1 for next steps.

[Code available here](#)