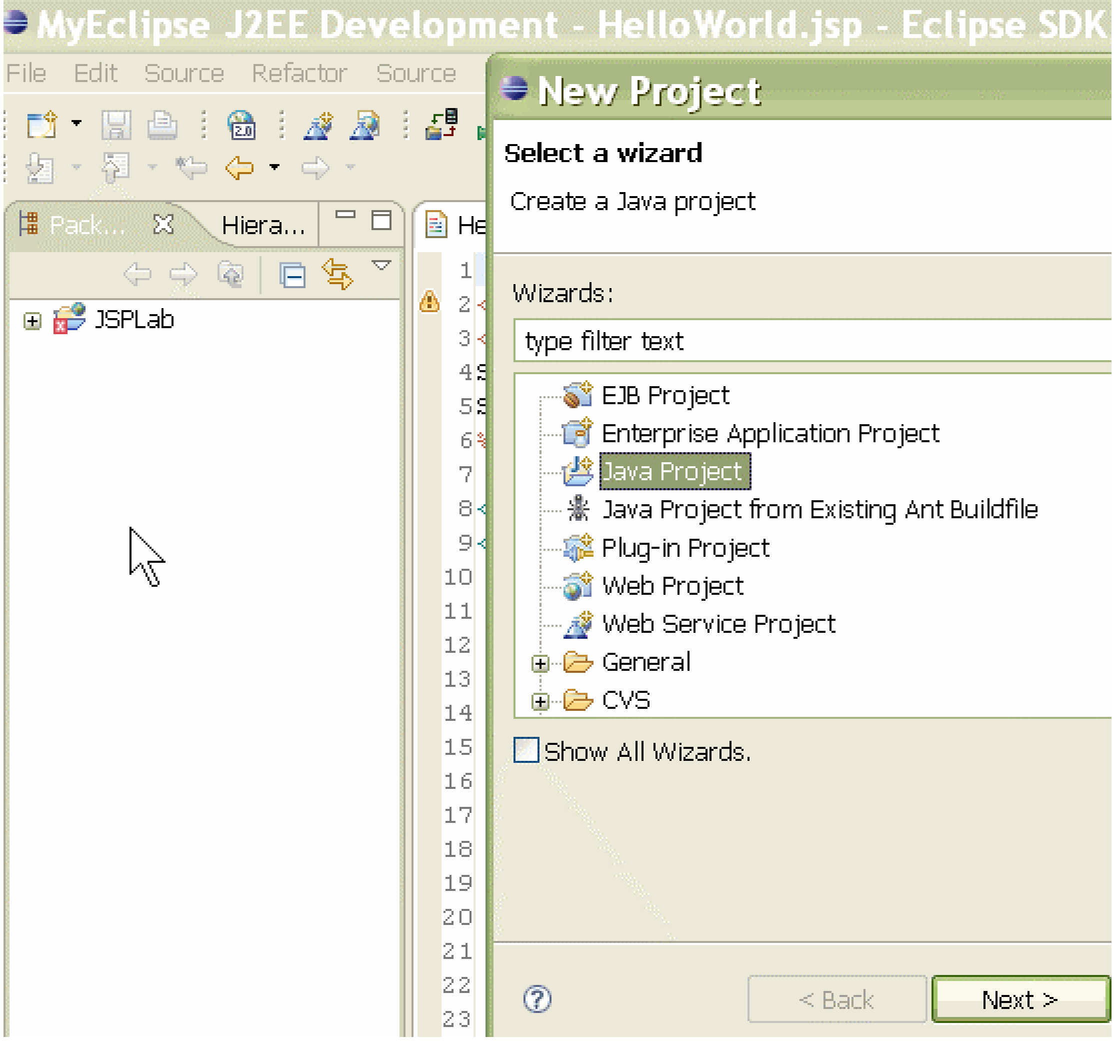
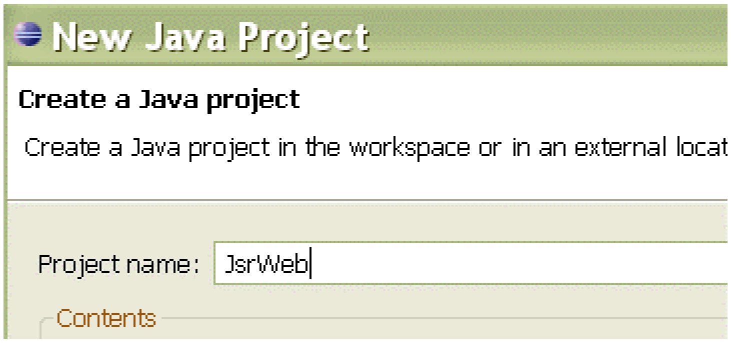
JSP and MySQL in MyEclipse

Exercise 3 SQL:

Stop Tomcat! Import JsrWeb project (you will need to create a "Java" project first).



....

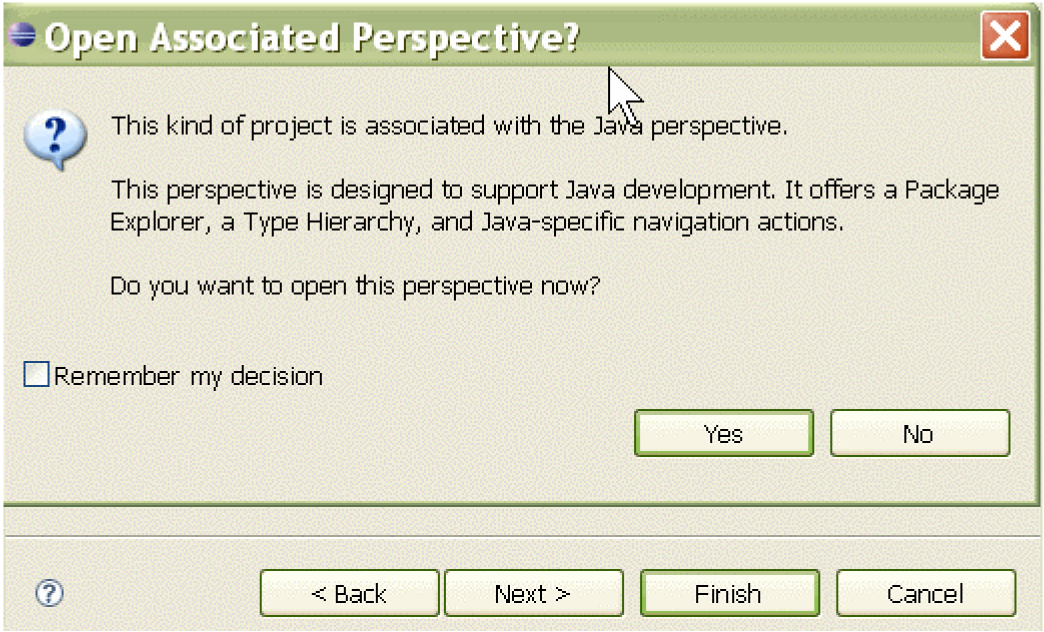


Create a Java project

Create a Java project in the workspace or in an external locat

Project name: JsrWeb

Name it JsrWeb Click FINISH, and switch to Java Perspective.



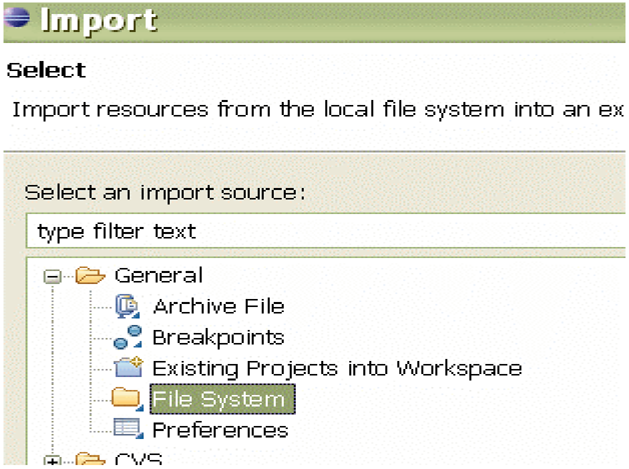
This kind of project is associated with the Java perspective.

This perspective is designed to support Java development. It offers a Package Explorer, a Type Hierarchy, and Java-specific navigation actions.

Do you want to open this perspective now?

☐ Remember my decision

From JsrWeb Java Project: Import/File System/



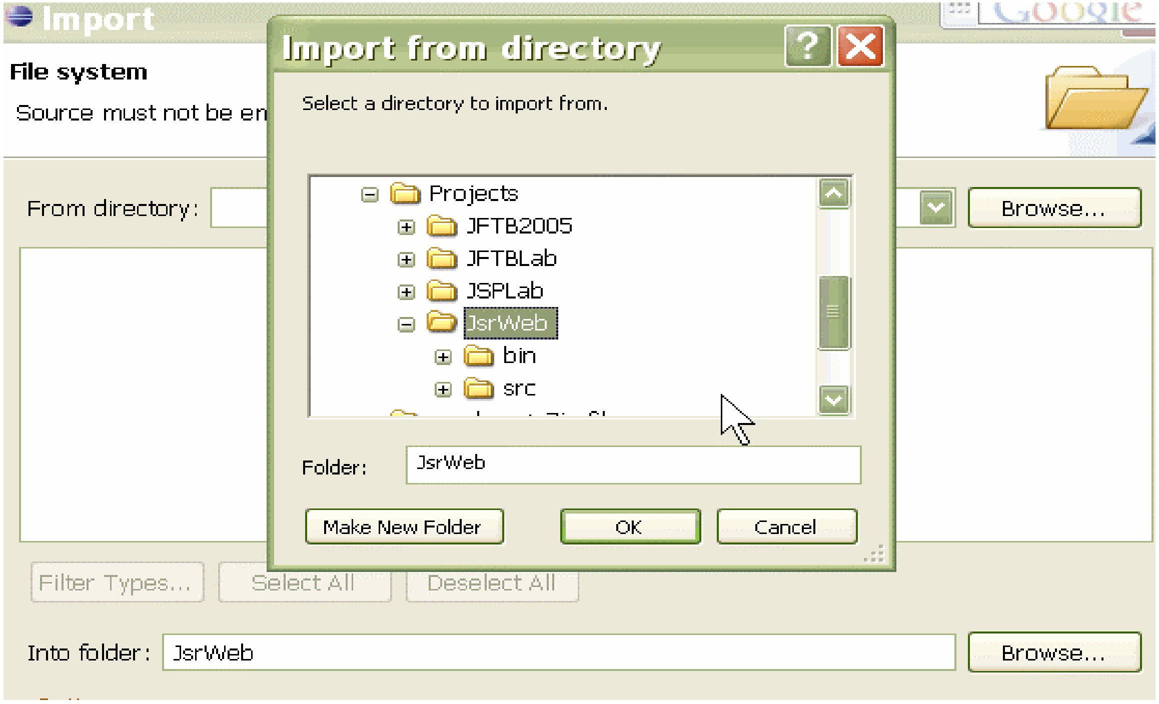
Select

Import resources from the local file system into an ex

Select an import source:

type filter text

Navigate to (Desktop)/JavaLabs/Projects/JsrWeb



Import

File system

Source must not be em

Import from directory

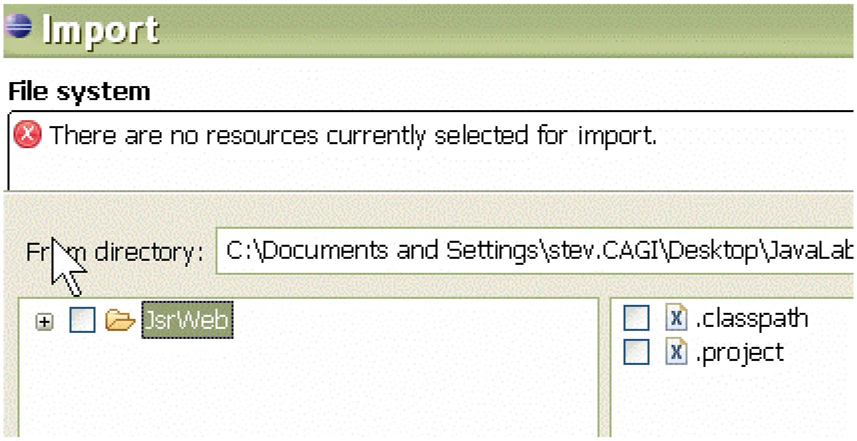
Select a directory to import from.

Projects
JFTB2005
JFTBLab
JSPLab
JsrWeb
bin
src

Folder: JsrWeb

Into folder: JsrWeb

......



Import

File system

There are no resources currently selected for import.

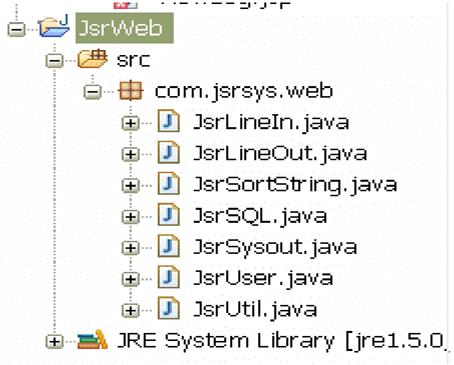
From directory: C:\Documents and Settings\stev.CAGI\Desktop\JavaLat

☐ JsrWeb

☐ .classpath
☐ .project

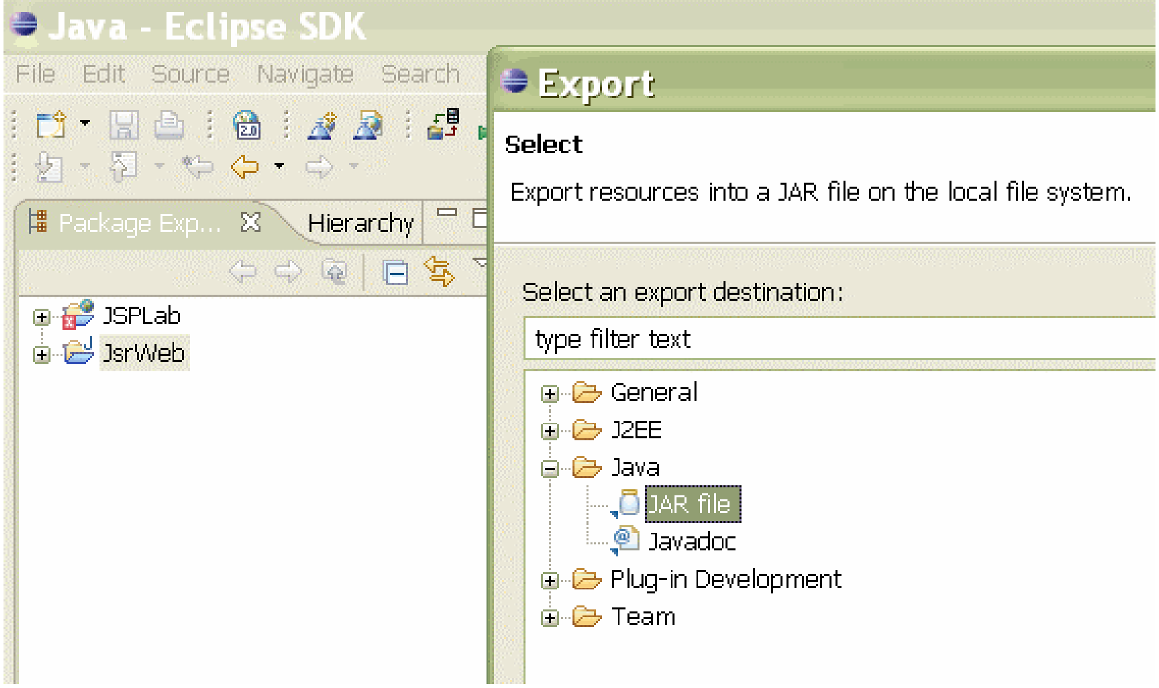
Be sure to check JsrWeb resource.

Then verify that programs exist in package com.jsrsys.web



```
JsrWeb
  src
    com.jsrsys.web
      JsrLineIn.java
      JsrLineOut.java
      JsrSortString.java
      JsrSQL.java
      JsrSysout.java
      JsrUser.java
      JsrUtil.java
  JRE System Library [jre1.5.0
```

Right click JsrWeb>Export>Jar File (Next).



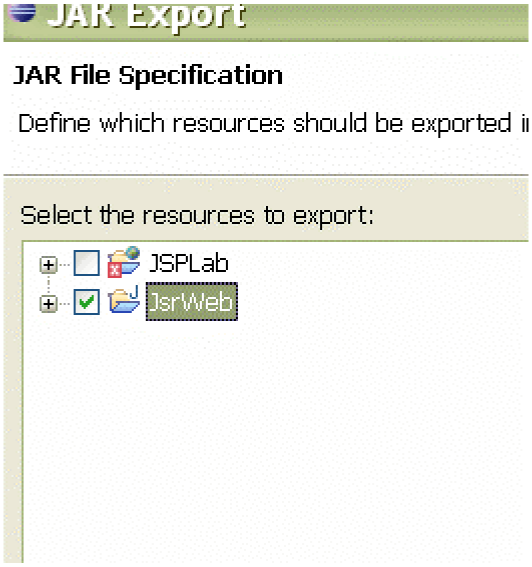
Select

Export resources into a JAR file on the local file system.

Select an export destination:

type filter text

...



JAR File Specification

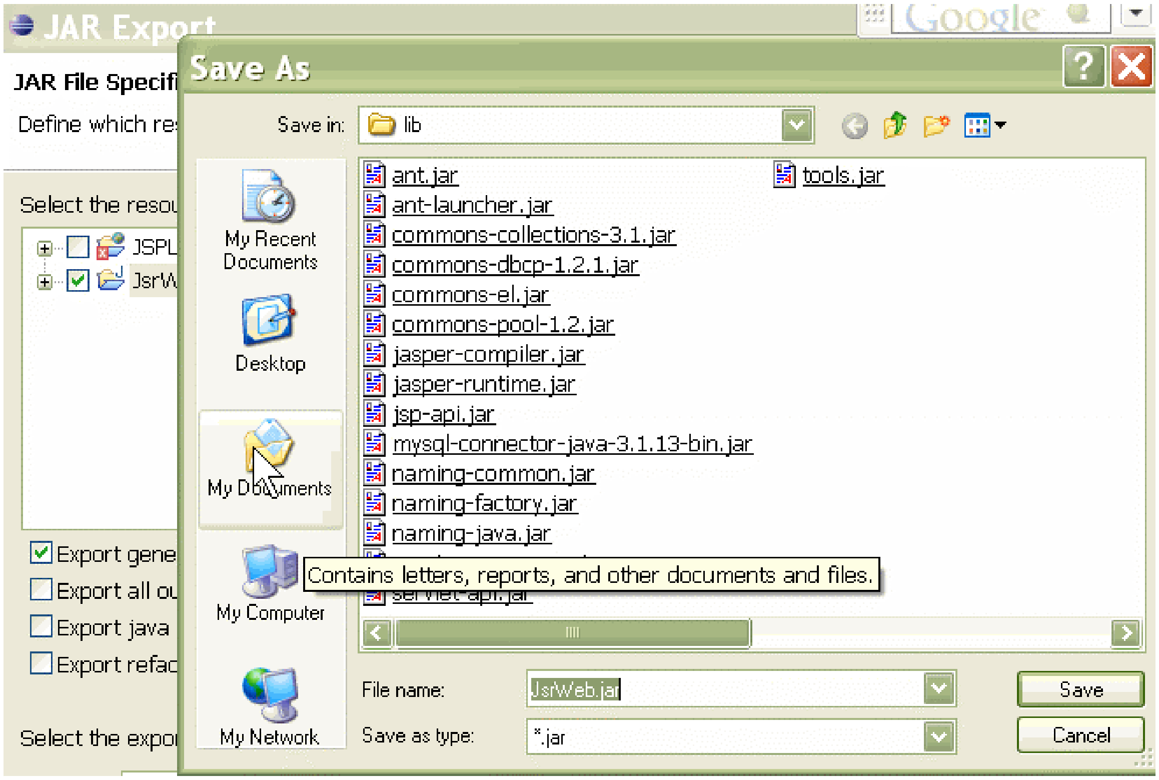
Define which resources should be exported i

Select the resources to export:

☐ JSPLab
☑ JsrWeb

Verify that JsrWeb is checked in Select resources… box;

...



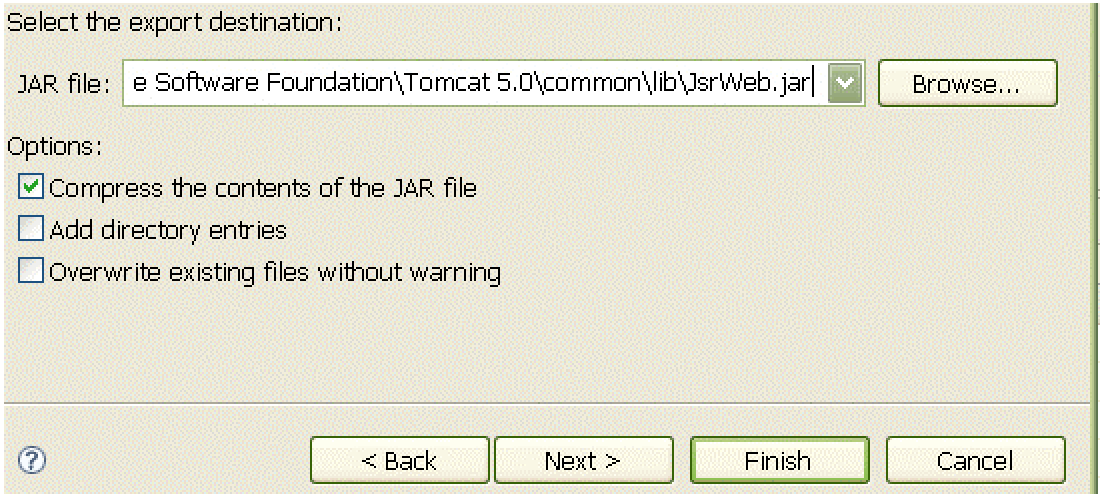
Save in: lib

Contains letters, reports, and other documents and files.

File name: JsrWeb.jar

Save as type: *.jar

in Jar File: (Browse) to C:\Program Files\Apache …\Tomcat 5.0\common\lib fill in name of target “JsrWeb.jar”. Click Save



Select the export destination:

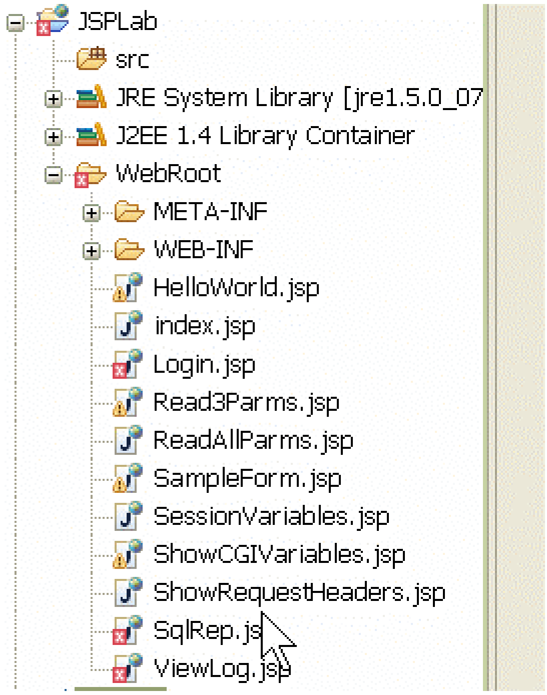
JAR file: e Software Foundation\Tomcat 5.0\common\lib\JsrWeb.jar

Options:

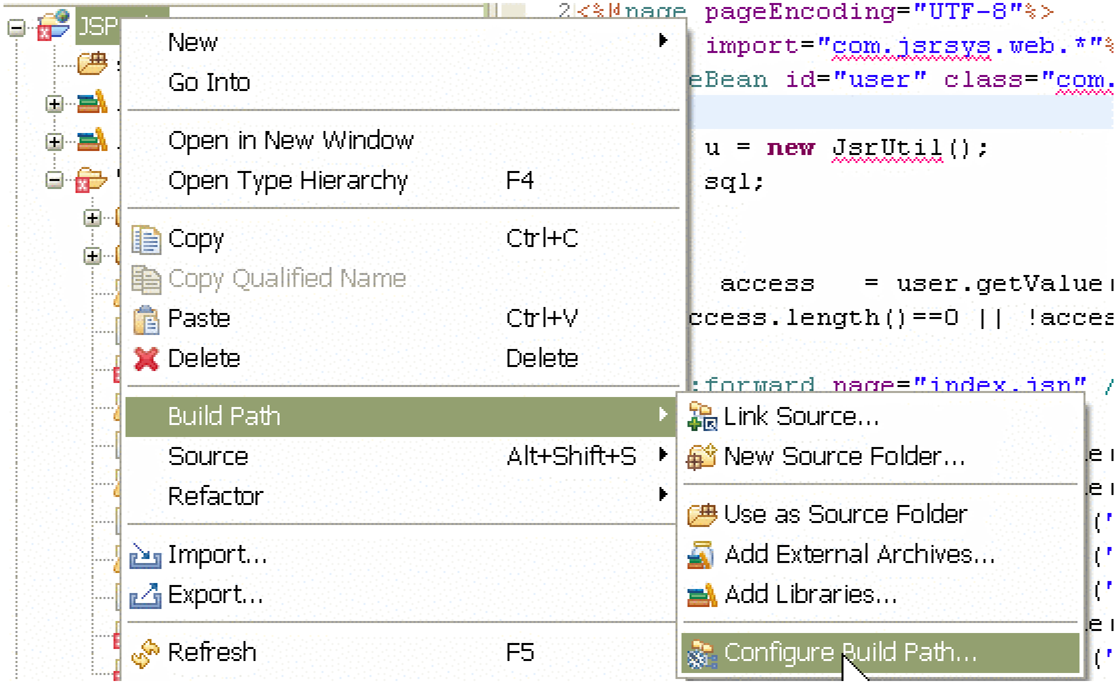
☑ Compress the contents of the JAR file
☐ Add directory entries
☐ Overwrite existing files without warning

Scroll right to be sure Jar file: … JsrWeb.jar then (Finish)

Note errors in JspLab on Login, SQLRep and ViewRep pages:



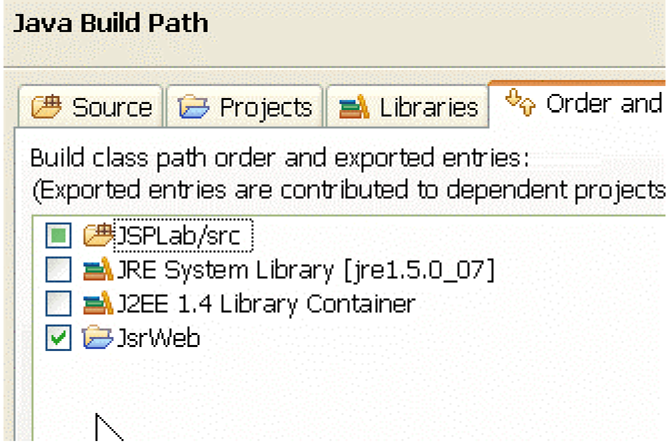
To fix: JspLab/BuildPath/Configure Build Path…



```
<%@page pageEncoding="UTF-8"%>
 import="com.jsrsys.web.*"%
eBean id="user" class="com.

 u = new JsrUtil();
 sql;

  access    = user.getValue
ccess.length()==0 || !acces

:forward page="index.jsp" /
```

...

And make sure JsrWeb is checked as a Source...



Build class path order and exported entries:
(Exported entries are contributed to dependent projects

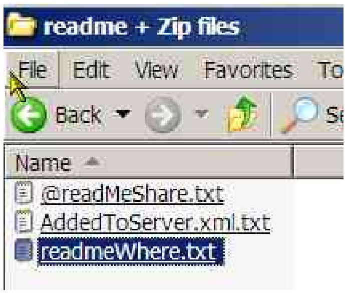
☑ JSPLab/src
☐ JRE System Library [jre1.5.0_07]
☐ J2EE 1.4 Library Container
☑ JsrWeb

Click OK

------------------------Before we start---------

Close any open Windows except JavaLabs and Eclipse

Navigate to JavaLabs. Click Readme+Zip files

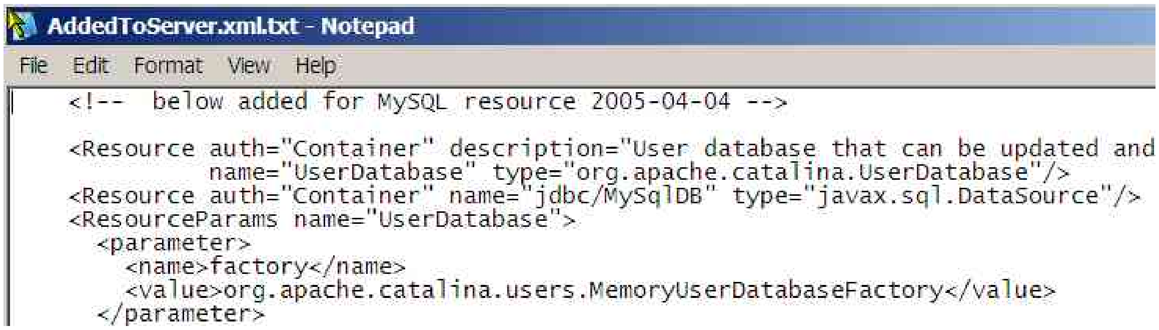


@readMeShare.txt
AddedToServer.xml.txt
readmeWhere.txt

Open @readMeShare.txt

These were all the steps needed to install the software for Java, Eclipse, MySQL, Tomcat and MyEclipse.

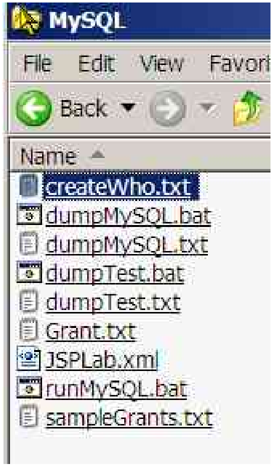
We are going to need to do two things so that JspLab pages can access the database. The following code has to be added to Server XML, Click AddtoServer.xml.txt.



```
<!--  below added for MySQL resource 2005-04-04 -->

<Resource auth="Container" description="User database that can be updated and
          name="UserDatabase" type="org.apache.catalina.UserDatabase"/>
<Resource auth="Container" name="jdbc/MySqlDB" type="javax.sql.DataSource"/>
<ResourceParams name="UserDatabase">
  <parameter>
    <name>factory</name>
    <value>org.apache.catalina.users.MemoryUserDatabaseFactory</value>
  </parameter>
```

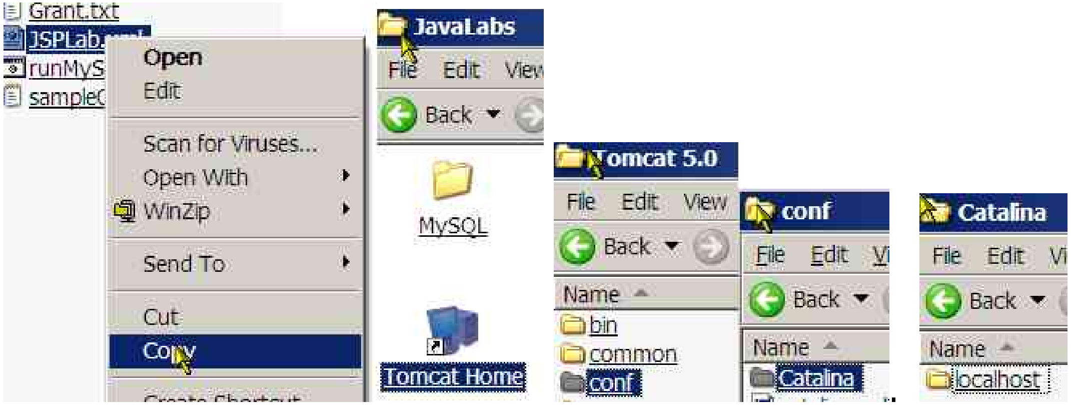
This has already been done...

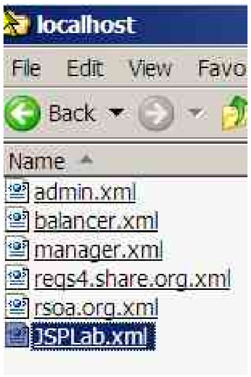
Now, navigate to the MySQL directory…Click Up, click MySQL.



We will be using these files later.

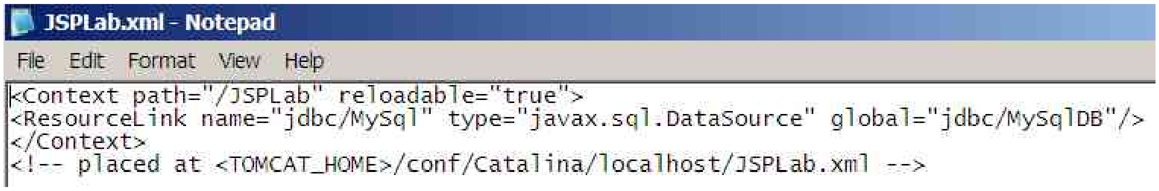
For now we need to copy the JSPLab.xml file to $TOMCAT_HOME\conf\Catalina\localhost.





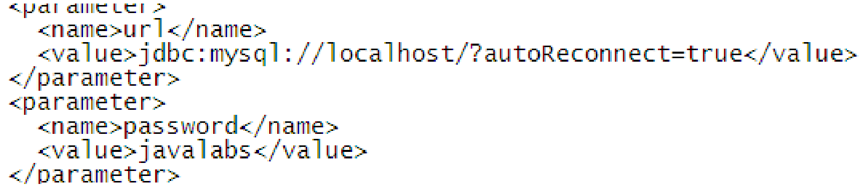
and paste JSPLab,xml into localhost

Open JSPLab.xml (right click/Edit) This allows programs in JSPLab to connect to the Pool defined in server XML.



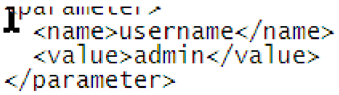
```
<Context path="/JSPLab" reloadable="true">
<ResourceLink name="jdbc/MySql" type="javax.sql.DataSource" global="jdbc/MySqlDB"/>
</Context>
<!-- placed at <TOMCAT_HOME>/conf/Catalina/localhost/JSPLab.xml -->
```

Close the localhost window. We now have to do the mysql grant for the username and password defined in server.xml



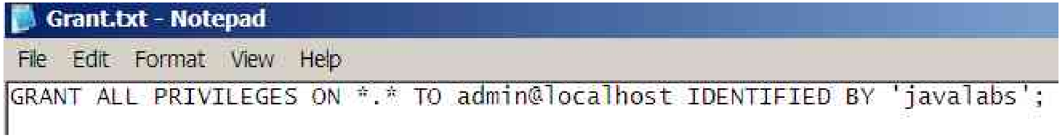
```
<parameter>
  <name>url</name>
  <value>jdbc:mysql://localhost/?autoReconnect=true</value>
</parameter>
<parameter>
  <name>password</name>
  <value>javalabs</value>
</parameter>
```

...



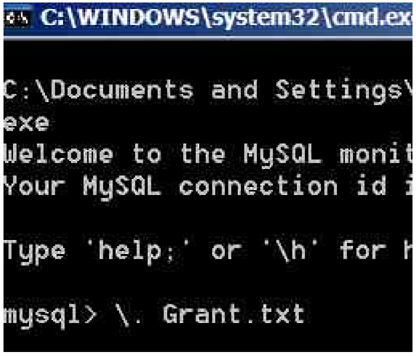
```
<parameter>
  <name>username</name>
  <value>admin</value>
</parameter>
```

This is found in the Grant.txt file in MySQL,



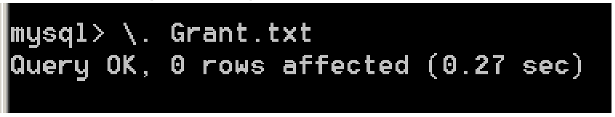
```
GRANT ALL PRIVILEGES ON *.* TO admin@localhost IDENTIFIED BY 'javalabs';
```

Click the RunMySQL.bat file and type \. Grant.txt



```
C:\Documents and Settings\
exe
Welcome to the MySQL monit
Your MySQL connection id i

Type 'help;' or '\h' for h

mysql> \. Grant.txt
```

Press enter to read the file. The \.space[Name of file] convention is allows you to keep a history of what you did. I strongly recommend this approach to typing into the command box. It greatly reduces the chance of “finger checks”.

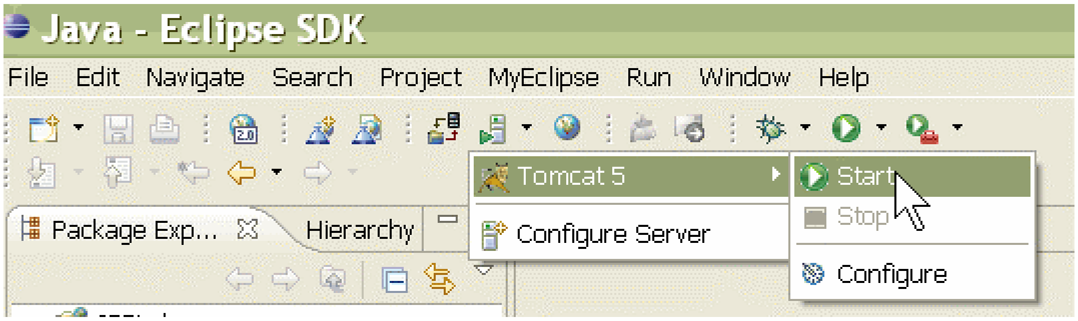


```
mysql> \. Grant.txt
Query OK, 0 rows affected (0.27 sec)
```

Now type quit to exit. Then press enter to exit the bat file.

We should now be ready to continue the exercise.

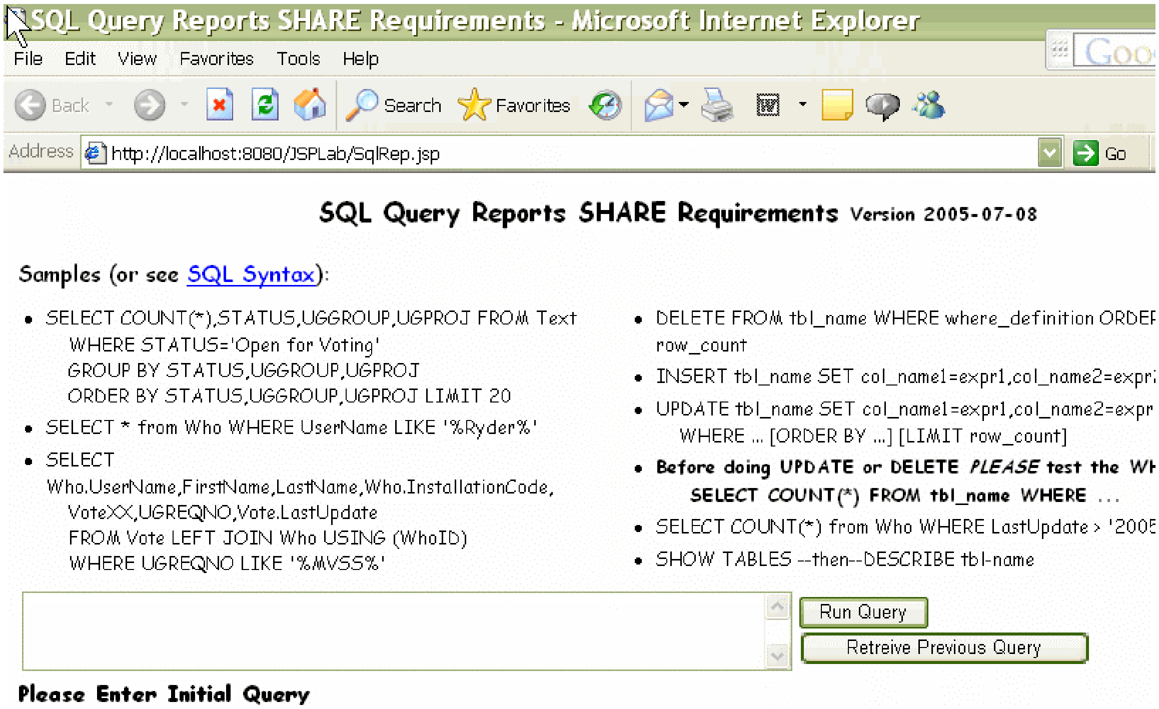
Only two windows should now be open: JavaLabs and MyEclipse.
Start Tomcat!



Clicking Tomcat IE in JavaLabs should open the index window

We are interested in the last two lines
SqlRep.jsp
ViewLog.jsp

Click the SQLRep line
You should see a page that starts "SQL Query Reports"



http://localhost:8080/JSPLab/SqlRep.jsp

SQL Query Reports SHARE Requirements Version 2005-07-08

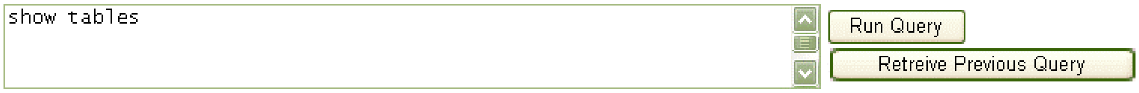
Samples (or see SQL Syntax):

- SELECT COUNT(*),STATUS,UGGROUP,UGPROJ FROM Text
  WHERE STATUS='Open for Voting'
  GROUP BY STATUS,UGGROUP,UGPROJ
  ORDER BY STATUS,UGGROUP,UGPROJ LIMIT 20
- SELECT * from Who WHERE UserName LIKE '%Ryder%'
- SELECT
  Who.UserName,FirstName,LastName,Who.InstallationCode,
  VoteXX,UGREQNO,Vote.LastUpdate
  FROM Vote LEFT JOIN Who USING (WhoID)
  WHERE UGREQNO LIKE '%MVSS%'
- DELETE FROM tbl_name WHERE where_definition ORDE
  row_count
- INSERT tbl_name SET col_name1=expr1,col_name2=expr
- UPDATE tbl_name SET col_name1=expr1,col_name2=expr
  WHERE ... [ORDER BY ...] [LIMIT row_count]
- **Before doing UPDATE or DELETE *PLEASE* test the W**
  SELECT COUNT(*) FROM tbl_name WHERE ...
- SELECT COUNT(*) from Who WHERE LastUpdate > '200
- SHOW TABLES --then--DESCRIBE tbl-name

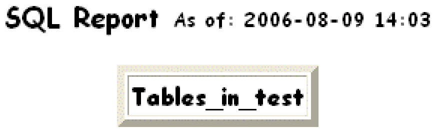
Please Enter Initial Query

If you do NOT, then ask for help.

Enter the following command "show tables" and click Run Query.



```
show tables
```

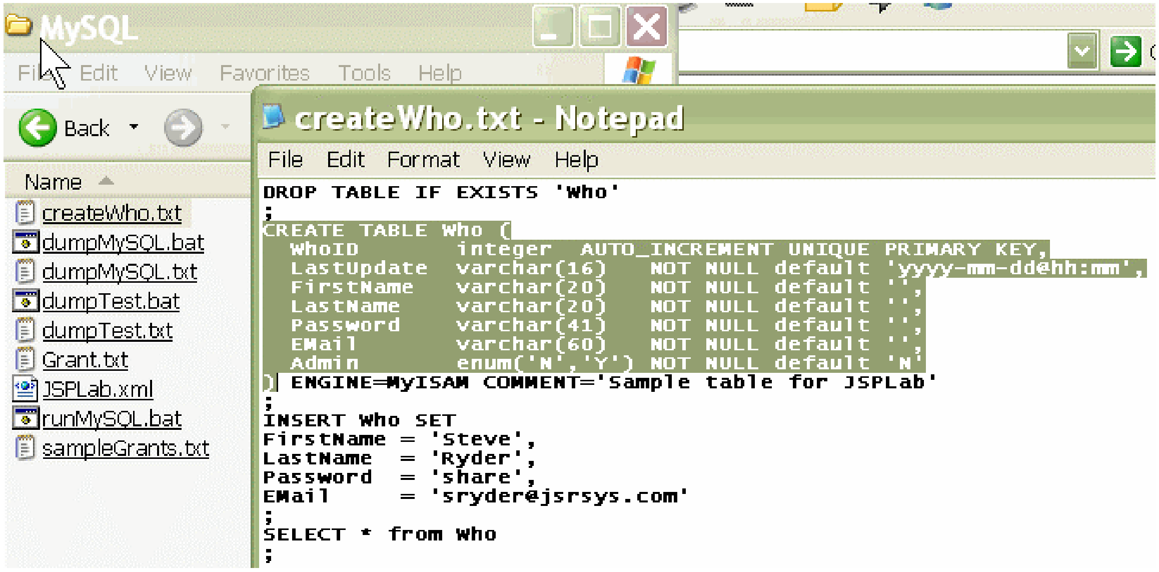


SQL Report As of: 2006-08-09 14:03

Tables_in_test

You should see a box with "**Tables_in_test**" because there are none.

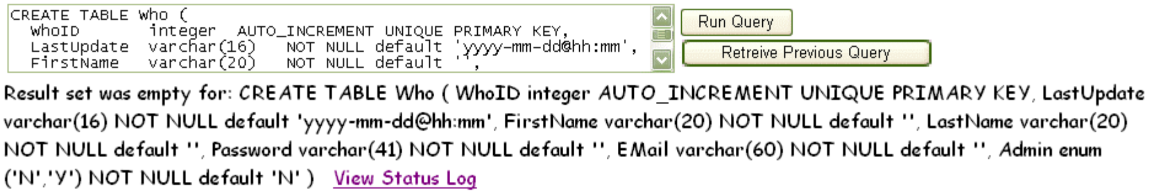
Navigate to the JavaLabs\MySQL directory and open createWho.txt file. Select the CREATE TABLE lines up to and including the last ")", but NOT the ENGINE=…



```
DROP TABLE IF EXISTS 'Who'
;
CREATE TABLE Who (
  WhoID        integer   AUTO_INCREMENT UNIQUE PRIMARY KEY,
  LastUpdate   varchar(16)    NOT NULL default 'yyyy-mm-dd@hh:mm',
  FirstName    varchar(20)    NOT NULL default '',
  LastName     varchar(20)    NOT NULL default '',
  Password     varchar(41)    NOT NULL default '',
  EMail        varchar(60)    NOT NULL default '',
  Admin        enum('N','Y') NOT NULL default 'N'
) ENGINE=MyISAM COMMENT='Sample table for JSPLab'
;
INSERT Who SET
FirstName  = 'Steve',
LastName   = 'Ryder',
Password   = 'share',
EMail      = 'sryder@jsrsys.com'
;
SELECT * from Who
;
```

Paste this into the text box of the SQL page and click Run Query.

You should see:

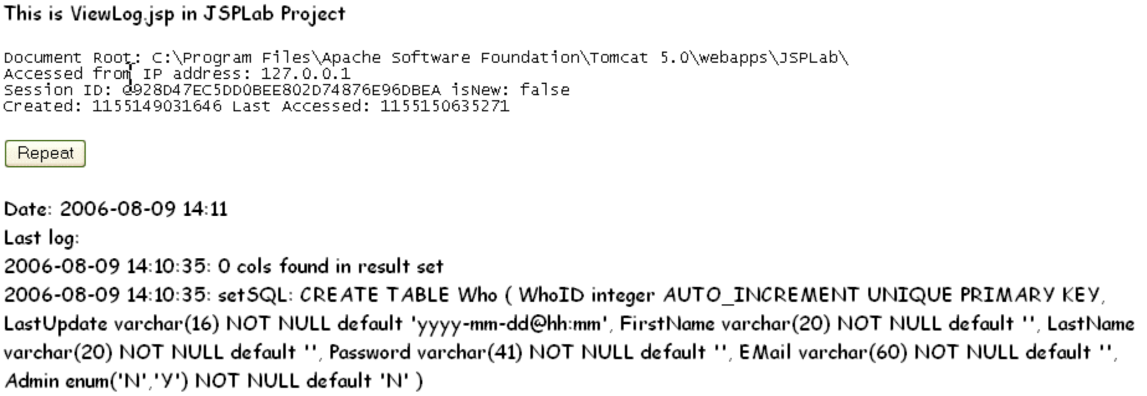


```
CREATE TABLE Who (
  WhoID        integer   AUTO_INCREMENT UNIQUE PRIMARY KEY,
  LastUpdate   varchar(16)    NOT NULL default 'yyyy-mm-dd@hh:mm',
  FirstName    varchar(20)    NOT NULL default '',
```

Result set was empty for: CREATE TABLE Who (WhoID integer AUTO_INCREMENT UNIQUE PRIMARY KEY, LastUpdate varchar(16) NOT NULL default 'yyyy-mm-dd@hh:mm', FirstName varchar(20) NOT NULL default '', LastName varchar(20) NOT NULL default '', Password varchar(41) NOT NULL default '', EMail varchar(60) NOT NULL default '', Admin enum ('N','Y') NOT NULL default 'N') View Status Log

This is not really an error and is something we will "fix" in the lab

Click the View Status Log line.



This is ViewLog.jsp in JSPLab Project

```
Document Root: C:\Program Files\Apache Software Foundation\Tomcat 5.0\webapps\JSPLab\
Accessed from IP address: 127.0.0.1
Session ID: C928D47EC5DD0BEE802D74876E96DBEA isNew: false
Created: 1155149031646 Last Accessed: 1155150635271
```

Date: 2006-08-09 14:11

Last log:

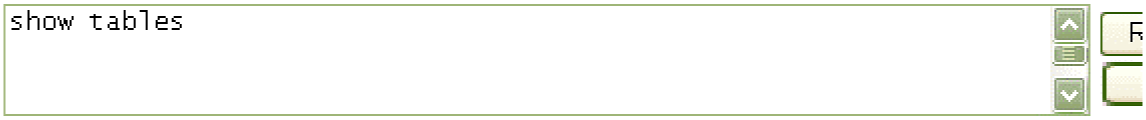
2006-08-09 14:10:35: 0 cols found in result set

2006-08-09 14:10:35: setSQL: CREATE TABLE Who (WhoID integer AUTO_INCREMENT UNIQUE PRIMARY KEY, LastUpdate varchar(16) NOT NULL default 'yyyy-mm-dd@hh:mm', FirstName varchar(20) NOT NULL default '', LastName varchar(20) NOT NULL default '', Password varchar(41) NOT NULL default '', EMail varchar(60) NOT NULL default '', Admin enum('N','Y') NOT NULL default 'N')

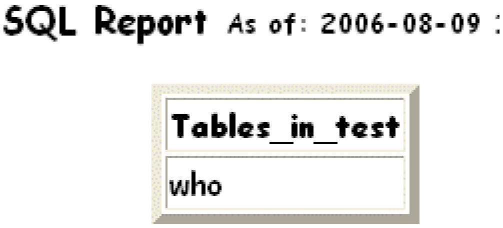
Note that even though the result set was 0, the command worked. The SQL jsp page makes the false assumption that all “queries” will return a value.

Close “ViewLog” which opened in a new window (or tab).

Enter the “show tables” command again and press Run Query



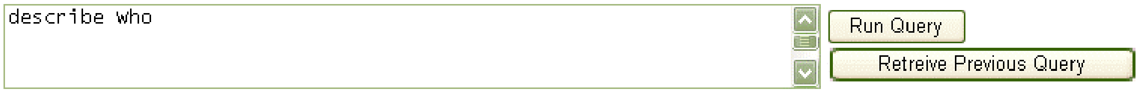
```
show tables
```



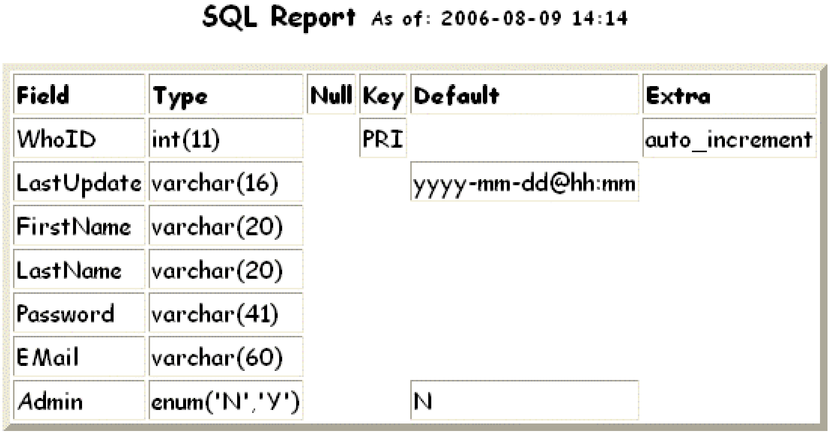
SQL Report As of: 2006-08-09

Tables_in_test
who

Enter “describe Who”, etc…



```
describe who
```

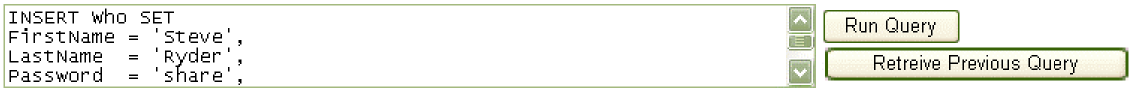


SQL Report As of: 2006-08-09 14:14

Field	Type	Null	Key	Default	Extra
WhoID	int(11)		PRI		auto_increment
LastUpdate	varchar(16)			yyyy-mm-dd@hh:mm	
FirstName	varchar(20)				
LastName	varchar(20)				
Password	varchar(41)				
EMail	varchar(60)				
Admin	enum('N','Y')			N	

You should see a table that describes Who.

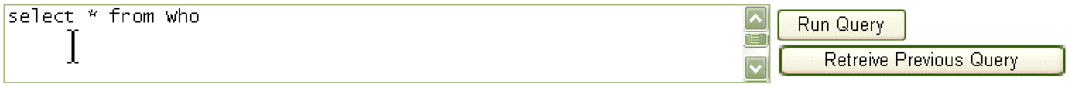
Now cut and paste the INSERT command from createWho.txt



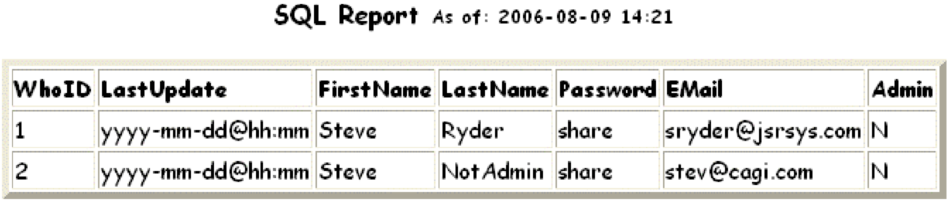
```
INSERT who SET
FirstName = 'Steve',
LastName  = 'Ryder',
Password  = 'share',
```

1 row(s) were updated by INSERT command.

Modify the values for the INSERT and insert a few more records, then enter "SELECT * from Who" or simply copy it from the "sample" commands.



```
select * from who
```

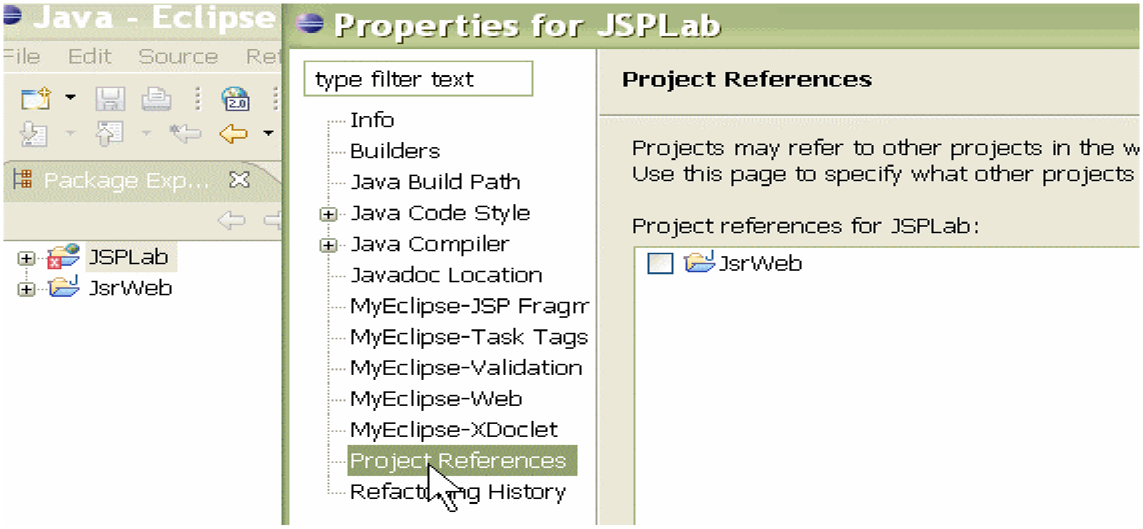


SQL Report As of: 2006-08-09 14:21

WhoID	LastUpdate	FirstName	LastName	Password	EMail	Admin
1	yyyy-mm-dd@hh:mm	Steve	Ryder	share	sryder@jsrsys.com	N
2	yyyy-mm-dd@hh:mm	Steve	NotAdmin	share	stev@cagi.com	N

Close the createWho.txt file, we are done with it.

Modify JSPLab properties (Right click, choose properties)
Then choose project references and select/check JsrWeb project and check OK.



Project References

Projects may refer to other projects in the w
Use this page to specify what other projects

Project references for JSPLab:

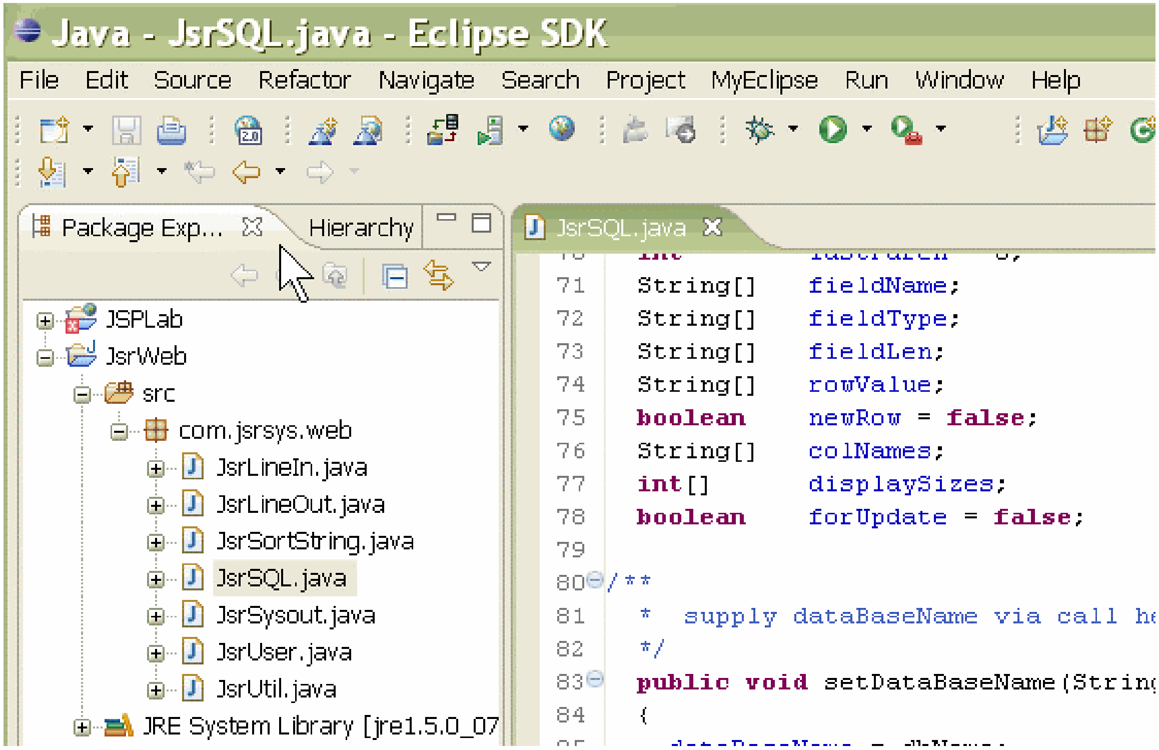
☐ JsrWeb

This will cause MyEclipse to modify the search path to use the project source for JsrSQL rather than the jsrweb.jar file that has been placed in common/lib.

STOP Tomcat

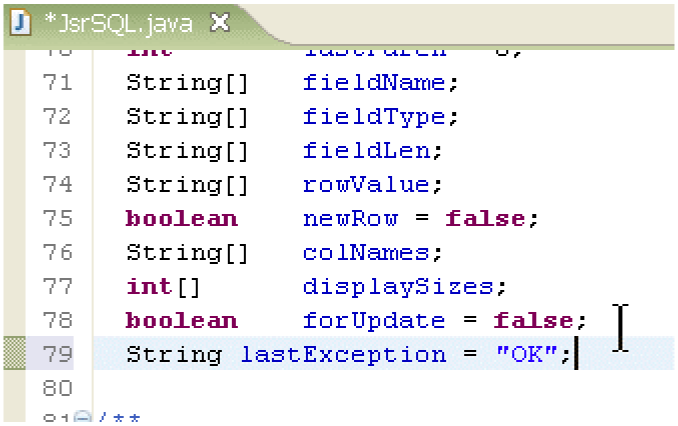
Modify JsrSQL program to set variable lastException to the contents of the exception if an exception occurs, or “OK” if no exception occurs. If you don’t see line numbers then click Window>Preference(+)General(+)Editor > Text Editor: check Line number on.

(+) JsrWeb (+) src Open JsrSQL



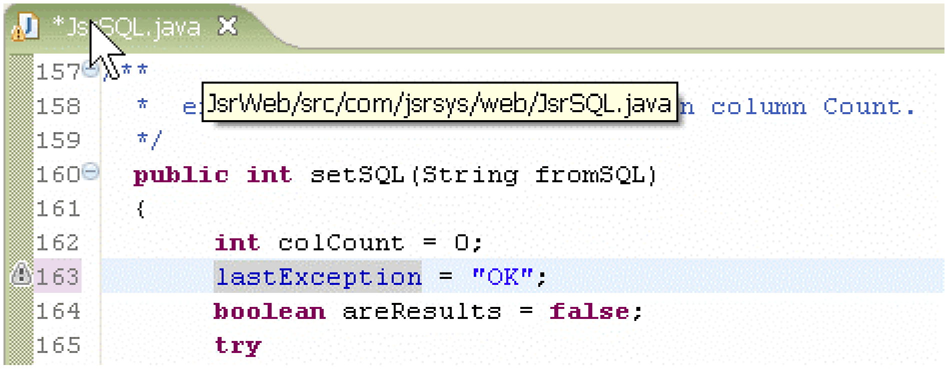
```
71   String[]    fieldName;
72   String[]    fieldType;
73   String[]    fieldLen;
74   String[]    rowValue;
75   boolean     newRow = false;
76   String[]    colNames;
77   int[]       displaySizes;
78   boolean     forUpdate = false;
79
80 /**
81   *  supply dataBaseName via call he
82   */
83   public void setDataBaseName(Strin
84   {
```

After line 78 as line 79 add
String lastException = “OK”;



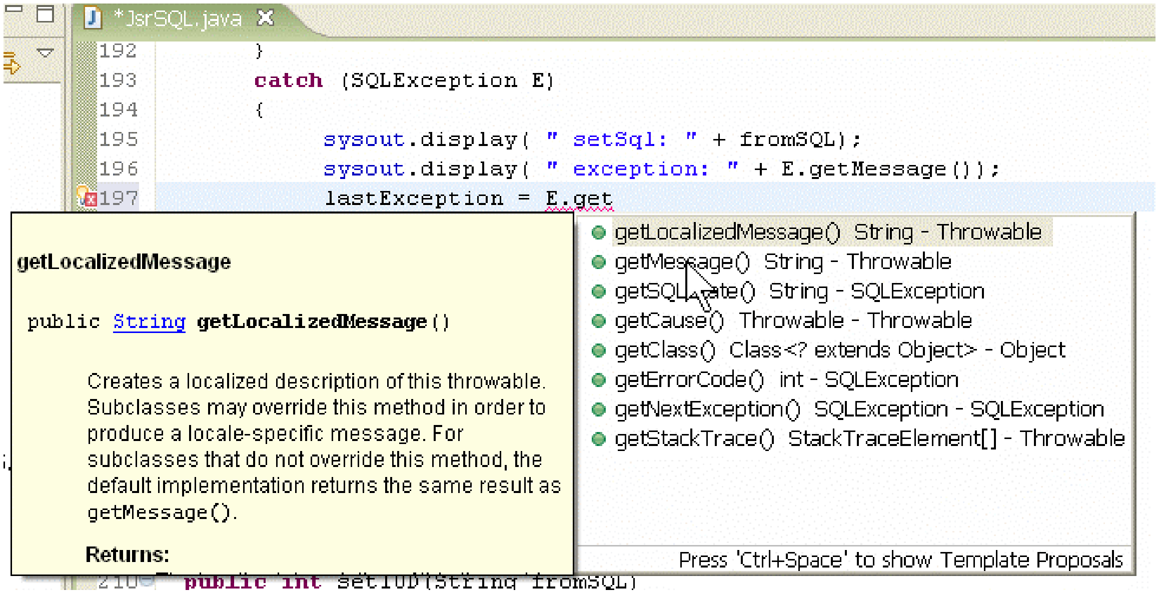
```
71      String[]    fieldName;
72      String[]    fieldType;
73      String[]    fieldLen;
74      String[]    rowValue;
75      boolean     newRow = false;
76      String[]    colNames;
77      int[]       displaySizes;
78      boolean     forUpdate = false;
79      String lastException = "OK";
80
```

In method setSQL after “int colCount=0” on line 162 insert lastException = “OK” as line 163



```
157  /**
158    * e                                    n column Count.
159    */
160  public int setSQL(String fromSQL)
161  {
162          int colCount = 0;
163          lastException = "OK";
164          boolean areResults = false;
165          try
```

After line 196 (the sysout.display of the exception) add lastException = E.getMessage();



```
192          }
193          catch (SQLException E)
194          {
195                  sysout.display( " setSql: " + fromSQL);
196                  sysout.display( " exception: " + E.getMessage());
197                  lastException = E.get
```

getLocalizedMessage

`public String getLocalizedMessage()`

Creates a localized description of this throwable. Subclasses may override this method in order to produce a locale-specific message. For subclasses that do not override this method, the default implementation returns the same result as `getMessage()`.

Returns:

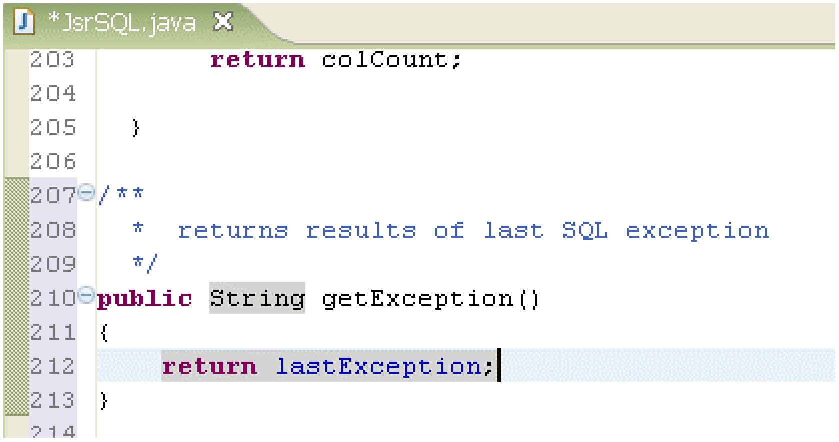
- getLocalizedMessage() String - Throwable
- getMessage() String - Throwable
- getSQLState() String - SQLException
- getCause() Throwable - Throwable
- getClass() Class<? extends Object> - Object
- getErrorCode() int - SQLException
- getNextException() SQLException - SQLException
- getStackTrace() StackTraceElement[] - Throwable

Press 'Ctrl+Space' to show Template Proposals

Note the popup HELP if we pause after entering just ...get

Add a method after line 207
called getException that returns lastException.

```
public String getException()
{
	return lastException;
}
```

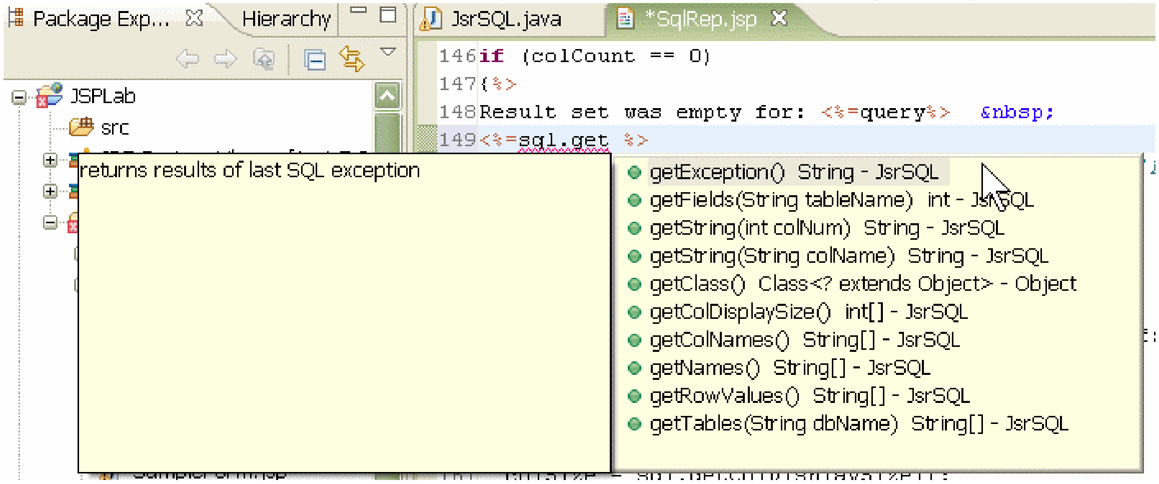


```
203         return colCount;
204
205    }
206
207/**
208    *  returns results of last SQL exception
209    */
210public String getException()
211 {
212       return lastException;
213 }
214
```

Save JsrSQL

Open SQLRep.jsp

After line 148 add “<br>Exception” <%=sql.getException()%>”.



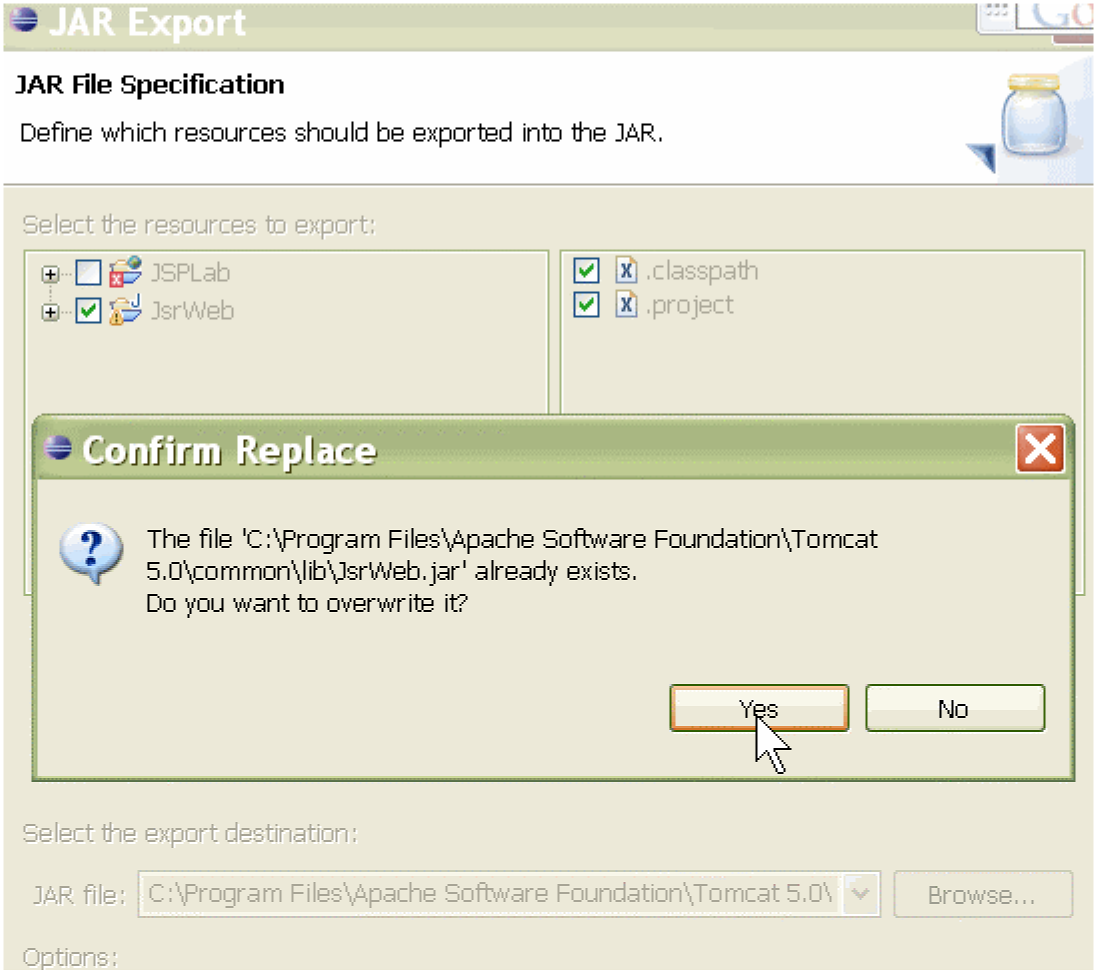
```
146if (colCount == 0)
147{%>
148Result set was empty for: <%=query%>  &nbsp;
149<%=sql.get %>
```

returns results of last SQL exception

- getException() String - JsrSQL
- getFields(String tableName) int - JsrSQL
- getString(int colNum) String - JsrSQL
- getString(String colName) String - JsrSQL
- getClass() Class<? extends Object> - Object
- getColDisplaySize() int[] - JsrSQL
- getColNames() String[] - JsrSQL
- getNames() String[] - JsrSQL
- getRowValues() String[] - JsrSQL
- getTables(String dbName) String[] - JsrSQL

Note that MyEclipse gets the popup from the source of the method we just added to JsrSQL!!!!

Because Tomcat uses the code from the jsrweb.jar file in common/lib, Export JsrWeb project to TOMCAT_HOME/common/lib/JsrWeb.jar. Note that Eclipse remembers all the previous paths.

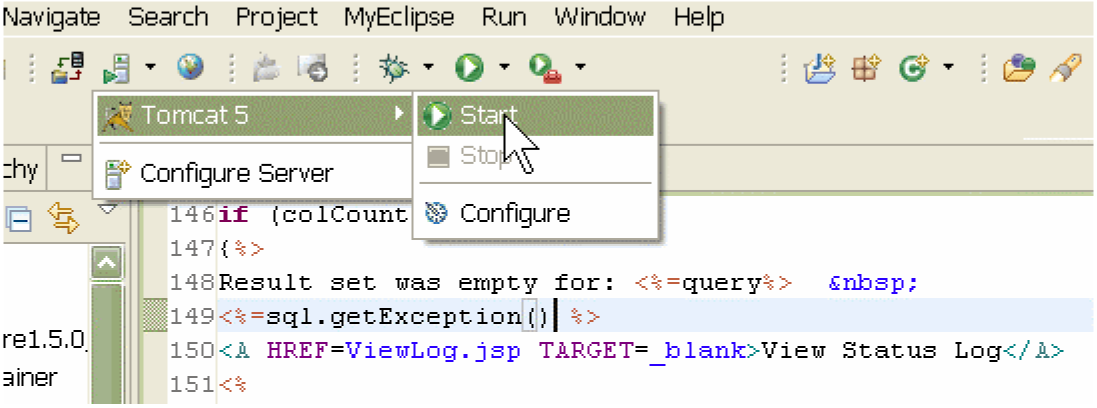


Define which resources should be exported into the JAR.

The file 'C:\Program Files\Apache Software Foundation\Tomcat 5.0\common\lib\JsrWeb.jar' already exists.
Do you want to overwrite it?

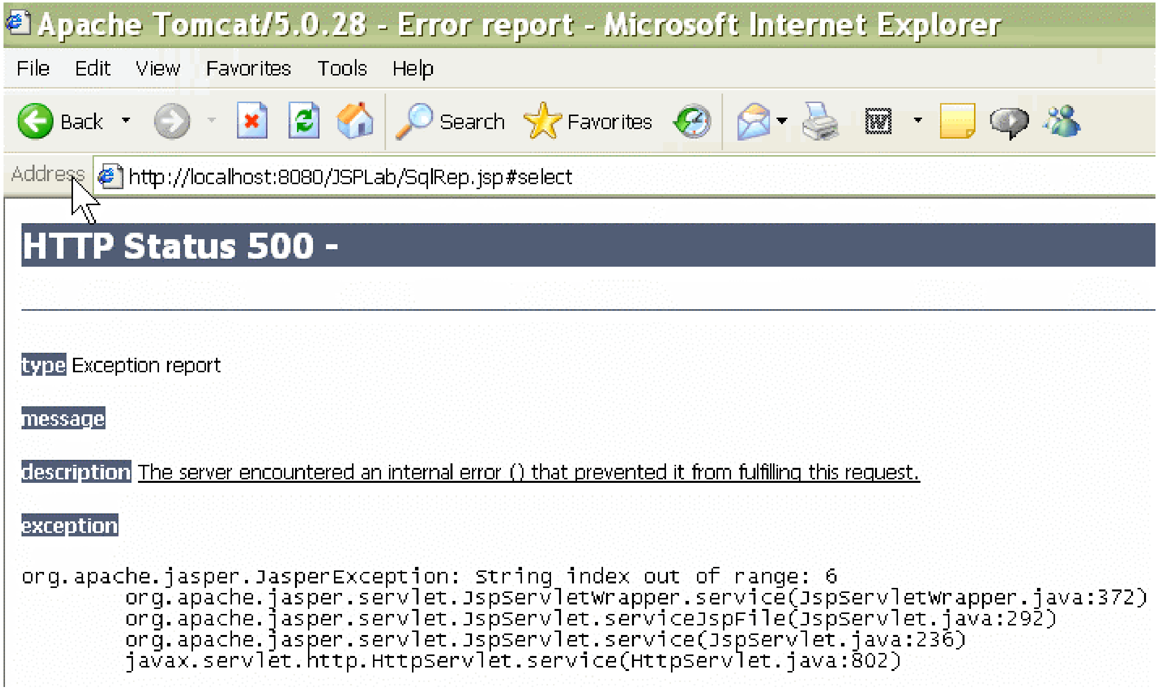
JAR file: C:\Program Files\Apache Software Foundation\Tomcat 5.0\

Then START Tomcat again.



```
if (colCount
{%>
Result set was empty for: <%=query%>  &nbsp;
<%=sql.getException() %>
<A HREF=ViewLog.jsp TARGET=_blank>View Status Log</A>
<%
```

Open up the SqlRep “window” again and enter some “bad” SQL. You should now see the “exception” message! However, try intering just “xxx” instead.


Apache Tomcat/5.0.28 - Error report - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Address

http://localhost:8080/JSPLab/SqlRep.jsp#select

HTTP Status 500 -

type Exception report

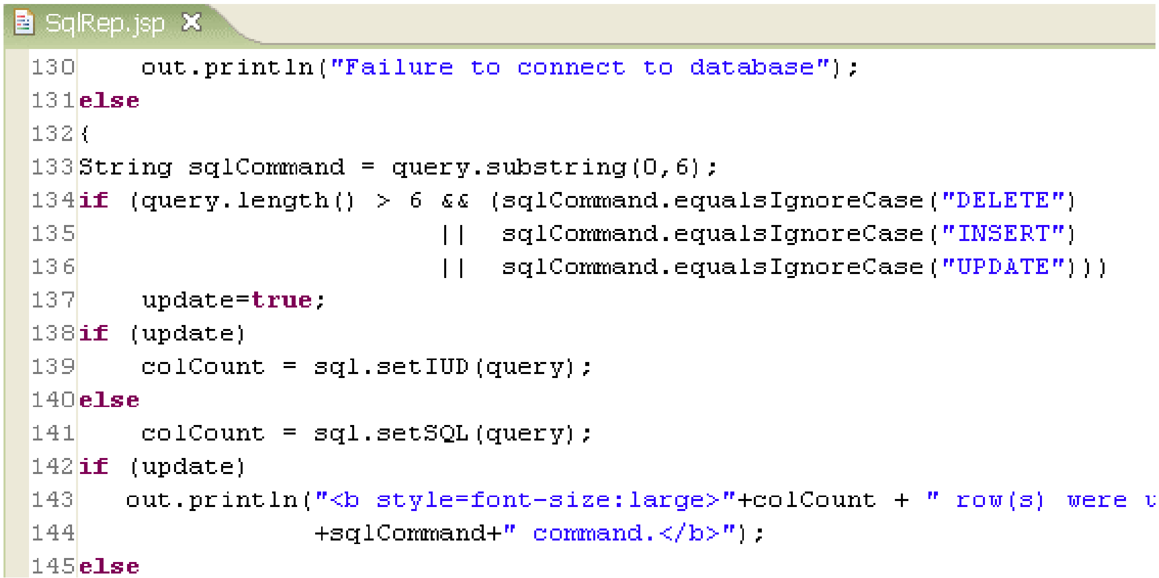
message

description The server encountered an internal error () that prevented it from fulfilling this request.

exception

```
org.apache.jasper.JasperException: String index out of range: 6
	org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:372)
	org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:292)
	org.apache.jasper.servlet.JspServlet.service(JspServlet.java:236)
	javax.servlet.http.HttpServlet.service(HttpServlet.java:802)
```

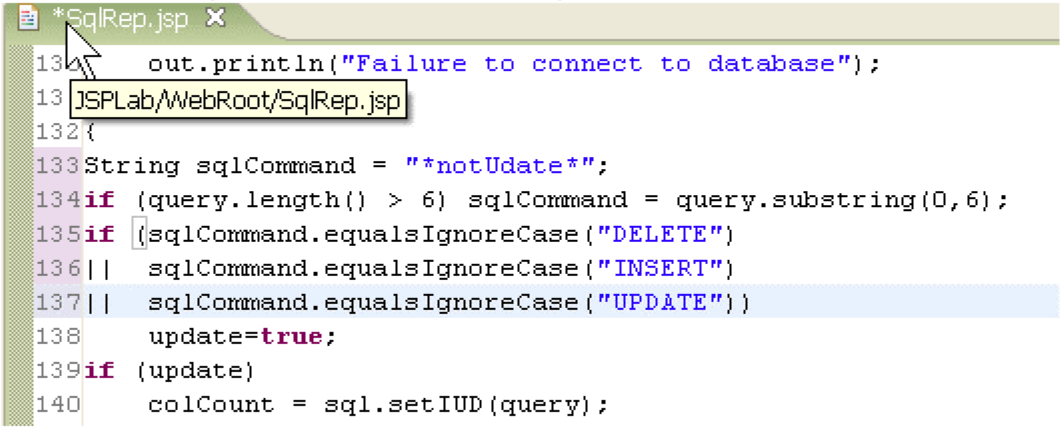
Ooops, this is a bug that was in my original code that I discovered during the last class (how embarrassing)!!!



```
    out.println("Failure to connect to database");
else
{
String sqlCommand = query.substring(0,6);
if (query.length() > 6 && (sqlCommand.equalsIgnoreCase("DELETE")
                       ||  sqlCommand.equalsIgnoreCase("INSERT")
                       ||  sqlCommand.equalsIgnoreCase("UPDATE")))
    update=true;
if (update)
    colCount = sql.setIUD(query);
else
    colCount = sql.setSQL(query);
if (update)
   out.println("<b style=font-size:large>"+colCount + " row(s) were u
               +sqlCommand+" command.</b>");
else
```

As you can see, I am doing the substring which assumes a length of at least six bytes BEFORE I check that the query length is > 6. This is a very common Java error. MyEclipse does have debugging facilities that tie this sort of error back to the line in question, but that is beyond the scope of this lab. Suffice it to say they work the same as regular Java debugging you covered in an earlier lab.

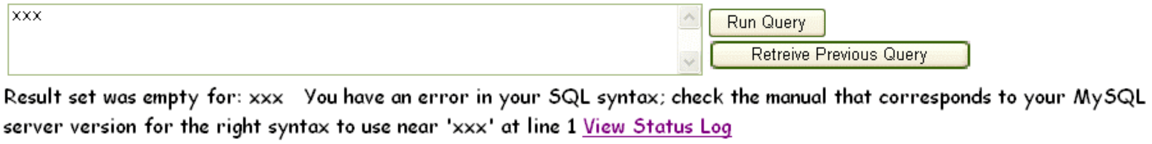
Here is one possible fix for the problem:



```
    out.println("Failure to connect to database");

{
String sqlCommand = "*notUdate*";
if (query.length() > 6) sqlCommand = query.substring(0,6);
if (sqlCommand.equalsIgnoreCase("DELETE")
||  sqlCommand.equalsIgnoreCase("INSERT")
||  sqlCommand.equalsIgnoreCase("UPDATE"))
    update=true;
if (update)
    colCount = sql.setIUD(query);
```

The purpose of this code was to call a different SQL method for update commands, which return the # of rows updated (rather than colCount (a bad reuse of a variable name). By the way, if you plan to use this example beyond the class you might want to add the lastException=E.getMessage() to the other methods as well. You might even want to do that as part of this lab. If you have a thumb drive, feel free to export the JSPLab and JsrWeb projects to it for your use at home.

Now when you enter a short string this is what you should see:



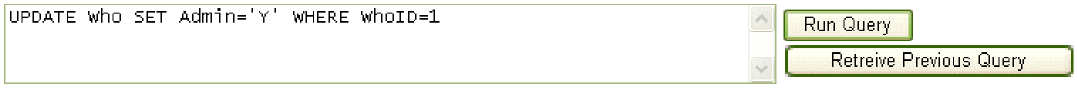
xxx

Result set was empty for: xxx You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'xxx' at line 1 View Status Log

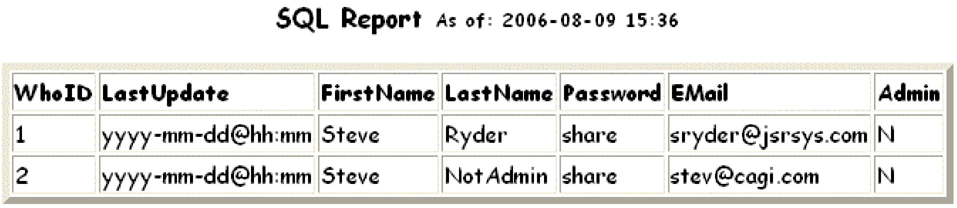
Note that the ViewLog.jsp class/page is also part of the example. Being an old COBOL programmer I know that having an error log can be very useful when something goes wrong so all my utility classes are laced with sysout.display… method invocations. The ViewLog.jsp page lets me see what is going on, even on the live system where I don't have MyEclipse debugging running.

Optional exercise:

Before doing this you will need to Update one of your “users” to Admin=’Y’.

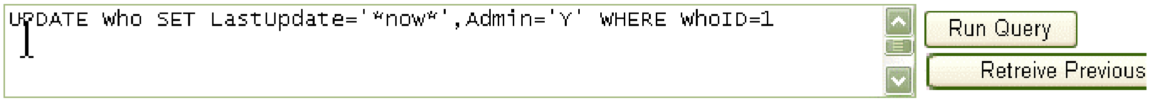


```
UPDATE who SET Admin='Y' WHERE whoID=1
```

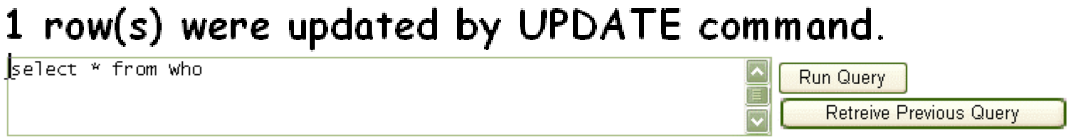


SQL Report As of: 2006-08-09 15:36

WhoID	LastUpdate	FirstName	LastName	Password	EMail	Admin
1	yyyy-mm-dd@hh:mm	Steve	Ryder	share	sryder@jsrsys.com	N
2	yyyy-mm-dd@hh:mm	Steve	NotAdmin	share	stev@cagi.com	N

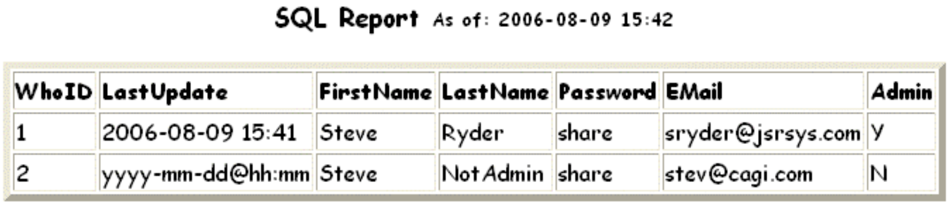


```
UPDATE who SET LastUpdate='*now*',Admin='Y' WHERE whoID=1
```



1 row(s) were updated by UPDATE command.

```
select * from who
```

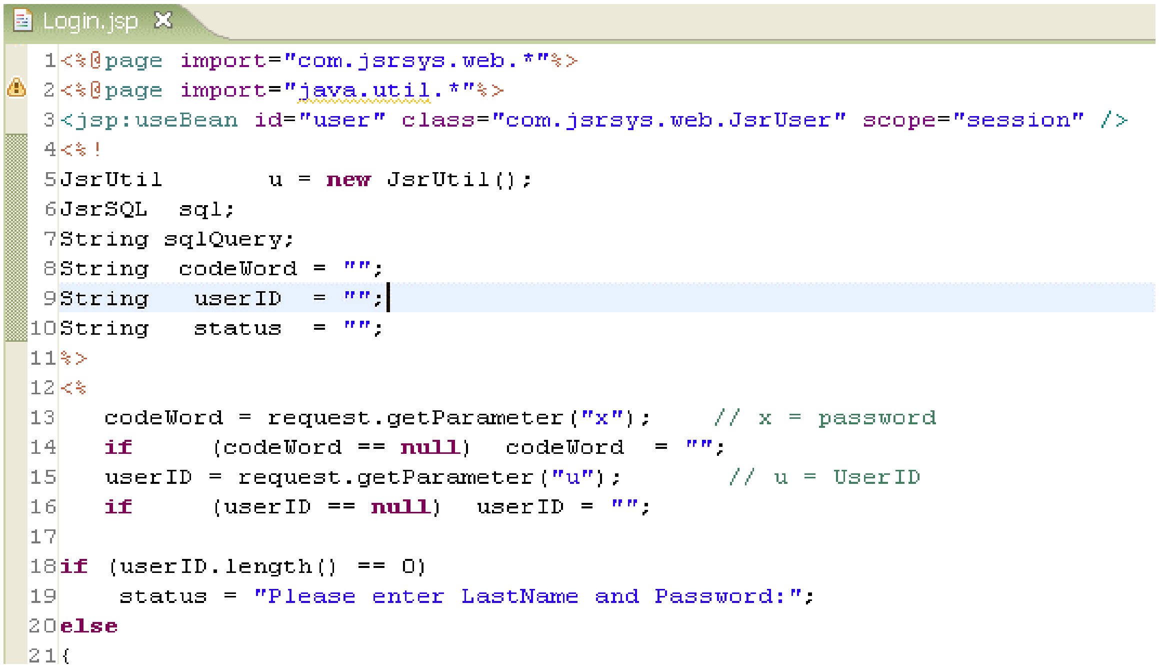


SQL Report As of: 2006-08-09 15:42

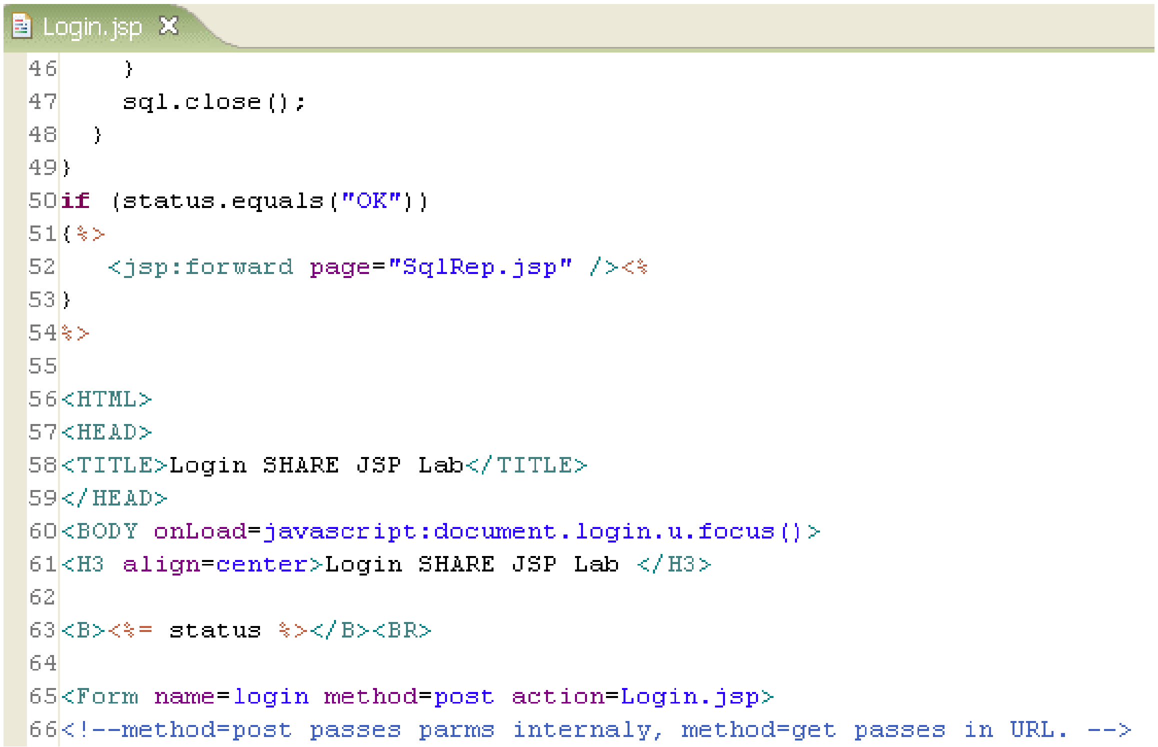
WhoID	LastUpdate	FirstName	LastName	Password	EMail	Admin
1	2006-08-09 15:41	Steve	Ryder	share	sryder@jsrsys.com	Y
2	yyyy-mm-dd@hh:mm	Steve	NotAdmin	share	stev@cagi.com	N

Note a “feature” of the JsrSQL class. JsrSQL will modify any input string of “*now*” to be the current date and time. This is not a feature of either JSP or the Java MySQL connector, but rather a value-add from my code.

Now do “Open with” Login.jsp to view a page that stores login information in the session persistent object JsrUser used by SqlRep to save prior queries

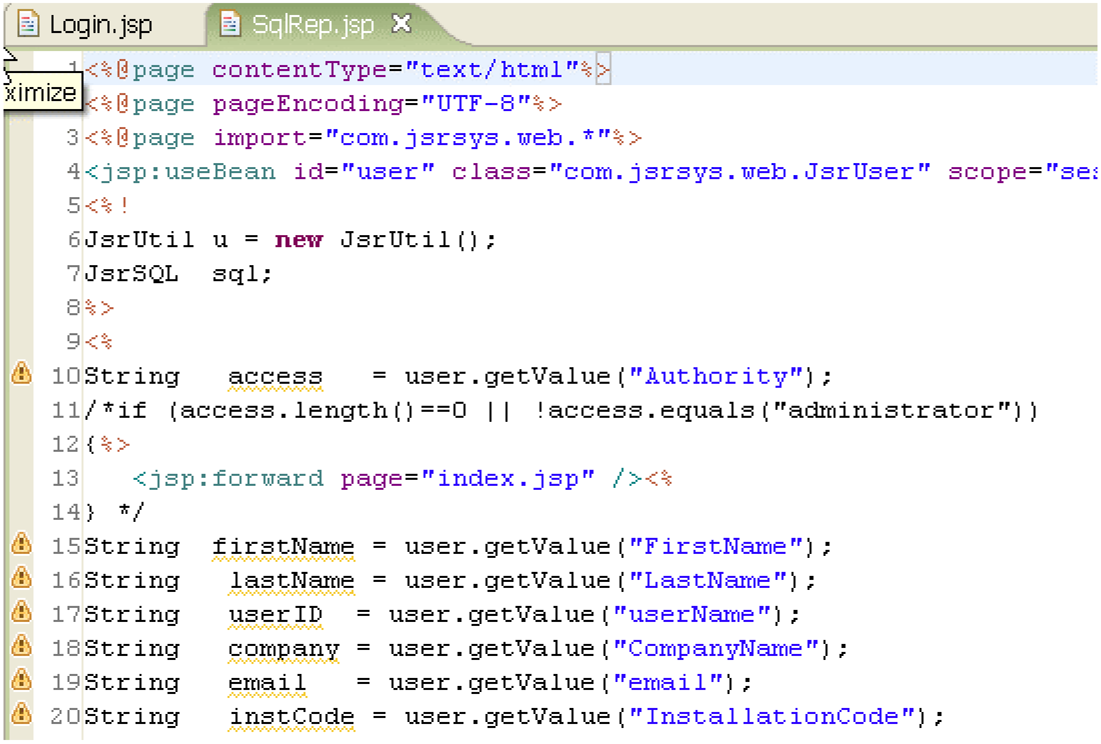


```
<%@page import="com.jsrsys.web.*"%>
<%@page import="java.util.*"%>
<jsp:useBean id="user" class="com.jsrsys.web.JsrUser" scope="session" />
<%!
JsrUtil       u = new JsrUtil();
JsrSQL  sql;
String sqlQuery;
String  codeWord = "";
String   userID  = "";
String   status  = "";
%>
<%
   codeWord = request.getParameter("x");    // x = password
   if      (codeWord == null)  codeWord  = "";
   userID = request.getParameter("u");        // u = UserID
   if      (userID == null)  userID = "";

if (userID.length() == 0)
    status = "Please enter LastName and Password:";
else
{
```



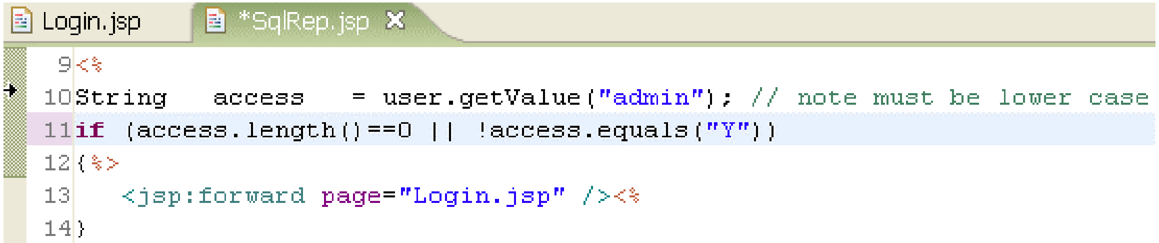
```
    }
    sql.close();
  }
}
if (status.equals("OK"))
{%>
   <jsp:forward page="SqlRep.jsp" /><%
}
%>

<HTML>
<HEAD>
<TITLE>Login SHARE JSP Lab</TITLE>
</HEAD>
<BODY onLoad=javascript:document.login.u.focus()>
<H3 align=center>Login SHARE JSP Lab </H3>

<B><%= status %></B><BR>

<Form name=login method=post action=Login.jsp>
<!--method=post passes parms internaly, method=get passes in URL. -->
```

SqlRep can also use this object to check the security attributes of the currently “logged in” user. This code is disabled in the example.



```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@page import="com.jsrsys.web.*"%>
<jsp:useBean id="user" class="com.jsrsys.web.JsrUser" scope="se
<%!
JsrUtil u = new JsrUtil();
JsrSQL  sql;
%>
<%
String   access   = user.getValue("Authority");
/*if (access.length()==0 || !access.equals("administrator"))
{%>
   <jsp:forward page="index.jsp" /><%
} */
String  firstName = user.getValue("FirstName");
String   lastName = user.getValue("LastName");
String   userID  = user.getValue("userName");
String   company = user.getValue("CompanyName");
String   email   = user.getValue("email");
String   instCode = user.getValue("InstallationCode");
```

Modify SqlRep.jsp to forward to Login.jsp if admin not = ‘Y’



```
<%
String   access   = user.getValue("admin"); // note must be lower case
if (access.length()==0 || !access.equals("Y"))
{%>
   <jsp:forward page="Login.jsp" /><%
}
```

Note: since making the slides I noticed that if a user is signed in but their admin value is “N”, Login page fails to display. I don’t know why this is, but I “fixed” it by changing the forward to page=”Login.jsp?x=” “.

Now try SqlRep again from the menu browser page. You should see the Login page where you can enter the LastName of the person to who you assigned the Y value for admin. But first, try entering a name of someone with a value of N. You should get bumped right back to the Login.jsp screen. Then try the one you set to Y and you should see the SqlRep screen.

If you get this far, then you win the prize and I will buy you a drink tonight at SCIDS!

...

...