

SERVLET JNDI DATASOURCE IN TOMCAT

14 May 2012 By Nithya Vasudevan 13,461 views 9 Comments

3
SHARES | 1 2 0

1. Environment Used

- JDK 6 (Java SE 6) (To install JDK – [Windows](#) , [Ubuntu](#))
- Eclipse Indigo IDE for Java EE Developers (3.7.1) (To install Eclipse, [refer this page](#))
- Apache Tomcat 6.x (To install Tomcat [refer this page](#))
- MySQL 5.5 (To install MySQL [refer this page](#))
- MySQL Connector/J 5.1 JAR file
- Java EE 5 API (Servlet 2.5)
- Java Database Connectivity (JDBC) API (java.sql.*, javax.sql.*)
- [Optional] For monitoring and analyzing HTTP headers between the browser and web servers, you can use one of these add-ons of Firefox
 - Live HTTP Headers
 - HttpFox

2. Setting up development environment

If you are new to developing Servlet with Tomcat and Eclipse, you can [read this page](#) before proceeding with this example.

3. Project Description

- This example explains how to configure datasource in Apache Tomcat and look it up in Servlet.
- This application uses one table 'Employee' and displays employee details to the user.
- We use Eclipse IDE for Java EE Developers and Apache Tomcat to run the Servlet.

4. What is javax.sql.DataSource?

- **DataSource** is a Java object which represents a **factory of connections** to the physical data source. DataSource is preferred over DriverManager facility.
- The DataSource interface is implemented by a driver vendor and will typically be **registered with a naming service** based on the Java Naming and Directory (JNDI) API.
- **DataSource is usually configured and managed by the application server** instead of your application. DataSource object is retrieved through a lookup operation using JNDI API and then used to create a Connection object.
- Advantage of DataSource:
 - A DataSource object has properties that can be modified when necessary. If the data source is moved to a different server, the property for the server can be changed and any code accessing that data source does not need to be changed.


5. Create Database and table in MySQL

This example uses one table 'Employee' and the description of the table is shown below.

| Field | Type | Key | Extra |
|-----------|--------------|-------------|----------------|
| emp_id | int | Primary Key | auto_increment |
| emp_name | varchar(255) | | |
| salary | double | | |
| dept_name | varchar(255) | | |

Table of Contents


- [1 Environment Used](#)
- [2 Setting up development environment](#)
- [3 Project Description](#)
- [4 What is javax.sql.DataSource?](#)
- [5 Create Database and table in MySQL](#)
- [6 Create Dynamic Web Project](#)
 - [6.1 Download MySQL connector](#)
 - [6.2 Configure Context](#)
 - [6.3 Create Servlet](#)
 - [6.4 Configure web.xml](#)
 - [6.5 Folder Structure](#)
- [7 Output](#)
- [8 Program control flow](#)



The Open Tutorials


Like

2,374 people like The Open Tutorials.



Facebook social plugin

The Open Tutorials



Follow

+1

+ 293

Popular Posts

- [Pagination in Servlet and JSP](#)
- [How to create a simple EJB3 project in Eclipse \(JBoss 7.1\)](#)
- [How to create a Servlet with Eclipse and Tomcat](#)
- [Android: Custom ListView with Image and Text using ArrayAdapter](#)
- [How to create and consume a simple Web Service using JAX WS](#)
- [Generate Java class from XML Schema using JAXB 'xjc' command](#)
- [Android: Expandable List View Example](#)
- [How to create a simple Restful Web Service using Jersey JAX RS API](#)
- [How to create EJB3 JPA Project in Eclipse \(JBoss AS 7.1\)](#)
- [How to configure Apache Tomcat in Eclipse IDE?](#)

Recent Comments

- Thirumala Reddy on [How to create a simple Restful Web Service using Jersey JAX RS API](#)
- Eclipse Apache Tomcat Kurulumu | Recep Dur on [How to configure Apache Tomcat in Eclipse IDE?](#)

- Open command prompt (Windows) or Terminal(Linux) and type
mysql -u [your-username] -p
and press enter and type the password.
- If you are using Windows, you can also use MySQL command line client which will be available in All programs menu.
- For creating a new database, [refer this page](#). In this example, the database name is 'exampledb'.
- After creating the database type the command "use <database_name>;"
- For creating a new table, [refer this page](#). In this example, the table name is 'employee'
- Insert some records in the table. [Refer this page](#) for inserting rows using MySQL.

```

MySQL 5.5 Command Line Client
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 133
Server version: 5.5.15 MySQL Community Server <GPL>

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use exampledb;
Database changed
mysql> desc employee;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emp_id | int(11) | NO | PRI | NULL | auto_increment |
| emp_name | varchar(255) | NO | | NULL | |
| salary | double | NO | | NULL | |
| dept_name | varchar(255) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql> select * from employee;
+-----+-----+-----+-----+
| emp_id | emp_name | salary | dept_name |
+-----+-----+-----+-----+
| 1 | John | 10000 | Development |
| 2 | Peter | 15000 | Testing |
| 3 | Abraham | 20000 | Administration |
| 4 | Ram | 35000 | Marketing |
| 5 | Diana | 40000 | Financial |
| 6 | Raj | 35000 | QA |
| 7 | Jennifer | 20000 | Sales |
| 8 | Jansi | 30000 | Purchase |
| 9 | Jenni | 60000 | Services |
| 10 | Sanjay | 35000 | Inventory |
| 11 | Yuvh | 32000 | Training |
| 12 | Jai | 38000 | Operational |
| 13 | Sita | 52000 | Legal |
| 14 | Vinay | 34000 | Insurance |
+-----+-----+-----+-----+
14 rows in set (0.00 sec)

mysql>

```

Wayne on [Android: Simple XML SAX Parser Tutorial](#)

Thirumala Reddy on [How to create and consume a simple Web Service using JAX WS](#)

Thirumala Reddy on [How to create and consume a simple Web Service using JAX WS](#)

Naveen kant Mishra on [Android: Custom ListView with Image and Text using ArrayAdapter](#)

maruti kutre on [How to create a simple Restful Web Service using Jersey JAX RS API](#)
[Android: RuntimeException: Unable to start activity ComponentInfoCopyQuery](#)
[CopyQuery | Question & Answer Tool for your Technical Queries,CopyQuery, ejjuat, query, copyquery, copyquery.com, android doubt, ios question, sql query, sqllite query, nodejsqu](#)
[on Android: Expandable List View Example](#)

Stir-Fried Code on [Android: How to load Image from URL in ImageView?](#)

Ravindra Chothe on [Android: Custom ListView with Image and Text using ArrayAdapter](#)

6. Create Dynamic Web Project

- Open Eclipse IDE
- For writing Servlet and JSP, we need to create a new **Dynamic Web project**. Create this project and name it as "ServletDataSource".

6.1. Download MySQL connector

- The connector can be downloaded from: <http://dev.mysql.com/downloads/connector/j/>.
- This tutorial uses 5.1 version. Unzip the connector to a safe location on your computer which contains MySQL Connector J JAR.
- Copy the MySQL Connector J JAR and paste it inside project's 'lib' folder.

6.2. Configure Context

- Configure the JNDI DataSource in Tomcat by adding a declaration for your resource to your Context in context.xml.
- This xml file contains context information and can be placed in one of the location as explained below,
 - In the /conf/context.xml file
 - In the /conf/[enginename]/[hostname]/context.xml.
 - META-INF/context.xml inside the application.

We will place the context.xml file in application's META-INF folder.

Create a new XML file in META-INF folder and copy the following content.

```

01 <?xml version="1.0" encoding="UTF-8"?>
02 <!-- The contents of this file will be loaded for each web application -->
03 <Context crossContext="true">
04
05     <!-- Default set of monitored resources -->
06     <WatchedResource>WEB-INF/web.xml</WatchedResource>
07
08     <Resource name="jdbc/testDB" auth="Container"

```

```

09         type="javax.sql.DataSource"
10         maxActive="100" maxIdle="30" maxWait="10000"
11         username="<YOUR_DATABASE_USERNAME>"
12         password="<YOUR_DATABASE_PASSWORD>"
13         driverClassName="com.mysql.jdbc.Driver"
14         url="jdbc:mysql://localhost/exampledb"/>
15     </Context>

```

Fill the username and password for your database and enter your database name in the url string.

6.3. Create Servlet

Here, we lookup the JDBC resource and store it in DataSource object. Using this object we create Connection object. We are creating SQL Statement and executing the query and printing the result.

```

01 package com.theopentutorials.servlets;
02
03 import java.io.IOException;
04 import java.io.PrintWriter;
05 import java.sql.Connection;
06 import java.sql.ResultSet;
07 import java.sql.SQLException;
08 import java.sql.Statement;
09 import javax.naming.Context;
10 import javax.naming.InitialContext;
11 import javax.naming.NamingException;
12 import javax.servlet.ServletException;
13 import javax.servlet.http.HttpServlet;
14 import javax.servlet.http.HttpServletRequest;
15 import javax.servlet.http.HttpServletResponse;
16 import javax.sql.DataSource;
17
18 public class EmployeeServlet extends HttpServlet {
19     private static final long serialVersionUID = 1L;
20     public EmployeeServlet() {}
21
22     public void doGet(HttpServletRequest request,
23         HttpServletResponse response)
24         throws ServletException, IOException {
25
26         Context envContext = null;
27         try {
28             envContext = new InitialContext();
29             Context initContext = (Context)envContext.lookup("java:/comp/env/");
30             DataSource ds = (DataSource)initContext.lookup("jdbc/testDB");
31             //DataSource ds = (DataSource)envContext.lookup("java:/comp/env/");
32             Connection con = ds.getConnection();
33
34             Statement stmt = con.createStatement();
35             String query = "select * from employee";
36             ResultSet rs = stmt.executeQuery(query);
37
38             PrintWriter out = response.getWriter();
39             response.setContentType("text/html");
40             out.print("<center><h1>Employee Details</h1></center>");
41             out.print("<html><body><table border='1' cellspacing=10 cellpadding=10>");
42             out.print("<th>Employee ID</th>");
43             out.print("<th>Employee Name</th>");
44             out.print("<th>Salary</th>");
45             out.print("<th>Department</th>");
46
47             while(rs.next())
48             {
49                 out.print("<tr>");
50                 out.print("<td>" + rs.getInt("emp_id") + "</td>");
51                 out.print("<td>" + rs.getString("emp_name") + "</td>");
52                 out.print("<td>" + rs.getDouble("salary") + "</td>");
53                 out.print("<td>" + rs.getString("dept_name") + "</td>");
54                 out.print("</tr>");
55             }
56             out.print("</table></body></html>");
57         } catch (SQLException e) {
58             e.printStackTrace();
59         } catch (NamingException e) {
60             e.printStackTrace();
61         }
62     }
63 }

```

- There is a global JNDI namespace, and it uses names that are vendor dependent, possibly not portable. Every EJB, JDBC pool, JMS Topic or Queue, you create in the server, has its name in the global JNDI.
- "java:comp/env" is a utility context and every resource (for example, JDBC, EJB, etc) has its own java:comp/env context populated with resources such as datasources, ejbs, etc.

- To lookup JDBC resources in that global namespace, we use “jdbc/” as defined in name attribute of element in context.xml. In our case, it is “jdbc/testDB”.

Instead of doing the lookup twice (one for global JNDI namespace and another for JDBC resource) as shown above, we can do a single lookup to the JDBC resource in the namespace as shown below.

```
Context envContext = new InitialContext();
DataSource ds = (DataSource)envContext.lookup("java:/comp/env/jdbc/testDB");
```

6.4. Configure web.xml

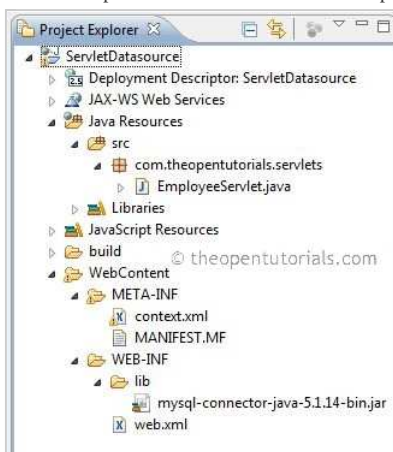
A resource reference is an element in a deployment descriptor that identifies the component's coded name for the resource. More specifically, the coded name references a connection factory for the resource. In the example in the following section, the resource reference name is “jdbc/testDB”.

Now create a WEB-INF/web.xml for this application.

```
01 <?xml version="1.0" encoding="UTF-8"?>
02 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
03         xmlns="http://java.sun.com/xml/ns/javaee"
04         xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
05         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.co
06         /web-app_2_5.xsd" id="WebApp_ID" version="2.5">
07     <resource-ref>
08         <description>DB Connection</description>
09         <res-ref-name>jdbc/testDB</res-ref-name>
10         <res-type>javax.sql.DataSource</res-type>
11         <res-auth>Container</res-auth>
12     </resource-ref>
13     <servlet>
14         <servlet-name>EmployeeServlet</servlet-name>
15         <servlet-class>com.theopentutorials.servlets.EmployeeServlet</servlet-cl
16     </servlet>
17     <servlet-mapping>
18         <servlet-name>EmployeeServlet</servlet-name>
19         <url-pattern>/employee.do</url-pattern>
20     </servlet-mapping>
21 </web-app>
```

6.5. Folder Structure

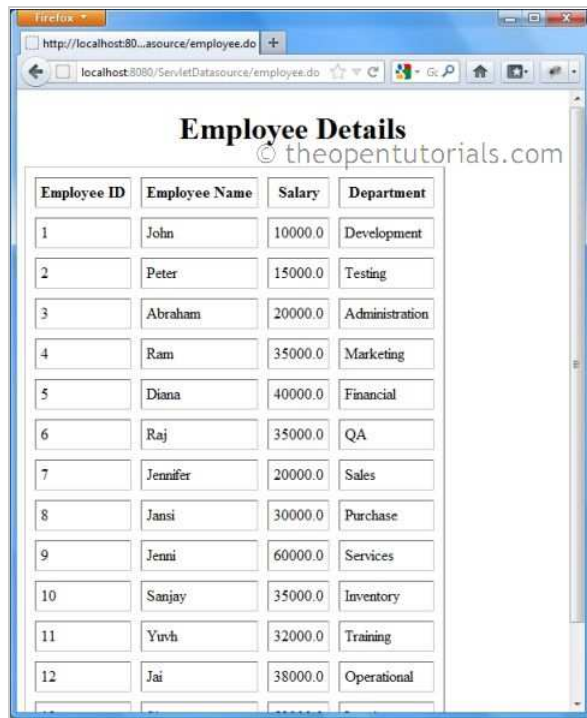
The complete folder structure of this example is shown below.



7. Output

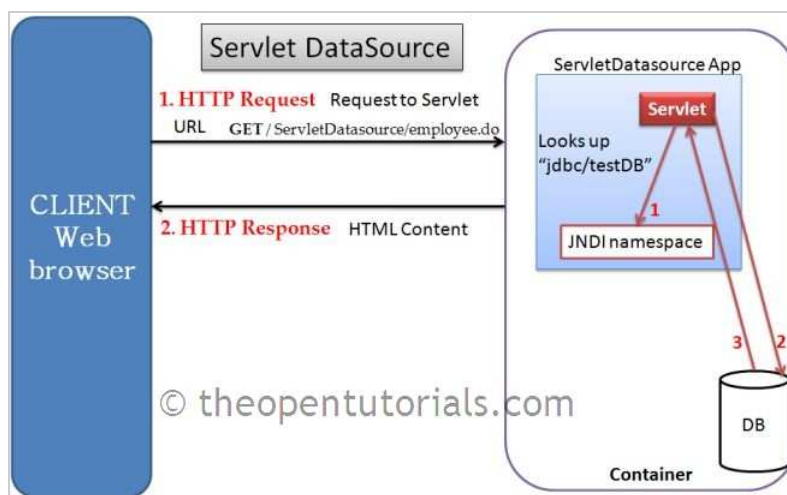
Use Ctrl + F11 in Eclipse to run the Servlet. The URL for requesting the Servlet is

<http://localhost:8080/ServletDataSource/employee.do>



| Employee ID | Employee Name | Salary | Department |
|-------------|---------------|---------|----------------|
| 1 | John | 10000.0 | Development |
| 2 | Peter | 15000.0 | Testing |
| 3 | Abraham | 20000.0 | Administration |
| 4 | Ram | 35000.0 | Marketing |
| 5 | Diana | 40000.0 | Financial |
| 6 | Raj | 35000.0 | QA |
| 7 | Jennifer | 20000.0 | Sales |
| 8 | Jansi | 30000.0 | Purchase |
| 9 | Jenni | 60000.0 | Services |
| 10 | Sanjay | 35000.0 | Inventory |
| 11 | Yuvh | 32000.0 | Training |
| 12 | Jai | 38000.0 | Operational |

8. Program control flow



- Web client (browser) sends requests to Servlet through request URL. Since the request is sent directly to Servlet it is a HTTP GET Request
- Servlet in doGet() method, looks up (using JNDI) JDBC resource (jdbc/testDB) in global JNDI namespace (java:/comp/env) and stores it in DataSource object. Using this object it creates Connection and executes SQL statement and responds with actual HTML content

The content of HTTP request and response headers can be monitored using browser add-ons such as Live HTTP Headers, HttpFox, etc

8.1. Related posts:

- How to create a Servlet with Eclipse and Tomcat
- Pagination in Servlet and JSP
- Getting checkbox values from HTML form in Servlet
- Get all parameters in HTML form using getParameterMap()

Tags: [accessing datasource from servlet](#), [Apache Tomcat](#), [configuring datasource in tomcat](#), [jdbc examples](#), [Servlet doGet example](#), [Servlet examples](#), [Servlet jdbc examples](#), [servlet jdbc mysql example](#), [using DataSource in Servlet](#)

ALSO ON THE OPEN TUTORIALS

[Enum Advanced Features](#) 1 comment[JAXB: Marshalling and Unmarshalling CDATA block](#) 3 comments[Rules for Developing EJB 3.x Session Beans](#) 1 comment[How to get list of Installed Apps in Android](#) 15 comments

WHAT'S THIS?

[Android: Parsing XML with children and attributes using DOM](#) 2 comments[Post-Redirect-Get \(PRG\) pattern in Servlet, JSP](#) 5 comments[JAXB: Marshalling and Unmarshalling CDATA block using EclipseLink ...](#) 9 comments[Android: How to send HTTP GET request to Servlet using HttpURLConnection](#) 11 comments

8 comments



Join the discussion...

Best ▾ Community

Share

Login ▾

**kadhiresh** • 18 days ago
Hi NIVAS (nithiya vasudevan),

you make me feel free about this concept thanks a lot

| • Reply • Share ›

**Shaik Abdul Shukoor** • 5 months ago
Hai Nithya,

This is Shukoor.I struggled a lot for this jndi connection as i am a newbie.This explanation is too good and thank you so much for your help.Plz keep on posting these articles as these are very useful for the newbies like me.Thank you once again

| • Reply • Share ›

**sandeep** • 8 months ago
Just wanted to add that if required, we can also configure connection pool in tomcat by using java code:[connection pool](#)

| • Reply • Share ›

**Rajesh** • a year ago
Hi Nithya,
Thank you very much.. I was struggled lot to make connection with db. Now i got clear idea. Please keep on posting such articles ,this will help a lot for beginners like me.

| • Reply • Share ›

**Ruthreshwar** • a year ago
Hi,
Thank you. This clearly explained me the JNDI concept. It was very useful. I have a doubt whether driver class will be loaded if we use this JNDI concept?

| • Reply • Share ›

**Nithya Vasudevan** → **Ruthreshwar** • a year ago
Yes. The driver class is loaded by the Application Server during start up.

| • Reply • Share ›

**Lalit Jha** • a year ago
Thanks.

It helped me a lot. I was looking for configuring JNDI in my application itself rather than in tomcat, to remove server specific configuration.

Can you suggest how configuring context in META-INF in application will behave on JBoss.

| • Reply • Share ›

**Praveen Macherla** → **Lalit Jha** • a year ago
I don't think you can do the same way in JBoss. In JBoss you have to define a *-ds.xml to define datasources. Check the JBoss DataSource configuration xml files which may be available in the jboss folder itself.

| • Reply • Share ›

Subscribe

Add Disqus to your site

Views

Pagination in Servlet and JSP - 75,271 views

How to create a simple EJB3 project in Eclipse (JBoss 7.1) - 69,078 views

Android: Custom ListView with Image and Text using ArrayAdapter - 67,822 views

How to create a Servlet with Eclipse and Tomcat - 63,491 views

How to create and consume a simple Web Service using JAX WS - 55,461 views

Recent Posts

Android: Contextual Action Bar for ListView Item Deletion using ActionBarSherlock

Android BadTokenException: Unable to add window, is your activity running

Android: Expandable List View Example

How to retrieve all rows from MySQL table using JDBC

How to retrieve a row from MySQL using JDBC

November 2013

M T W T F S S

1 2 3

4 5 6 7 8 9 10

11 12 13 14 15 16 17

18 19 20 21 22 23 24

25 26 27 28 29 30

« Jul