

探討 JSP 鏈結資料庫之方法_以使用 JNDI 為例

蔡曜仲 蘇文松 蔡曜先

彰化縣教育網路中心

bryan,happysong,yaoxin @php.boe.chc.edu.tw

摘要

現今各種程式語言與資料庫鏈結時往往需要透過煩雜的步驟，有些實作程式將這些步驟寫成一個函式，但這種作法仍缺乏一致性的界面，而 JNDI(Java Naming and Directory Interface)的技術則提供一致性的界面，並將這些鏈結資源物件化，在撰寫程式時可以方便的取用這些物件，達到和資料庫鏈結的目的。而本研究的目的則是實作與探討，在 Resin 中使用 JNDI 與資料庫鏈結時所應注意的設定與事項。

關鍵詞： Resin、java、JSP、JNDI、MySQL。

Abstract

Nowadays, when the codes connect to the database, it always needs complex steps. Lots of practice leads these steps to a function. But it is still lack of united interface. However, the technique of JNDI(Java Naming and Directory Interface) could offer united interface and make these data source to be objects. When coding, you use these objects easily to connect to database. In this research, what we do is to practice and to research what configuration and things we must pay attention to.

Keywords: Resin、java、JSP、JNDI、MySQL.

1. 前言

透過網頁存取資料庫已是現今資訊界相當普遍的作法，如何鏈結與管理眾多的資料庫資源則是程式開發者相當重要的課題，透過 JNDI，可以幫程式開發者更精鍊的撰寫程式碼，並且更有條理的管理鏈結資料庫的資源。JNDI 是 Sun Microsystems(昇陽公司)所提供的一種物件管理界面，透過 JNDI 可以將每個鏈結資料庫的資源物件化，並利用 Sun Microsystems 的 JSTL 技術，在程式當中取用設定好的鏈結資源。本文將說明如何透過正確的設定 Resin，使 JNDI 與資料庫鏈結。

1.1 研究目的

本研究的目的旨在探討，在 Resin 當中使用 JNDI 鏈結資料庫時，如何將 JDBC 函式庫放到正確的位置，並證明使用 JNDI 鏈結資料庫可以減少程式碼的撰寫，增加效率。

1.2 研究限制

本研究所使用的各種軟體套件如下：

Resin3.0.14 架設在 Windows2000 sp3

MySQL 4.1.5 架設在 FreeBSD5.1，其它版本或資料庫則有待進一步驗證。

2. 文獻探討

本研究旨在探討如何在 Resin 當中正確的使用 JNDI 與資料庫鏈結，因此本文中就 Resin 與 MySQL 的相關文獻進行探討。

2.1 網頁伺服器 Resin

網頁伺服器 Resin 是現今能提供 JSP(JavaServer Page) & Servlet 伺服的網頁伺服器之一，它具有以下特徵：

- Full specification support for JSP and servlets
- Load balancing
- Built-in Web server with high-performance HTML processing
- Resin XTP templates for dynamic web page creation
- Virtual hosting
- Model-View-Controller support with JSP Templates
- Web services using Burlap and Hussein
- Resin Hardcore—incorporation of Resin into a Linux kernel
- J2EE and EJB support

Resin 是一個能夠完全支援 J2EE 技術的網頁伺服器，因此，它本身也提供了鏈結資料庫的功能，以提昇鏈結資料庫的方便性。

2.2 資料庫 MySQL

MySQL database 是使用相當廣泛的免費資料庫。它的主要特徵有以下幾點([1])：

- Speed. MySQL is fast. The developers contend that MySQL is about the fastest database you can get. You can investigate this claim by visiting <http://www.mysql.com/benchmark.html>, a performance-comparison page on the MySQL Web site.
- Ease of use. MySQL is a high-performance but relatively simple database system and is much less

complex to set up and administer than larger systems.

- Query language support. MySQL understands SQL, the language of choice for all modern database systems.

- Capability. Many clients can connect to the server at the same time. Clients can use multiple databases simultaneously. You can access MySQL interactively using several interfaces that let you enter queries and view the results: command-line clients, Web browsers, or X Window System clients. In addition, a variety of programming interfaces are available for languages such as C, Perl, Java, PHP, and Python. You can also access MySQL using applications that support ODBC (Open Database Connectivity), a database communications protocol developed by Microsoft.

Thus, you have the choice of using prepackaged client software or writing your own for custom applications.

- Connectivity and security. MySQL is fully networked, and databases can be accessed from anywhere on the Internet, so you can share your data with anyone, anywhere. But MySQL has access control so that people who shouldn't see your data can't. To provide additional security, MySQL now supports encrypted connections using the Secure Sockets Layer (SSL) protocol.

- Portability. MySQL runs on many varieties of UNIX, as well as on other non-UNIX systems, such as Windows and OS/2. MySQL runs on hardware from home PCs to high-end servers.

- Small size. MySQL has a modest distribution size, especially compared to the huge disk space footprint of certain commercial database systems.

- Availability and cost. MySQL is an Open Source project, freely available under the terms of the GNU General Public License (GPL). This means that MySQL is free for most in-house uses. (If you want to sell MySQL or services that require it, that is a different situation and you should contact MySQL AB.)

- Open distribution. MySQL is easy to obtain; just use your Web browser. If you don't understand how something works or are curious about an algorithm, you can get the source code and poke around in it. If you don't like how something works, you can change it. If you think you've found a bug, report it; the developers listen.

使用 MySQL database 主要是因為其為免費軟體，而且安裝與啟動均相當容易，可以在許多種作業環境中穩定執行。

3. 實作

實作的步驟主要分為三個部分，一是安裝、啟動 MySQL database，二是安裝、啟動 Resin，三

則為建立、撰寫一個測試的 website，接著就是修改相關設定並進行測試。

3.1 環境架構

表 1 環境架構

主機	pc1	pc2
Software	MySQL 4.1.5	Resin 3.0.14
O.S.	FreeBSD5.1	Windows 2000 sp3
IP	192.168.0.1	192.168.0.2

如表 1 所示，本研究使用兩台電腦主機分別使用作業系統 FreeBSD5.1 安裝 MySQL4.1.5，與作業系統 Windows2000 sp3 安裝 Resin3.0.14。並且撰寫程式碼測試鏈結。

3.2 建立資料庫

由於本研究著重於如何架構 JNDI，故不詳述 MySQL 的安裝過程，但建立資料庫的語法如下，可依此語法建立一個可供鏈結的資料庫。

表 2 建立資料庫語法

```

CREATE DATABASE resin_test;

CREATE TABLE `table1` (
  `field1` TINYINT(4) NOT NULL DEFAULT '0',
  `field2` CHAR(24) DEFAULT NULL,
  PRIMARY KEY (`field1`)
)ENGINE=MyISAM DEFAULT CHARSET=big5;

INSERT INTO `table1` (`field1`, `field2`) VALUES
(3, '蔡依林'),
(2, '侯佩岑'),
(1, '林志玲');

```

3.3 下載與安裝 Resin 3.0.14

Resin 的官方網址為 "http://www.caucho.com/" 下載 Resin 3.0.14 的畫面如圖 1，點選 resin-3.0.14.zip 就可以下載，存到 pc2 的硬碟當中，解壓縮後，就可會產生一個 resin-3.0.14 的資料夾。

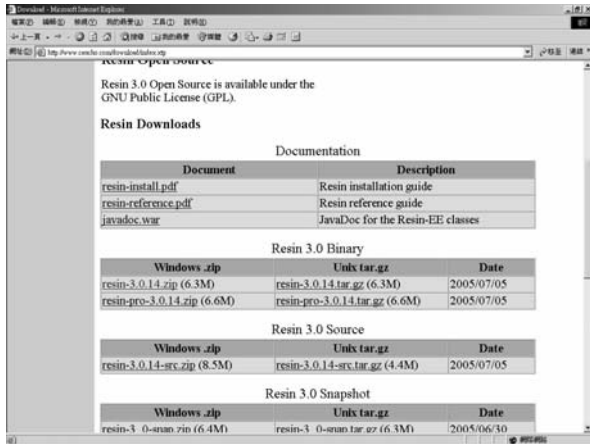


圖 1 Resin 下載頁面

3.4 修改設定

在 resin-3.0.14/conf/ 當中，可以找到 resin.conf 這個檔案，此檔案即為 Resin 伺服器的設定檔，可以用文字編輯器將此檔案開啟，找到 <database></database> 這個設定區塊，並且修改其中的若干設定，如表 3 和表 4：

表 3 為安裝完畢之後的原始設定，找到設定 JNDI 的 <database></database> 區塊後，首先，將第 19 行的註解符號移到第 6 行，也就是取消 <database></database> 區塊的註解。接著的修改第 8、10、11、12 行，也就是讓 Resin 伺服器啟動時，就與 pc1 自動產生鏈結，以供網頁中的程式碼取用。

修改的部份如下：

- 第 8 行：jdbc/mysql → jdbc/resin_test
 - 第 10 行：jdbc/mysql://localhost:3306/database → jdbc/mysql://192.168.0.1:3306/resin_test
 - 第 11 行：填入 mysql 的 user ID
 - 第 12 行：填入 user 的 password
- 此即為先前在 pc1 建立用戶的 ID 和 password。

表 3 設定檔片段_原始檔

```

01 <!--
02 - Sample database pool configuration
03 -
04 - The JDBC name is java:comp/env/jdbc/test
05 -
06
07 <database>
08 <jndi-name>jdbc/mysql</jndi-name>
09 <driver type="org.gjt.mm.mysql.Driver">
10 <url>jdbc:mysql://localhost:3306/database </url>
11 <user></user>
12 <password></password>
13 </driver>
14 <prepared-statement-cache-size>8</prepared-statement-cache-size>
15 <max-connections>20</max-connections>
16 <max-idle-time>30s</max-idle-time>
17 </database>
18
19 -->

```

表 4 設定檔片段_原始檔

```

01 <!--
02 - Sample database pool configuration
03 -
04 - The JDBC name is java:comp/env/jdbc/test
05 -
06 -->
07
08 <database>
09 <jndi-name>jdbc/resin_test</jndi-name>
10 <driver type="org.gjt.mm.mysql.Driver">
11 <url>jdbc:mysql://192.168.0.1:3306/resin_test</url>
12 <user>username</user>
13 <password>password</password>
14 </driver>
15 <prepared-statement-cache-size>8</prepared-statement-cache-size>
16 <max-connections>20</max-connections>
17 <max-idle-time>30s</max-idle-time>
18 </database>
19

```

3.4 建立網站

在 resin-3.0.14/webapps 中建立三個資料夾：

1. resin-3.0.14/webapps/resin_test
 2. resin-3.0.14/webapps/resin_test/WEB-INF
 3. resin-3.0.14/webapps/resin_test/WEB-INF/lib
- 並且要將 sun 所提供的 jstl.jar 與 standard.jar 兩個函式庫放到 lib 裏面，以期可以使用 JSTL 標籤函式庫，架構如圖 2。

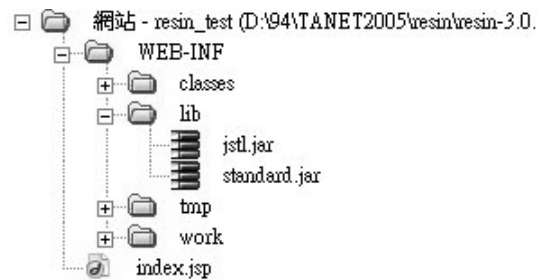


圖 2 網站架構

3.5 置入函式庫

MySQL 所提供的 JDBC 函式庫是程式與資料庫鏈結的主要介面，其中包含了驅動程式和鏈結的介面函式，因此，鏈結的過程一定要將函式置入。

函式庫的取得可以到 MySQL 的官方網站下載，下載網址為：

<http://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-3.1.10.zip/from/pick>

可以取得 mysql-connector-java-3.1.10.zip 的檔案，解壓縮之後可以找到 mysql-connector-java-3.1.10-bin.jar 這就是 JNDI 所需要的 JDBC 函式庫，關於這檔案的放置位置，即為本研究的重點，根據 Mastering Resin 一書中所提到的敘述如，表 5([2])。

表 5 Mastering Resin 內容

I. Library Placement

When you're using the JNDI configuration, you must place the driver class or JAR in an appropriate location where the server can find it. As you saw earlier in the chapter, one of the first places the system attempts to find the driver is on the global classpath. If the driver isn't on the classpath, the system looks in the /lib directory of the server. In most cases, this is the best place to put the driver because it is specific to the server. You have the option of putting the driver in the application's WEB-INF/lib or WEB-INF/classes directory if you want it to be more aligned with the specific application.

依其文意，JDBC 函式庫可以放置在三個地方：

- 1. resin-3.0.14/lib
2. resin_test/WEB-INF/lib
3. resin_test/WEB-INF/classes

使用 3.6 的程式碼實際測試，只有放在 resin-3.0.14/lib 下，才能被找到，測試報告如下：

- 1. 將 JDBC 函式庫放到 resin-3.0.14/lib 下 resin 可以正常啟動，並且可以正常瀏覽到網頁如圖 3。

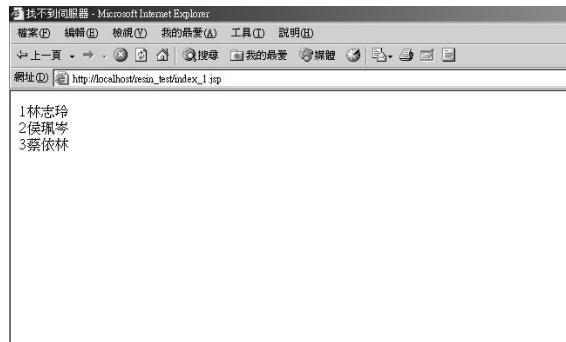


圖 3 瀏覽網頁

- 2. 將 JDBC 函式庫放到 resin_test/WEB-INF/lib 下 resin 啟動時出現以下訊息如圖 4，表示 resin 啟動時找不到 JDBC 函式庫。

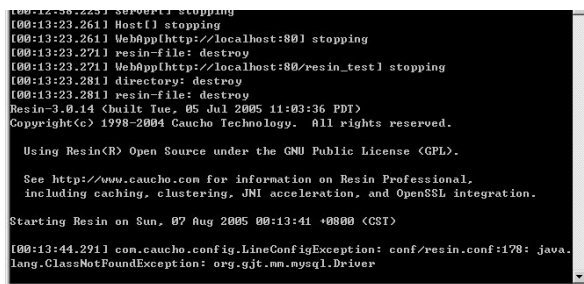


圖 4 錯誤訊息

- 3. 將 JDBC 函式庫解壓縮後，放到 resin_test/WEB-INF/classes 下 resin 啟動時出現以下訊息如圖 4，表示 resin 啟動時找不到 JDBC 函式庫。

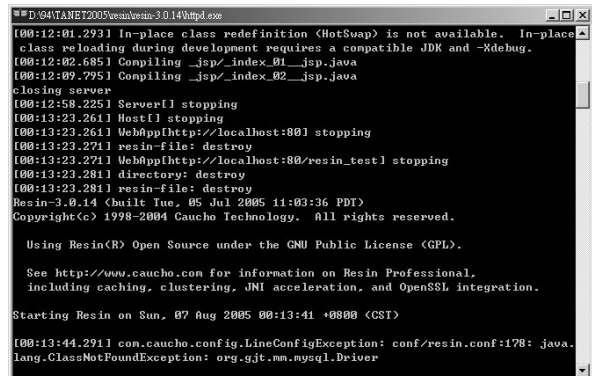


圖 5 錯誤訊息

此外，官方網站也有一個網頁提到 JDBC 函式庫的放置位置，內容如，表 6([4])。

表 6 官方網站內容

Driver Configuration
The jar file with the driver in it can be placed in WEB-INF/lib, although it is often best to place your database driver's jar file in \$RESIN_HOME/lib/local/, which makes the driver available to all of your web applications.

依其文意，JDBC 函式庫可以放在兩個地方：

- 1. resin_test/WEB-INF/lib
2. resin-3.0.14/WEB-INF/lib
3. resin-3.0.14/lib/local

經測試之後，結果如下：

- 1. 將 JDBC 函式庫放在 resin_test/WEB-INF/lib 出現如圖 5 的錯誤訊息。
2. 將 JDBC 函式庫放在 resin-3.0.14/WEB-INF/lib 出現如圖 5 的錯誤訊息。
3. 將 JDBC 函式庫放在 resin-3.0.14/lib/local，則可以看到如圖 3 的畫面。

由以上的測試得知，JDBC 的函式庫應當放在 \$RESIN_HOME/lib 底下或這個目錄下的目錄內，Resin 在使用 JNDI 與資料庫鏈結時才能正確找到 JDBC 的函式庫。

3.6 撰寫程式碼

接著在 resin_test 底下建立一個檔案 index.jsp 程式碼如表 5：

表 7 index_1.jsp

```
<%@ page contentType="text/html; charset=big5" language="java"
import="java.sql.*" errorPage="" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=big5">
<title>無標題文件</title>
</head>

<body>
<sql:query dataSource="jdbc/resin_test" var="resin">
    SELECT * FROM table1
</sql:query>

<c:forEach items="${resin.rows}" var="rs">
    ${rs.field1} ${rs.field2} <br>
<br>
</c:forEach>
</body>
</html>
```

表 8 index_2.jsp

```
<%@ page contentType="text/html; charset=big5"
language="java" import="java.sql.*" errorPage="" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=big5">
<title>無標題文件</title>
</head>

<body>
<%
Class.forName("org.gjt.mm.mysql.Driver");
String myuser      ="*****";
String mypassword  ="*****";
String database ="resin_test";
String charSet
="jdbc:mysql://163.23.253.162:3306/" + database +
"?user="+myuser+"&password="+mypassword+"
&useUnicode=true&characterEncoding=Big5";
Connection con = DriverManager.getConnection(charSet);
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("SELECT * FROM
table1");
while(rs.next()){
out.print(rs.getString("field1"));
out.print(rs.getString("field2"));
out.print("<br>");
}
rs.close();
stmt.close();
%>
</body>
</html>
```

表 7 為使用 JNDI 的介面來鏈結資料庫的程式碼，表 8 則為一般的鏈結方法。比較 <body></body> 標籤之間所撰寫的字元數，表 7 只用了 163 個字元，而表 8 則用了 527 個字元，

從以上的兩組程式碼可以明顯的看出，透過 JNDI 介面來鏈結資料庫不但可以結省程式碼的撰寫，而且還可以將程式碼重複利用。

4. 結論

在參考 Resin 的官方網站與相關文獻時，均有題到 JDBC 的函式庫可以放在 website/WEB-INF/lib 底下，但經測試均推翻以上說法，JDBC 函式庫一定要放在 \$RESIN_HOME/lib 底下或其下的目錄內，才能讓 Resin 在使用 JNDI 介面時找到 JDBC 的函式庫。

另外，使用 JNDI 介面可以有效減少鏈結資料庫程式碼的撰寫，並重複使用鏈結設定。

參考文獻

- [1] Paul DuBois, "MySQL, Second Edition", Introduction, Sams, page 3, July 2003.
- [2] Richard Hightower and Joseph D. Gradecki, "Mastering Resin", Wiley Publishing, Chapter 11, 2003.
- [3] Todd M. Thomas, "Java Data Access—JDBC, JNDI, and JAXP", M&T Books An imprint of Hungry Minds, Chapter 1, 2002.
- [4] <http://www.caucho.com/resin-3.0/db/config.xtp>, Aug 2005.