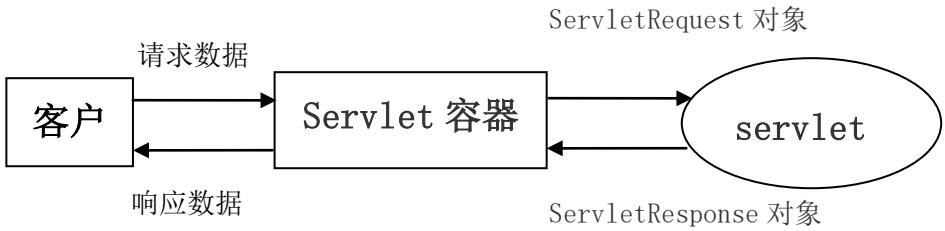
Tomcat 容器与 Servlet/JSP

Jakarta Tomcat 服务器是一种 Servlet/JSP 容器。Servlet 是一种运行在支持 Java 语言的服务器上的组件。Servlet 扩展了 Java Web 服务器功能。



一、运行环境

1、jdk-1_5_0_02-windows-i586-p.exe
下载地址：http://java.sun.com/javase/downloads/index.jsp
2、jakarta-tomcat-5.5.7.exe 或 jakarta-tomcat-5.5.7.zip
下载地址：http://tomcat.apache.org/index.html

Servlet/JSP Spec	Apache Tomcat version
2.5/2.1	6.0.10
2.4/2.0	5.5.23
2.3/1.2	4.1.34
2.2/1.1	3.3.2

3、开发环境
集成开发环境 NetBeans 7
下载地址：http://netbeans.org/index_zh_CN.html

开发环境：Eclipse 3.1

二、JAVA 包及类库和方法查询文件

jdk-1_5_0-doc.zip
servlet-2_4-fr-spec-doc.zip
下载地址：http://tomcat.apache.org/index.html

三、JDK 和 TOMCAT 配置：

1、 JAVA-HOME = D:\jdk1.5.0_02
2、 PATH = %JAVA-HOME%\bin
或 D:\jdk1.5.0_02\bin
3、 CLASSPATH =

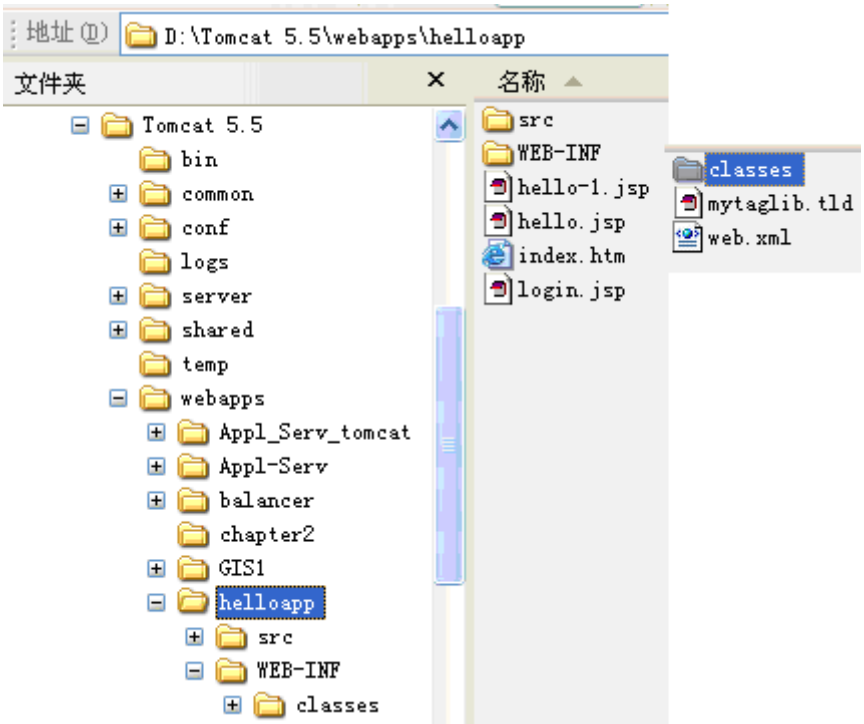
.;D:\jdk1.5.0_02\LIB\TOOLS.JAR;
D:\jdk1.5.0_02\LIB\DT.JAR;
D:\TOMCAT5.5\COMMON\LIB\servlet-api.jar;
D:\TOMCAT5.5\COMMON\LIB\jsp-api.jar

4、 TOMCAT_HOME = D:\TOMCAT5.5

5、 CATALINA_HOME = D:\TOMCAT5.5

四、配置JSP及Servlet

安装jakarta-tomcat完成后，目录结构如下图：



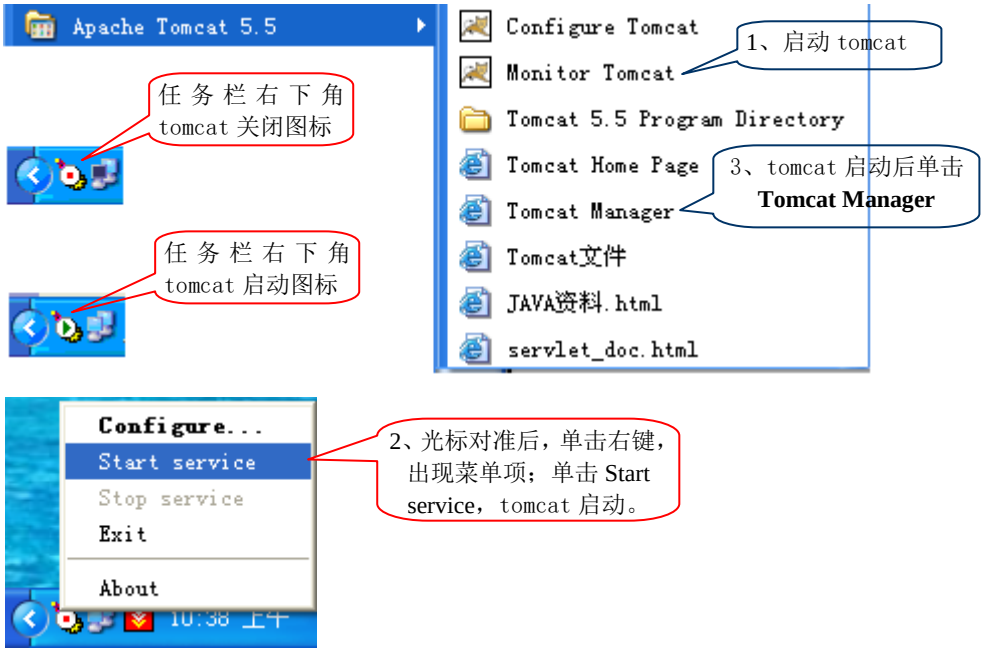
在 webapps 文件夹下，建立你的应用程序文件夹，如 helloapp，在 helloapp 下建立一个子文件夹 WEB-INF，当中放有 **web.xml** 配置文件（可从其他文件夹中复制一个过来，如从 examples 文件夹下复制）；在 WEB-INF 下再建立一个子文件夹 classes。

你的 JSP 和 Html 文件直接放在 helloapp 下； 编译好的 JavaBean、Servlet 的 class 文件放在 WEB-INF 下的 classes 目录，而且包的路径要与目录路径一致。

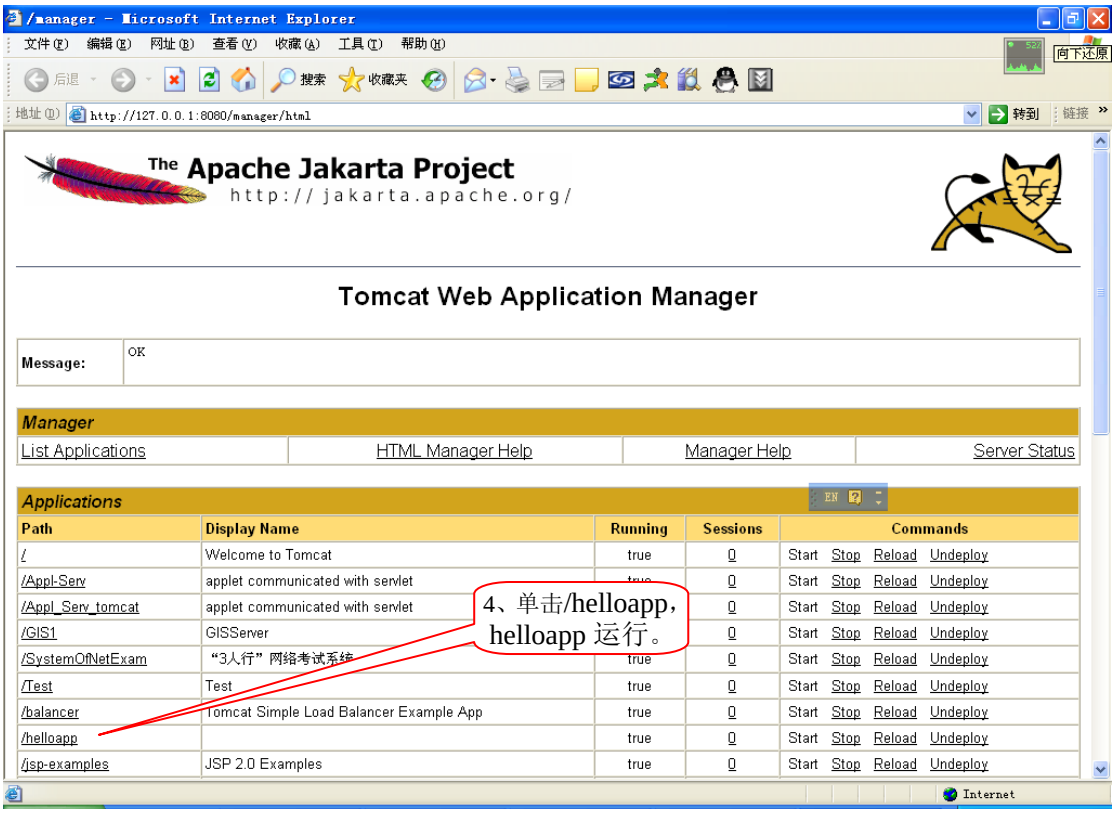
四、运行：

1、启动数据库；

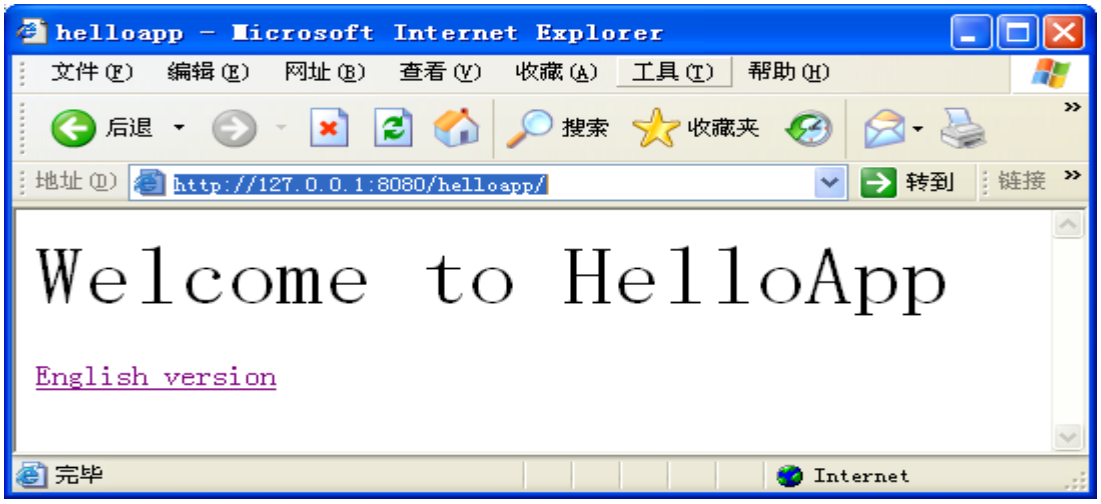
2、按图中 1 和 2 步骤启动（**Monitor Tomcat**）**Tomcat** 服务器；



3、按图中 3 步骤启动“**Tomcat Web Application Manager**”。单击 Tomcat Manager，输入用户名：Admin，回车进入 Tomcat Web 应用管理界面（**Tomcat Web Application Manager**）。



4、启动你自己的应用程序。见上图，单击/helloapp 或在浏览器的地址栏中直接输入 http://127.0.0.1:8080/helloapp/ helloapp 应用程序运行，见下图。



五、参考文件

1、Tomcat 与 Java Web 开发技术详解，电子工业出版社

2、开发专家之 Sun ONE JSP 应用开发详解，定价：49.00 元 （含光檄），版次：2002 年 1 月第 1 版，2002 年午 1 月第 1 次印刷，电子工业出版社，北京 BEIJING。

3、Apache 2.0 手册中文版翻译项目 （Apache HTTP 服务器 2.0 版本）
http://www.douzhe.com/apm/ApacheManual/zh-cn/

《Tomcat 与 JavaWeb 开发技术详解》第 6 章问题解答

【作者】 孙卫琴

问题 1（第 6 章，第 74 页）：如何在 MySQL 数据库中以批处理方式执行 books.sql 文件中的 SQL 语句。

答复：

如果不想在 MySQL.exe 客户程序中手工输入 SQL 语句，也可以直接运行 books.sql，步骤为先转到 MySQL 安装目录的 bin 目录下，输入如下命令：

C:\mysql\bin> mysql -u root -p <C:\sourcecode\chapter6\books.sql

接下来会提示输入 root 用户的口令，此处输入口令"1234"：

Enter password: ****

接下来 MySQL 客户程序就会自动执行 C:\sourcecode\chapter6\books.sq 文件中的所有 SQL 语句。在以上 mysql 命令中，"<"后面设定 SQL 脚本文件的路径。

问题 2（第 6 章，第 79 页）：为什么把书中的例子改为连结到其他数据库后（如 Ms SQL Server），程序运行时就会抛出找不到 JDBC 驱动程序类的错误？

答复：

直接使用 JDBC 访问其他数据库时，必须提供它的驱动程序，存放位置和 mysqldriver.jar 文件的位置相同，可放在 <CATALINA_TOME>/common/lib 目录或 <CATALINA_HOME>/webapps/bookstore/WEB-INF/lib 目录下。SQL Server2000 数据库驱动

程序的下载地址为：

http://www.microsoft.com/china/sql/downloads/2000/jdbc.asp

问题 3（第 6 章，第 79 页）：如果通过 JDBC-ODBC 驱动程序访问数据库，能否不在操作系统中配置 ODBC 数据源？

答复：

必须在 Windows 操作系统中先配置 ODBC 数据源，然后才能通过 JDBC-ODBC 驱动程序访问该 ODBC 数据源。

问题 4（第 6 章，第 89 页）：连结数据库有两种方式：a.通过 DataSource 数据源连结数据库，b.直接通过 JDBC 连结数据库，这两种方式有何联系和区别，各自适用场合是什么？

答复：

（1）联系：JDBC2.0 提供了 javax.sql.DataSource 接口，DataSource 接口的实现类本身通过 JDBC API 来连结数据库。如果要了解 Tomcat 实现 DataSource 的细节，可以察看

org.apache.commons.dbcp.BasicDataSource 类的源代码，在 Apache 网站上可以下载。

（2）区别：通过 JDBC API 连结数据库是一种最原始、最直接的方法。而 DataSource 封装了通过 JDBC API 来连结数据库的细节，它采用数据库连结池机制，把可用的数据库连结保存在缓存中，避免每次访问数据库都建立数据库连结，这样可以提高访问数据库的效率。

（3）适用场合： 在任何 Java 应用中，都可以直接通过 JDBC API 连结数据库，如果需要的话，可以手工编程实现数据库连结池。当 Java 应用运行在 JavaWeb 容器或 EJB 容器中时，可以优先考虑使用由容器提供的 DataSource。以 Tomcat 容器为例，DataSource 实例被作为 JNDI 资源发布到 Tomcat 容器中，Tomcat 容器负责维护 DataSource 实例的生命周期，Java Web 应用通过 JNDI 来获得 DataSource 实例的引用。

问题 5（第 6 章，第 83 页）：书中把 MySQL 的驱动程序包命名为 mysqldriver.jar，很容易让人误解，不如直接使用 mysql-connector-java-3.X.X-bin.jar 更好？

答复：

MySQL 的 JDBC 驱动程序的下载地址为：

http://dev.mysql.com/downloads/connector/j/3.1.html

把从这个下载地址获得的压缩软件包展开，将获得 MySQL 驱动程序包,名为“mysql-connector-java-3.X.X-bin.jar”，本书把它改名为“mysqldriver.jar”，使得书中引用这个文件名比较方便，能保持版面的简洁。

问题 6（第 6 章，第 82 页）：发现一个问题，书中加载并注册 MySQL 驱动程序用了两行代码：

```
Class.forName("com.mysql.jdbc.Driver");
DriverManager.registerDriver(new com.mysql.jdbc.Driver());
```

以上两行代码可以用一行代码代替：

```
DriverManager.registerDriver(Class.forName("com.mysql.jdbc.Driver").newInstance());
```

答复：

言之有理，你提供的这种方式更严谨，更合理。不过书上的例子也会正常编译并运行。

尽管在 DbJsp.jsp 中并没有通过<import>指令显式引入 com.mysql.jdbc.Driver 类，Tomcat 在编译第二行代码时并不会抛出无法解析 com.mysql.jdbc.Driver 的错误，这是因为 mysqldriver.jar 文件已经被发布到 <CATALINA_TOME>/common/lib 目录或 <CATALINA_HOME>/webapps/bookstore/WEB-INF/lib 目录下，如果发布到 <CATALINA_TOME>/common/lib 目录下，当 Tomcat 服务器启动时就会加载 mysqldriver.jar 文件中的所有 Java 类；如果发布到<CATALINA_HOME>/webapps/bookstore/WEB-INF/lib 目录下，当 Tomcat 服务器启动 bookstore 应用时就会加载 mysqldriver.jar 文件中的所有 Java 类，无论在哪一种情况下，都会保证当编译 DbJsp.jsp 文件时，com.mysql.jdbc.Driver 类已经被加载到 JVM 中。