

# 基础 Servlet 编程

实验编号: exp08

## 0.1 实验目的

掌握基础的 Servlet Web 编程技术。

## 0.2 实验要求

设计 Servlet 程序、MySQL 数据库操作程序和 JSP 页面程序，完成用户注册功能。

软件开发环境依赖如下：

**Apache Tomcat** 选择 8.0 及以上版本，下载地址：<https://tomcat.apache.org>

**Eclipse** 选择 Eclipse Java EE for Web Developers, Version Mars.2 Release(4.5.2) 及以上版本

**MySQL Database**

**mysql-connector-java-<version>-bin.jar**

## 0.3 知识和技能储备

### 0.3.1 如何在 Eclipse 中配置 Apache Tomcat 服务器

**安装并测试 Tomcat 服务器**

下载并解压缩 Apache Tomcat 的 ZIP 包，将解压后目录放置到特定的位置，如：C:\Program files\Eclipse 即完成安装。双击 Tomcat 安装目录 bin 下的 startup.bats 批处理文件，既可以启动 Tomcat 服务器，默认在 8080 端口响应用户请求。Tomcat 启动后，可以在浏览器中访问 <http://localhost:8080> 测试 Tomcat。

**注意：**如果出现 Tomcat 无法启动，请自行搜索尝试排除 JDK/JRE 环境变量配置和默认端口号 8080 冲突问题。

## 在 Eclipse 中配置 Tomcat 服务器

选择 Windows → Preferences → Server → Runtime Environments → Add..., 进入服务器配置页面; 选择服务器类型, 在此我们选择 Apache 的 Tomcat 8.0; 点击 Next 按钮打开服务器目录和 JDK 选择界面, 选择相应版本的 Tomcat 的安装路径, 点击 Finish 完成服务器配置。

### 0.3.2 如何部署 MySQL JDBC 驱动

解压缩 mysql-connector-java-x.x.x.tar, 将解压后目录中的 mysql-connector-java-x.x.x-bin.jar 分别复制到 Apache Tomcat 下的 lib 目录和后续所创建的动态 Web 项目目录 WebContent/WEB-INF/lib/中; 使用 Maven 构建 Web 项目则需要配置 Maven Dependency。

例如:

```
1 >ls ~/Installed/apache-tomcat-7.0.32/lib/
2 annotations-api.jar          mysql-connector-java-5.1.22-bin.jar
3 catalina-ant.jar             servlet-api.jar
4 catalina-ha.jar              tomcat-api.jar
5 ... ..
6 >ls ~/Documents/Workspace/JavaEE_EXP_01_01/WebContent/WEB-INF/lib/
7 mysql-connector-java-5.1.22-bin.jar
```

### 0.3.3 如何安装操作 MySQL 数据库

#### 安装 MySQL

参考 MySQL 安装手册: [guide-for-mysql-zip-archive-installation-20171106.pdf](#)

#### 操作 MySQL

登陆 MySQL 数据库。

```
1 > mysql -u root -p
3 Welcome to the MySQL monitor. Commands end with ; or \g.
4 Your MySQL connection id is 40
5 Server version: 5.1.66-0ubuntu0.10.04.1 (Ubuntu)
7 Copyright (c) 2000, 2012, Oracle and/or its affiliates. All rights reserved.
9 Oracle is a registered trademark of Oracle Corporation and/or its
10 affiliates . Other names may be trademarks of their respective
11 owners.
13 Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

查看数据库状态。

```
1 mysql> SHOW DATABASES;
2 +-----+
```

```

3 | Database |
4 | +-----+
5 | information_schema |
6 | mysql |
7 | +-----+
8 2 rows in set (0.00 sec)

```

MySQL 数据库操作及 JDBC 基本编程参考手册：handbook-mysql-and-jdbc-20171106.pdf

### 0.3.4 如何调试 Java EE 程序

（教师讲授 / 学生自行搜索学习 Java EE 程序调试方法。）

### 0.3.5 Servlet 编程技术

Servlet 编程相关对象及方法使用参考课程幻灯片及示例代码。

## 0.4 实验过程、步骤及原始记录

### 0.4.1 创建数据库

本实验程序的主要功能为通过 Web 进行用户的注册并将注册信息存入数据库，所以首先我们需要在 MySQL 里创建相应数据库以及表。

#### 登陆 MySQL，创建 testdb 数据库，创建 testuser 测试用户

```

1 > mysql -u root -p // 此处 > 提示符为 MS-DOS 或 Linux/Mac 下 Shell 提示符

3 mysql> CREATE DATABASE testdb;
4 Query OK, 1 row affected (0.02 sec)
5 We create a new testdb database. We will use this database throughout the tutorial.

7 mysql> CREATE USER 'testuser'@'localhost' IDENTIFIED BY 'test123';
8 Query OK, 0 rows affected (0.00 sec)

10 mysql> USE testdb;
11 Database changed

13 mysql> GRANT ALL ON testdb.* TO 'testuser'@'localhost';
14 Query OK, 0 rows affected (0.00 sec)

```

#### 创建 Staffs 表

```

1 mysql> DROP TABLE IF EXISTS Staffs; // 如果存在 Staffs 表则删除
2 Query OK, 0 rows affected (0.15 sec)

```

```

4  mysql> use testdb; // 使用 testdb 数据库
5  Database changed

7  mysql> SHOW TABLES; // 列出该数据库中的表
8  Empty set (0.00 sec)

10 mysql> CREATE TABLE IF NOT EXISTS Staffs(Id INT PRIMARY KEY AUTO_INCREMENT,
11      username VARCHAR(50), pass VARCHAR(50), tel VARCHAR(50), age INT)
12      ENGINE=InnoDB; // 创建 Staffs 表
13 Query OK, 0 rows affected (0.43 sec)

15 mysql> SHOW TABLES; // 列出该数据库中的表
16 +-----+
17 | Tables_in_testdb |
18 +-----+
19 | Staffs            |
20 +-----+
21 1 row in set (0.00 sec)

23 mysql> SELECT * FROM Staffs; // 查询
24 Empty set (0.00 sec)

26 mysql> SHOW COLUMNS FROM Staffs; // 查看表结构
27 +-----+-----+-----+-----+-----+-----+
28 | Field | Type      | Null | Key | Default | Extra |
29 +-----+-----+-----+-----+-----+-----+
30 | Id     | int(11)   | NO   | PRI | NULL    | auto_increment |
31 | username | varchar(50) | YES  |     | NULL    |                |
32 | pass    | varchar(50) | YES  |     | NULL    |                |
33 | tel     | varchar(50) | YES  |     | NULL    |                |
34 | age     | int(11)   | YES  |     | NULL    |                |
35 +-----+-----+-----+-----+-----+-----+
36 5 rows in set (0.13 sec)
37 mysql>

```

至此，我们完成在 MySQL 的数据库 testdb 中 Staffs 表的创建。

## 0.4.2 创建 Eclipse 工程

在 Eclipse 中，参考课程幻灯片创建 Dynamic Web Project，需注意以下事项：

- Default output folder，可以默认不修改；
- 选中 Generate web.xml deployment descriptor；
- 注意 mysql-connector-java-x.x.x-bin.jar 的部署，参考手册上述说明。

工程完成后，目录结构如图 1 所示，可作为参考。

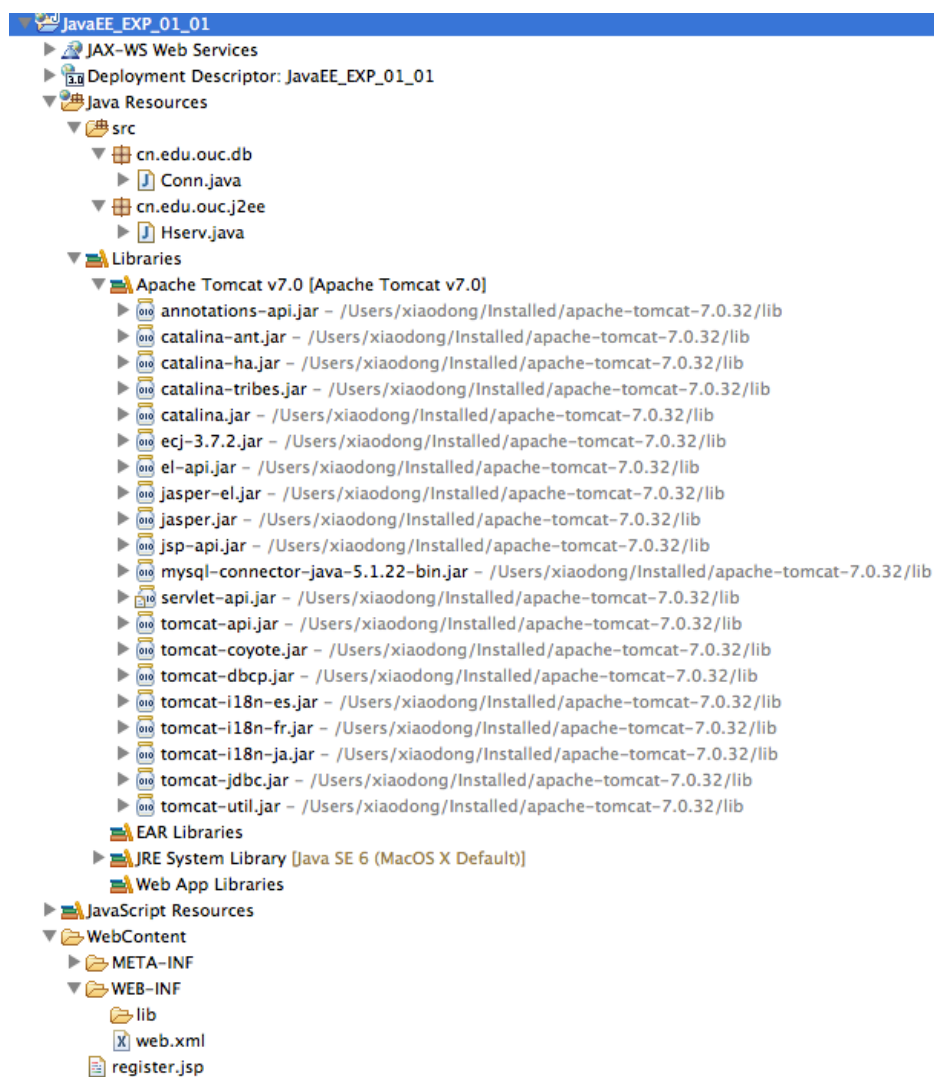


图 1: 工程目录结构

### 0.4.3 创建 JSP 用户注册页面

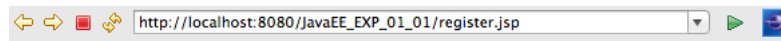
作为用户交互并完成信息输入的接口，我们需要创建 JSP 用户注册页面 `register.jsp`，并将其部署在工程的 `WebContent` 的根目录下。

```
1 > ls /Users/xiaodong/Documents/Workspace/JavaEE_EXP_01_01/WebContent
2 META-INF WEB-INF register.jsp
```

`register.jsp` 文件内容如下。

```
1 <%@ page language="java" import="java.util.*" pageEncoding="utf-8" %>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/
  html4/loose.dtd">
4 <html>
```





## 用户注册程序

用户名:	<input type="text"/>
密 码:	<input type="password"/>
电 话:	<input type="text"/>
年 龄:	<input type="text"/>
<input type="button" value="提交"/>	<input type="button" value="重置"/>

图 2: register.jsp 页面

```
7 private Connection con = null;
8 private Statement stmt = null;
9 private ResultSet rs = null;

11 public void connDB() throws Exception {
12     String url = "jdbc:mysql://localhost:3306/testdb";
13     String username = "testuser";
14     String password = "test123";

16     Class.forName("org.gjt.mm.mysql.Driver").newInstance();
17     con = DriverManager.getConnection(url, username, password);
18 }

20 public void closeDB() throws SQLException {
21     if (rs != null) {
22         rs.close();
23         rs = null;
24     }

26     if (stmt != null) {
27         stmt.close();
28         stmt = null;
29     }

31     if (con != null) {
32         con.close();
33         con = null;
34     }
35 }

37 public ResultSet SelectedSql(String sql) throws SQLException {
39     if (sql == null || sql.equals(" "))
```

```

40     return null;
41     sql = sql.trim();// 去掉字符串两边的空格
42     stmt = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
43         ResultSet.CONCUR_READ_ONLY);
44     rs = stmt.executeQuery(sql);
45     return rs;
46 }

48 public void UpdateSQL(String sql) throws SQLException {
49     stmt = con.createStatement();
50     stmt.executeUpdate(sql);
51 }
52 }

```

### 0.4.5 创建 Servlet 类

我们需要完成 Servlet 类 Hserv.java 并将其放入 cn.edu.ouc.j2ee 包中，使得 register.jsp 页面中表单数据提交操作能够被正确的 Servlet 组件响应。Hserv.java 的代码如下。

```

1  package cn.edu.ouc.j2ee;

3  import java.io.IOException;
4  import java.io.PrintWriter;
5  import java.sql.SQLException;

7  import javax.servlet.ServletException;
8  import javax.servlet.annotation.WebServlet;
9  import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;

13 import cn.edu.ouc.db.Conn; // 导入数据库操作 JavaBean Conn

15 @WebServlet("/Hserv")
16 public class Hserv extends HttpServlet {
17     private static final long serialVersionUID = 1L;
18     private String username;
19     private String pass;
20     private String tel;
21     private String age;

23     Conn cndb = new Conn();

25     public void service(HttpServletRequest request, HttpServletResponse response)
26         throws ServletException, IOException {
27         username = request.getParameter("username");
28         pass = request.getParameter("pass");
29         tel = request.getParameter("tel");
30         age = request.getParameter("age");

```



```

32 response.setContentType("text/html; charset=GB2312");
33 PrintWriter out = response.getWriter();
34 request.setCharacterEncoding("gb2312");
35 String sql = "insert into Staffs(username, pass, tel, age) values ("
36     + username + "',' ' + pass + "',' ' + tel + "',' ' + age + '")";
37 try {
38     try {
39         cndb.connDB();
40     } catch (Exception e) {
41         e.printStackTrace();
42     }
43
44     cndb.UpdateSQL(sql);
45     out.print("<html><body>");
46     out.print("Insert seccessfully.");
47     out.print("</html></body>");
48     cndb.closeDB();
49
50 } catch (SQLException e) {
51     e.printStackTrace();
52 }
53 }
54
55 public void init() throws ServletException {
56     super.init();
57 }
58 }

```

#### 0.4.6 配置 Servlet

修改工程目录 WebContent/WEB-INF 中的 web.xml 配置文件，代码如下。可参考课件说明。

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     xmlns="http://java.sun.com/xml/ns/javaee"
4     xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
5     xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
6     http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
7     id="WebApp_ID" version="3.0">
8
9     <display-name>JavaEE_EXP_01_01</display-name>
10
11     <welcome-file-list>
12         <welcome-file>index.html</welcome-file>
13         <welcome-file>index.htm</welcome-file>
14         <welcome-file>index.jsp</welcome-file>
15         <welcome-file>default.html</welcome-file>
16         <welcome-file>default.htm</welcome-file>
17         <welcome-file>default.jsp</welcome-file>
18     </welcome-file-list>

```

```

20 <!-- Add begin -->
21 <servlet>
22 <servlet-name>Hserv</servlet-name>
23 <servlet-class>cn.ouc.edu.j2ee.Hserv</servlet-class>
24 </servlet>
25
26 <servlet-mapping>
27 <servlet-name>Hserv</servlet-name>
28 <url-pattern>/Hserv</url-pattern>
29 </servlet-mapping>
30 <!-- Add end -->
31
32 </web-app>

```

至此，整个工程代码开发完毕。

#### 0.4.7 工程代码功能测试

在你所创建的工程上单击鼠标右键，点击 Run As ➔ Run on Server，选择 Tomcat Server 后，点击 Finish，在弹出的 Eclipse Web Browser 视图中访问 <http://localhost:8080/yourproject/register.jsp>，完整输入表单数据后，点击提交。

##### 注意

表单测试数据请使用英文，中文测试尚存在一些编码问题没有解决。

登陆 MySQL 数据库，查看 Staffs 表项的变化。

```

1 mysql> SELECT * FROM Staffs;
2  +---+-----+-----+-----+-----+
3  | Id | username | pass   | tel   | age |
4  +---+-----+-----+-----+-----+
5  | 1 | Smith   | wwwxxdsf | 98765 | 45 |
6  | 2 | Smith   | wwwxxdsf | 98765 | 45 |
7  | 3 | Kevin   | dgdfadfa | 9999999 | 30 |
8  +---+-----+-----+-----+-----+
9  3 rows in set (0.01 sec)

```

### 0.5 实验要求

1. 完成上述实验，并撰写实验报告，实验报告需要包含最终实现的界面截图。
2. 在完成上述实验基础上，添加新功能，要求能够在页面中显示出当前已经注册的用户列表，包括用户名、电话和年龄，显示方式基于 HTML 表格或 DIV，其访问入口为/liststaffs。视图可以在 Servlet 中直接硬编码，也可以采用 Forword 方式转发至 stafflist.jsp。
3. 要求采用 Maven 重复上述构建本实验项目工程，参考手册：guide-eclipse-maven-create-Dynamic-Web-Project-20180525.pdf。

4. 要求自行学习使用 Bootstrap CSS 前端技术美化渲染 register.jsp、listsaffs.jsp 等页面。教程参考：<http://www.bootcss.com/>和 <http://getbootstrap.com/2.3.2/>，为后续实验及期末综合设计做好技术储备。