知识点列表

编号	名称	描述	级别
1	改进 ListEmpServlet	通过本案例理解添加员工和查看员工列表流程	*
2	案例 1：重定向	掌握并熟练应用重定向，理解其流程	***
3	案例 2：雇员的增减改查	Servelt 中进行数据库操作的综合应用	**
4	什么是 DAO	了解 DAO 的含义	*
5	如何写一个 DAO	掌握 DAO 的编程步骤	**
6	dao 工厂（工厂模式）	理解并掌握	**

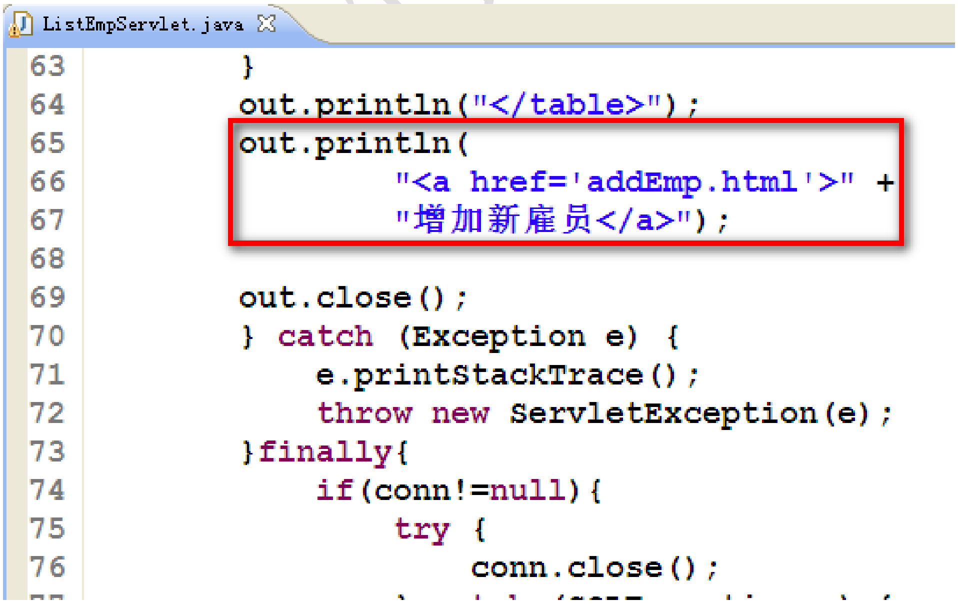
注：	"*"理解级别	"**"掌握级别	"***"应用级别

1. 案例演示 1：改进 ListEmpServlet *

演示

步骤 1

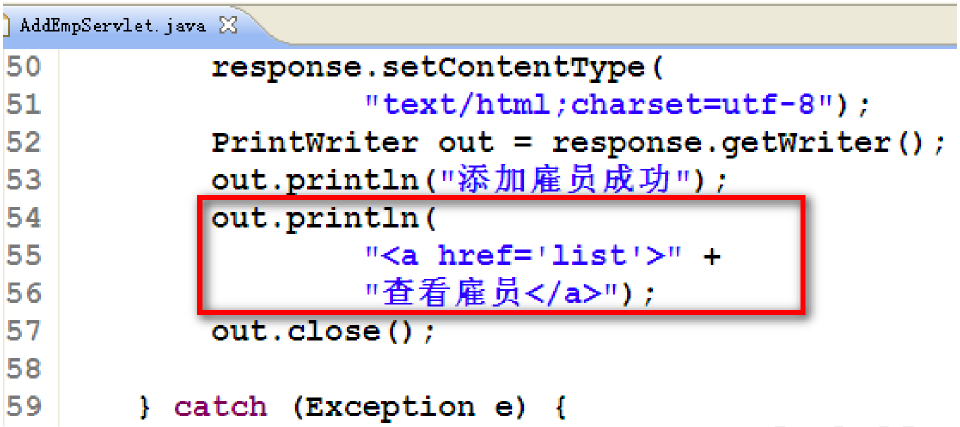
在 ListEmpServlet.java 中添加代码“链接”



```
        }
        out.println("</table>");
        out.println(
                "<a href='addEmp.html'>" +
                "增加新雇员</a>");

        out.close();
        } catch (Exception e) {
            e.printStackTrace();
            throw new ServletException(e);
        }finally{
            if(conn!=null){
                try {
                    conn.close();
```

步骤 2

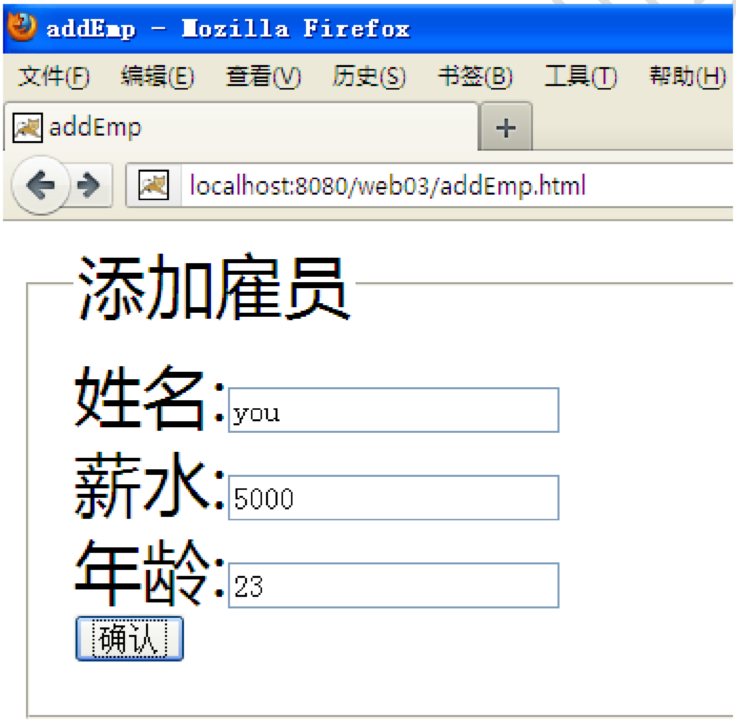
在 AddEmpServlet.java 中添加代码“链接”



```
        response.setContentType(
                "text/html;charset=utf-8");
        PrintWriter out = response.getWriter();
        out.println("添加雇员成功");
        out.println(
                "<a href='list'>" +
                "查看雇员</a>");
        out.close();

    } catch (Exception e) {
```

结果演示

1) 添加新用户



localhost:8080/web03/addEmp.html

添加雇员

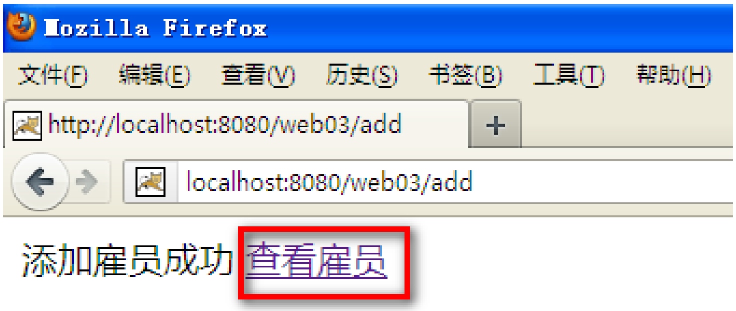
姓名: you

薪水: 5000

年龄: 23

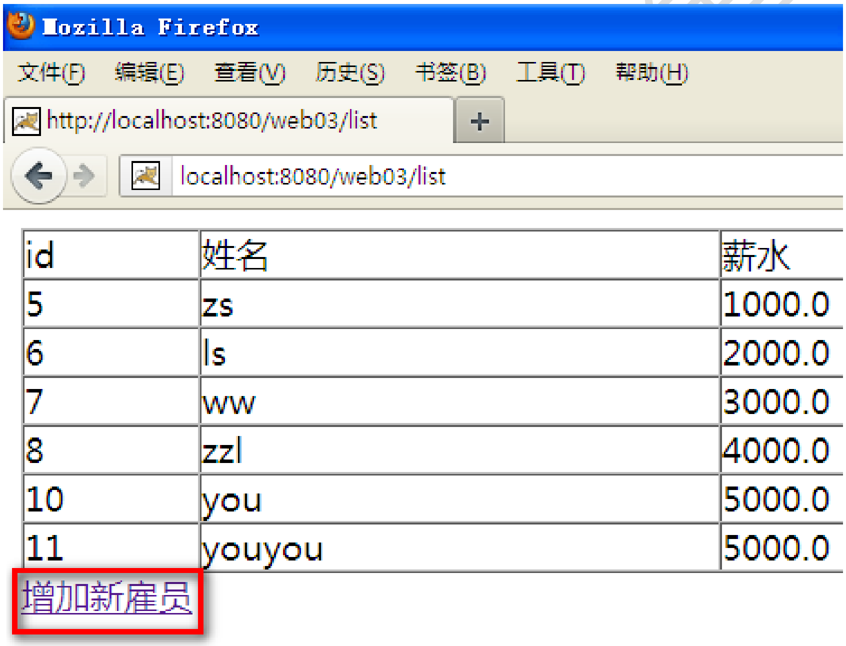
确认

2) 添加成功后显示“添加雇员成功”，点击“查看雇员”链接


Mozilla Firefox
文件(F) 编辑(E) 查看(V) 历史(S) 书签(B) 工具(T) 帮助(H)
http://localhost:8080/web03/add
+
localhost:8080/web03/add

添加雇员成功 查看雇员

3) 返回“查看雇员”页面，可以重新添加新雇员

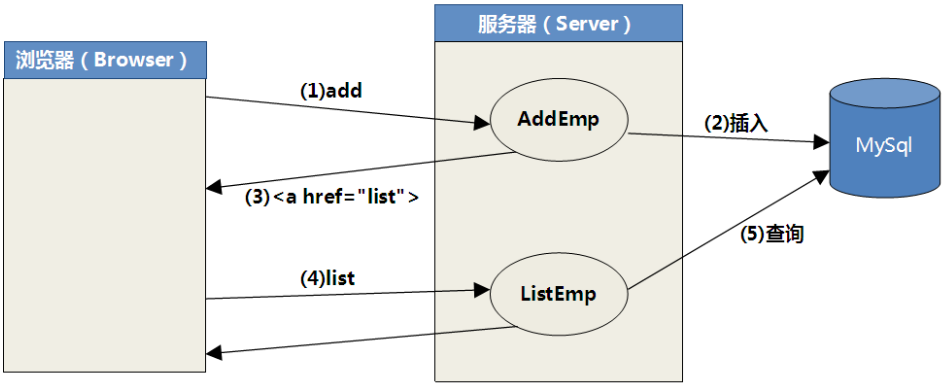

Mozilla Firefox
文件(F) 编辑(E) 查看(V) 历史(S) 书签(B) 工具(T) 帮助(H)
http://localhost:8080/web03/list
+
localhost:8080/web03/list

id	姓名	薪水
5	zs	1000.0
6	ls	2000.0
7	ww	3000.0
8	zzl	4000.0
10	you	5000.0
11	youyou	5000.0

增加新雇员

图示演示

添加员工和查看员工列表流程图

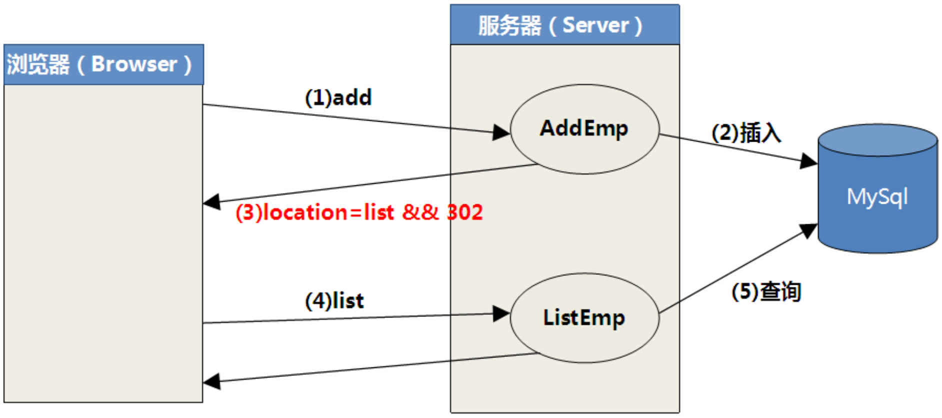


1) 用户在浏览器提交 form 表单，通过 add 到服务器找到 AddEmp
2) AddEmp 向数据库中插入一条数据
3) AddEmp 返回给用户一条信息“数据插入成功”，并且提供一个查看员工信息的链接
4) 用户点击链接<a herf="list">到服务器找到 ListEmp
5) ListEmp 到数据库中查询数据并返回给浏览器

2. 重定向 ***

图示演示

重定向



a. 用户在浏览器提交 form 表单，通过 add 到服务器找到 AddEmp
b. AddEmp 向数据库中插入一条数据
c. **AddEmp 返回给用户一个消息头 location= list（ListEmp 的地址）和 302 状态码**
当浏览器碰到状态码 302 后，浏览器会自动向新地址发请求

d.　浏览器自动通过 list 找到 ListEmp

e.　ListEmp 到数据库中查询数据并返回给浏览器

1)　**什么是重定向?**

服务器向浏览器发送一个状态码 302 及一个消息头 location，浏览器收到后，会立即向 location 所指向的地址发送请求。

2)　**如何重定向**

response.sendRedirect(String url);　　其中,url 是重定向的地址。

3)　**编程要注意的问题**

✓　重定向之前，不能够有 out.close()或者 out.flush()操作。

✓　重定向之前，如果 response 对象缓存有数据，则这些数据会被清空。

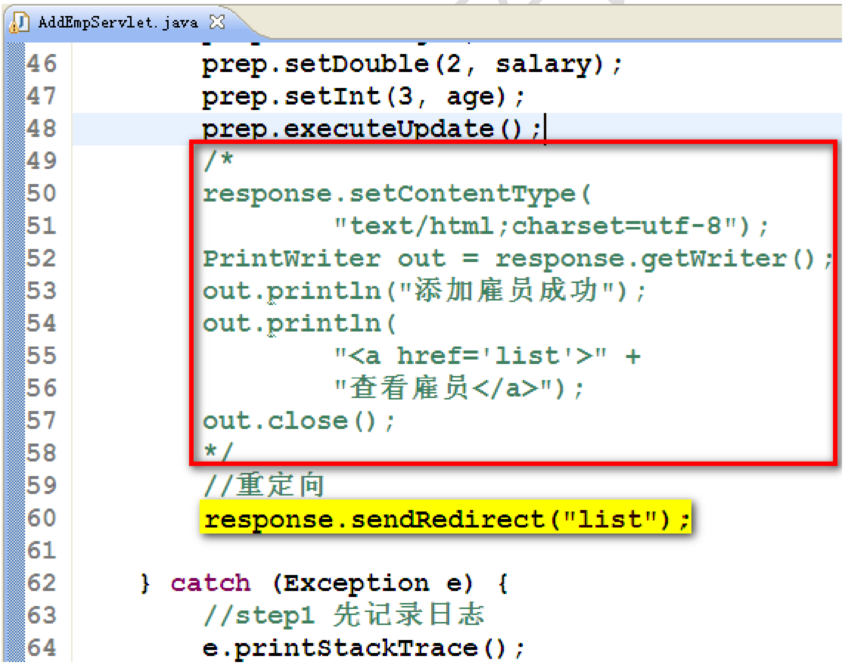
4)　**重定向的特点**

✓　重定向的地址是任意的

✓　重定向之后，浏览器地址栏的地址变成了 location 所指定的地址。

演示

重定向

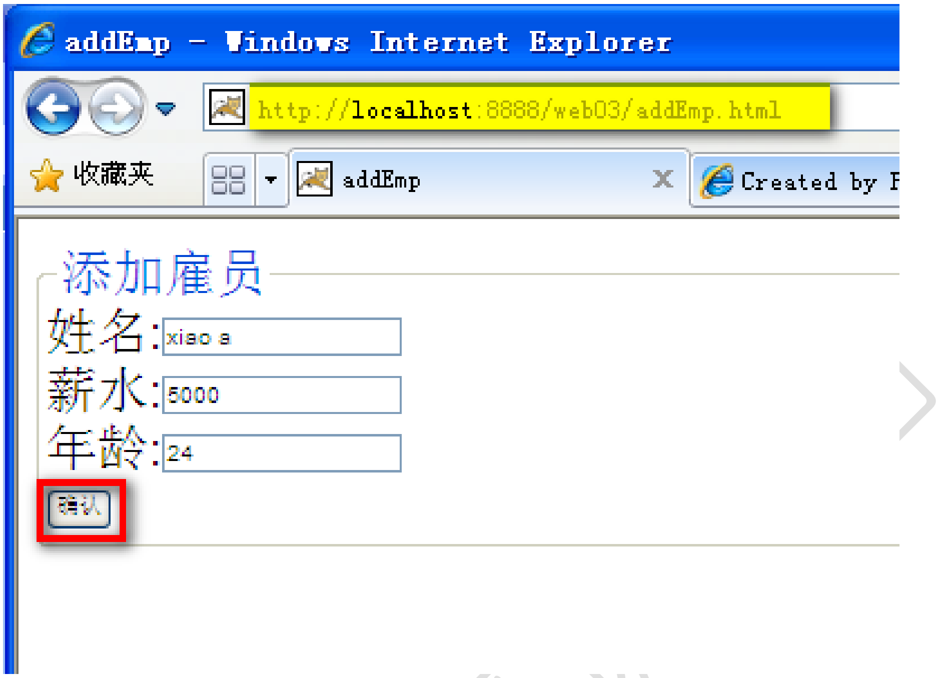


```
            prep.setDouble(2, salary);
            prep.setInt(3, age);
            prep.executeUpdate();
            /*
            response.setContentType(
                    "text/html;charset=utf-8");
            PrintWriter out = response.getWriter();
            out.println("添加雇员成功");
            out.println(
                    "<a href='list'>" +
                    "查看雇员</a>");
            out.close();
            */
            //重定向
            response.sendRedirect("list");

        } catch (Exception e) {
            //step1 先记录日志
            e.printStackTrace();
```

执行结果

1)　**开启 TCP/IP Monitor，访问“http://localhost:8080/web03/addEmp.html”**

点击“确定”提交数据



addEmp - Windows Internet Explorer
http://localhost:8888/web03/addEmp.html
收藏夹
addEmp
Created by F

添加雇员

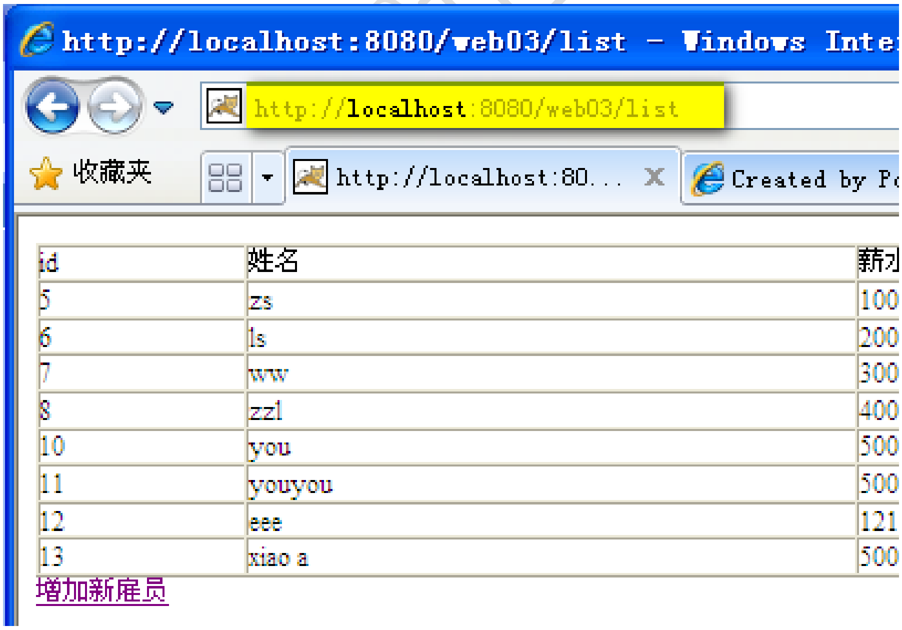
姓名: xiao a

薪水: 5000

年龄: 24

确认

2) 页面直接跳转到“http://localhost:8080/web03/list”

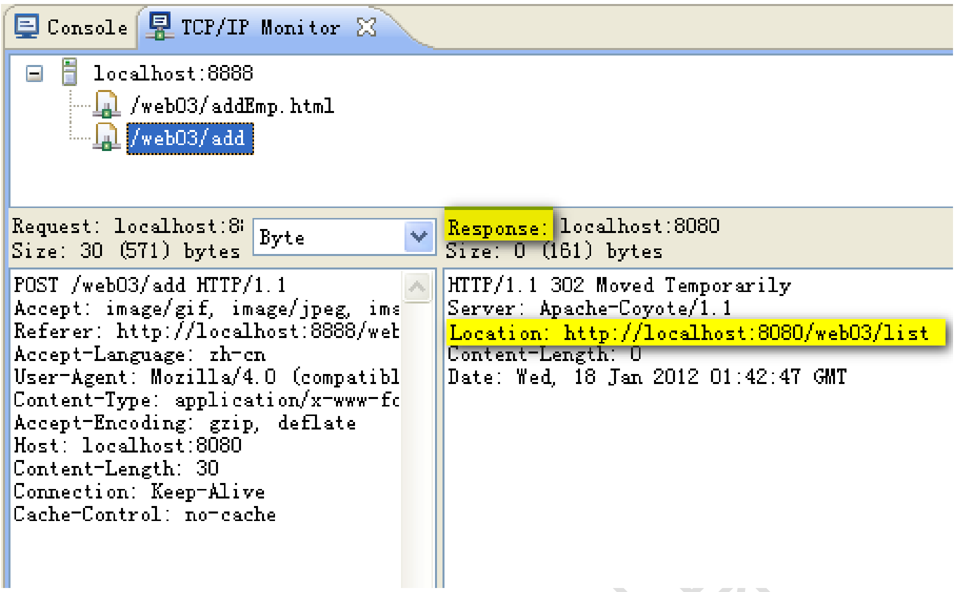


http://localhost:8080/web03/list - Windows Inte
http://localhost:8080/web03/list
收藏夹
http://localhost:80...
Created by Po

id	姓名	薪z
5	zs	100
6	ls	200
7	ww	300
8	zzl	400
10	you	500
11	youyou	500
12	eee	121
13	xiao a	500

增加新雇员

3) TCP/IP Monitor 显示

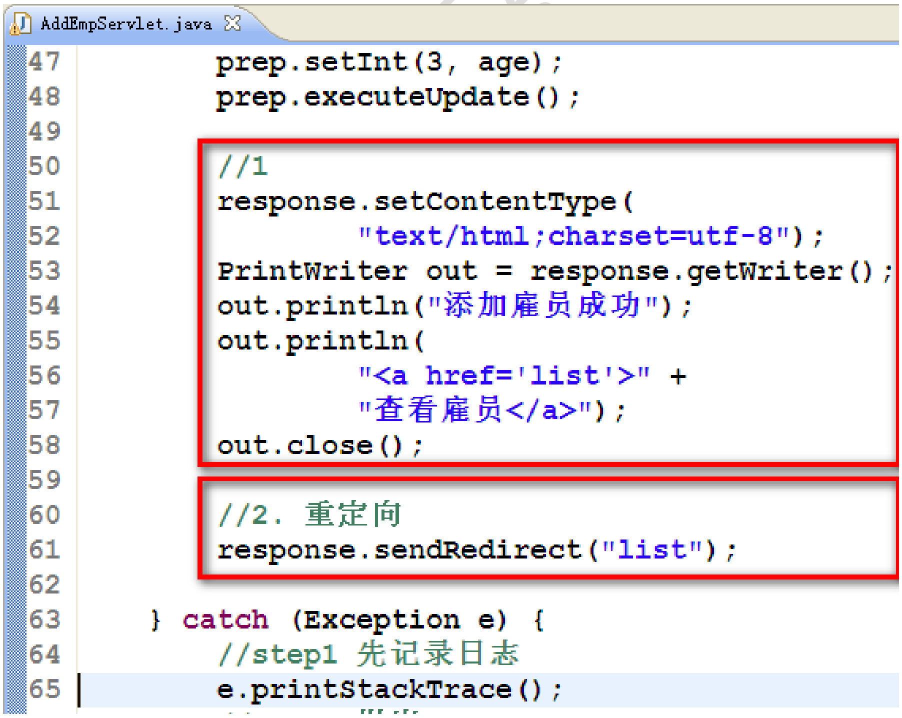


```
Request: localhost:8!
Size: 30 (571) bytes

POST /web03/add HTTP/1.1
Accept: image/gif, image/jpeg, ima
Referer: http://localhost:8888/wek
Accept-Language: zh-cn
User-Agent: Mozilla/4.0 (compatibl
Content-Type: application/x-www-fc
Accept-Encoding: gzip, deflate
Host: localhost:8080
Content-Length: 30
Connection: Keep-Alive
Cache-Control: no-cache
```

```
Response: localhost:8080
Size: 0 (161) bytes

HTTP/1.1 302 Moved Temporarily
Server: Apache-Coyote/1.1
Location: http://localhost:8080/web03/list
Content-Length: 0
Date: Wed, 18 Jan 2012 01:42:47 GMT
```

需要注意的问题 01

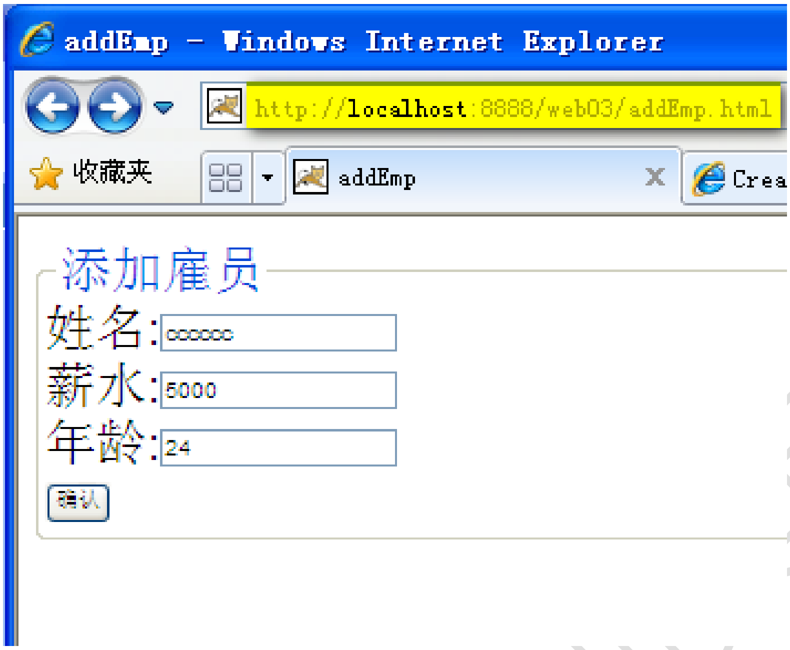
重定向之前，不能够有 out.close()或者 out.flush()操作，演示如下所示：



```java
            prep.setInt(3, age);
            prep.executeUpdate();

            //1
            response.setContentType(
                    "text/html;charset=utf-8");
            PrintWriter out = response.getWriter();
            out.println("添加雇员成功");
            out.println(
                    "<a href='list'>" +
                    "查看雇员</a>");
            out.close();

            //2. 重定向
            response.sendRedirect("list");

        } catch (Exception e) {
            //step1 先记录日志
            e.printStackTrace();
```

执行结果

1) 访问 addEmp.html



addEmp - Windows Internet Explorer

http://localhost:8888/web03/addEmp.html

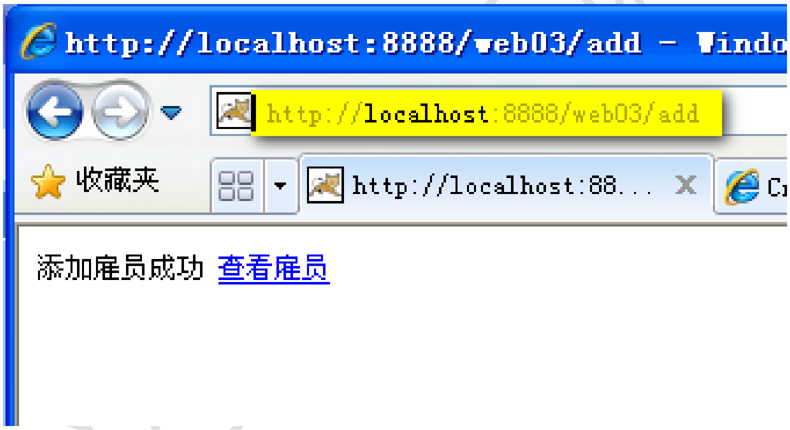
添加雇员

姓名: oooooo

薪水: 5000

年龄: 24

2) 重定向无效

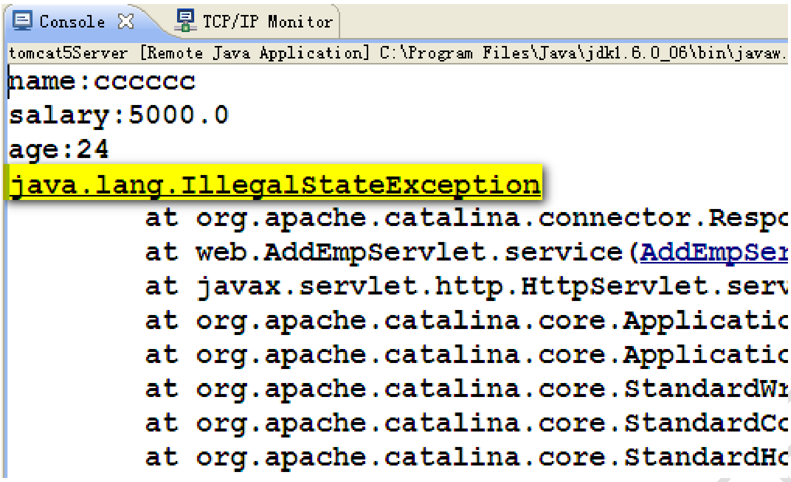


http://localhost:8888/web03/add - Windo

http://localhost:8888/web03/add

添加雇员成功 查看雇员

3) 并且出现异常

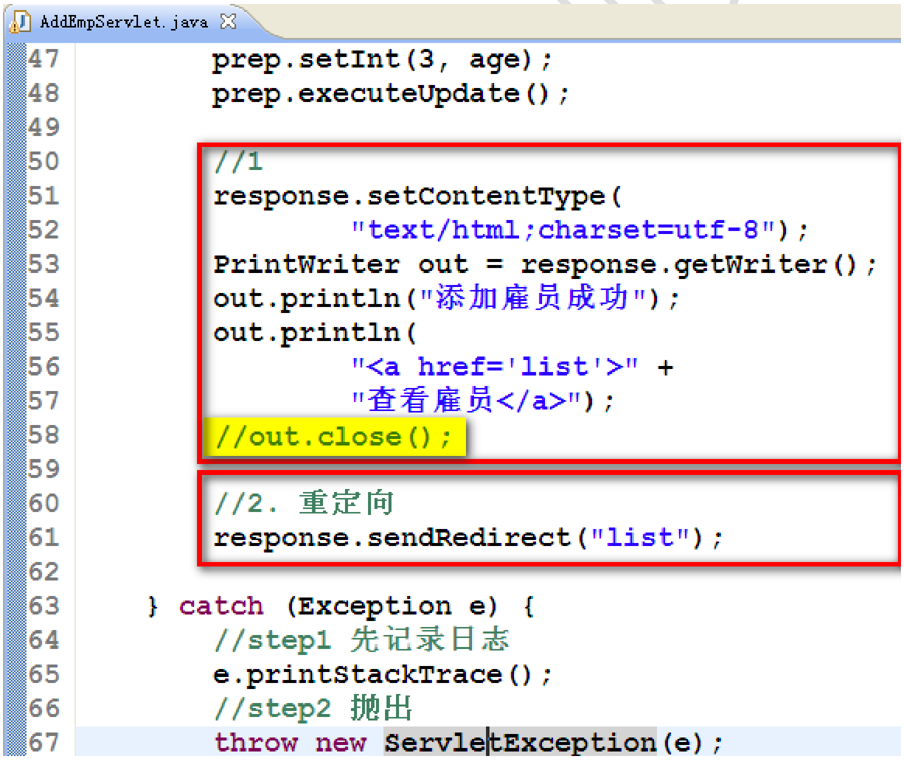


Console ☒ TCP/IP Monitor

```
tomcat5Server [Remote Java Application] C:\Program Files\Java\jdk1.6.0_06\bin\javaw.
name:cccccc
salary:5000.0
age:24
java.lang.IllegalStateException
	at org.apache.catalina.connector.Respc
	at web.AddEmpServlet.service(AddEmpSer
	at javax.servlet.http.HttpServlet.serv
	at org.apache.catalina.core.Applicatic
	at org.apache.catalina.core.Applicatic
	at org.apache.catalina.core.StandardWr
	at org.apache.catalina.core.StandardCc
	at org.apache.catalina.core.StandardHc
```

需要注意的问题 02

重定向之前，如果 response 对象缓存有数据，则这些数据会被清空

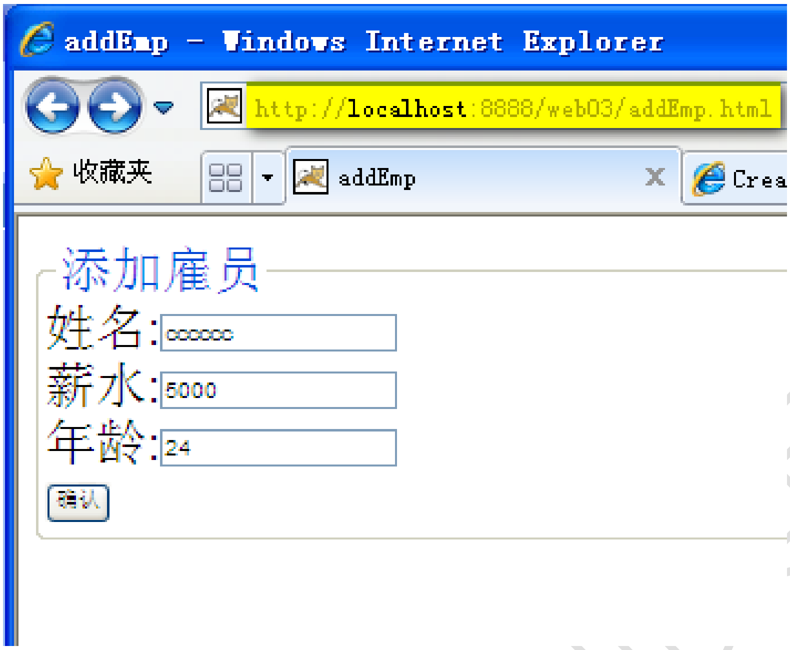


AddEmpServlet.java ☒

```
        prep.setInt(3, age);
        prep.executeUpdate();

        //1
        response.setContentType(
                "text/html;charset=utf-8");
        PrintWriter out = response.getWriter();
        out.println("添加雇员成功");
        out.println(
                "<a href='list'>" +
                "查看雇员</a>");
        //out.close();

        //2. 重定向
        response.sendRedirect("list");

    } catch (Exception e) {
        //step1 先记录日志
        e.printStackTrace();
        //step2 抛出
        throw new ServletException(e);
```

执行结果

1) 访问 addEmp.html



addEmp - Windows Internet Explorer

http://localhost:8888/web03/addEmp.html

添加雇员

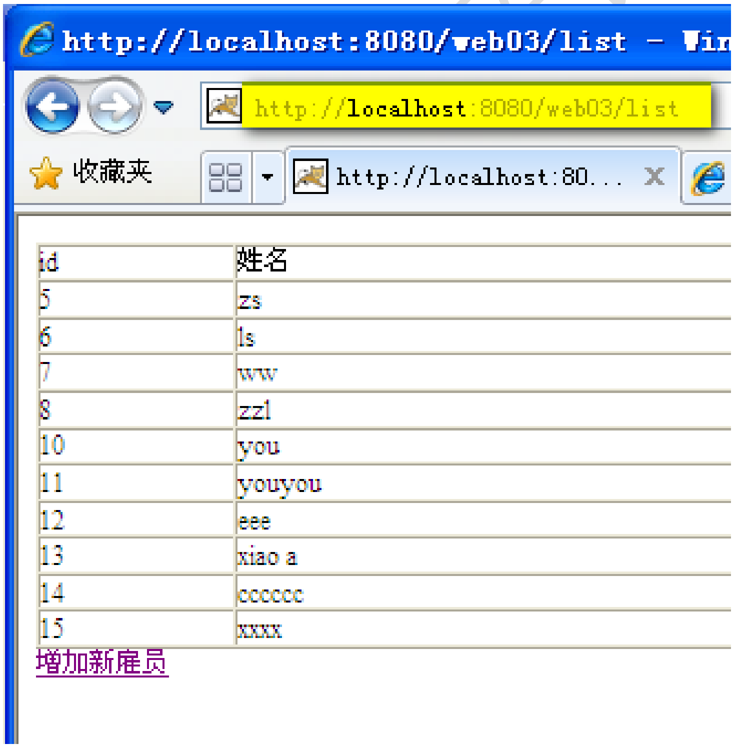
姓名: cccccc

薪水: 5000

年龄: 24

2)　可以重定向

可以重定向，因为 response 中的数据在重定向前被清空了，out.println(...)没有起作用



http://localhost:8080/web03/list - Win

http://localhost:8080/web03/list

id	姓名
5	zs
6	ls
7	ww
8	zzl
10	you
11	youyou
12	eee
13	xiao a
14	cccccc
15	xxxx

增加新雇员

【案例 1】重定向

1) AddEmpServlet.java

```
package web;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class AddEmpServlet extends HttpServlet{

   public void service(HttpServletRequest request,
                                 HttpServletResponse response)
      throws ServletException,IOException{

       //这行代码要放在 getParameter()执行之前。
       request.setCharacterEncoding("utf-8");

       String name = request.getParameter("name");
       double salary = Double.parseDouble(
                                 request.getParameter("salary"));
       int age = Integer.parseInt(
                        request.getParameter("age"));
       System.out.println("name:" + name);
       System.out.println("salary:" + salary);
       System.out.println("age:" + age);

       //访问数据库
       Connection conn = null;
       try {
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager
            .getConnection(
            "jdbc:mysql://localhost:3306/jd1109db2",
            "root","root");
        PreparedStatement prep =
            conn.prepareStatement(
            "insert into t_emp(name,salary,age) " +
            "values(?,?,?)");
        prep.setString(1, name);
        prep.setDouble(2, salary);
        prep.setInt(3, age);
        prep.executeUpdate();
        /*
        response.setContentType(
                "text/html;charset=utf-8");
        PrintWriter out = response.getWriter();
        out.println("添加雇员成功");
        out.println(
                "<a href='list'>" +
                "查看雇员</a>");
        out.close();
        */
        //重定向
        response.sendRedirect("list");

    } catch (Exception e) {
        //step1 先记录日志
        e.printStackTrace();
        //step2 抛出
        throw new ServletException(e);

    }finally{
        if(conn!=null){
            try {
                conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
```

```
                    }
                }
            }
        }
}
```

2) ListEmpServlet.java

```
package web;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ListEmpServlet extends HttpServlet{
    public void service(HttpServletRequest request,
                                    HttpServletResponse response)
        throws ServletException,IOException{
        //访问数据库
        Connection conn = null;
        try {
                Class.forName(
                        "com.mysql.jdbc.Driver");
                conn =
                        DriverManager.getConnection(
                        "jdbc:mysql://localhost:3306/jd1109db2",
                        "root","root");
                Statement stat =
                        conn.createStatement();
```

```
ResultSet rst = stat.executeQuery(
        "select * from t_emp");

//使用查询得到的结果，生成一个表格
response.setContentType(
        "text/html;charset=utf-8");
PrintWriter out =
    response.getWriter();
out.println(
        "<table border='1' " +
        "width='60%' " +
        "cellpadding='0' " +
        "cellspacing='0'>");
out.println("<tr>" +
        "<td>id</td>" +
        "<td>姓名</td>" +
        "<td>薪水</td>" +
        "<td>年龄</td></tr>");
while(rst.next()){
    long id =
        rst.getLong("id");
    String name =
        rst.getString("name");
    double salary =
        rst.getDouble("salary");
    int age =
        rst.getInt("age");
    out.println("<tr><td>"
            + id + "</td><td> "
            + name + "</td><td>"
            + salary + "</td><td> "
            + age + "</td></tr>");
}
out.println("</table>");
out.println(
        "<a href='addEmp.html'>" +
        "增加新雇员</a>");
```

```
		out.close();
		} catch (Exception e) {
			e.printStackTrace();
			throw new ServletException(e);
		}finally{
			if(conn!=null){
				try {
					conn.close();
				} catch (SQLException e) {
					e.printStackTrace();
				}
			}
		}
	}
}
```

3) addEmp.html

```
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=utf-8">
<title>addEmp</title>
</head>
<body style="font-size:30px;">
	<form action="add" method="post">
		<fieldset>
			<legend>添加雇员</legend>
			姓名:<input name="name"/><br/>
			薪水:<input name="salary"/><br/>
			年龄:<input name="age"/><br/>
			<input type="submit" value="确认"/>
		</fieldset>
	</form>
</body>
</html>
```

4) err.html

```
<html>
<head>
<meta http-equiv="Content-Type"
      content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body style="font-size:30px;color:red;">
      发生了系统错误，请稍后
      <a href="addEmp.html">重试</a>
</body>
</html>
```

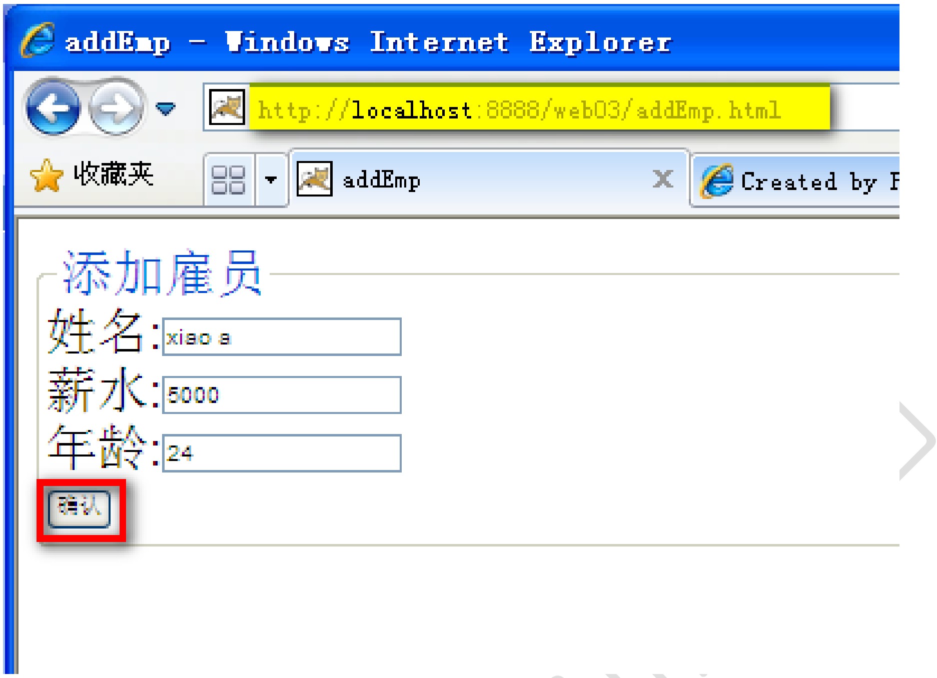
5) web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4"
      xmlns=
      "http://java.sun.com/xml/ns/j2ee"
      xmlns:xsi=
      "http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation=
      "http://java.sun.com/xml/ns/j2ee
      http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
      <servlet>
            <servlet-name>
                  addEmpServlet</servlet-name>
            <servlet-class>
                  web.AddEmpServlet</servlet-class>
      </servlet>
      <servlet>
            <servlet-name>
                  listEmpServlet</servlet-name>
            <servlet-class>
                  web.ListEmpServlet</servlet-class>
      </servlet>
      <servlet-mapping>
            <servlet-name>addEmpServlet</servlet-name>
            <url-pattern>/add</url-pattern>
```

```
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>
            listEmpServlet</servlet-name>
        <url-pattern>
            /list</url-pattern>
    </servlet-mapping>
    <error-page>
        <!-- 方式1
        <error-code>500</error-code>-->
        <!-- 方式2 -->
        <exception-type>
        javax.servlet.ServletException
        </exception-type>
        <location>/error.html</location>
    </error-page>
</web-app>
```

执行结果

1) **开启 TCP/IP Monitor，访问“http://localhost:8888/web03/addEmp.html”点击“确定”提交数据**


addEmp - Windows Internet Explorer
http://localhost:8888/web03/addEmp.html
收藏夹
addEmp
Created by F

添加雇员

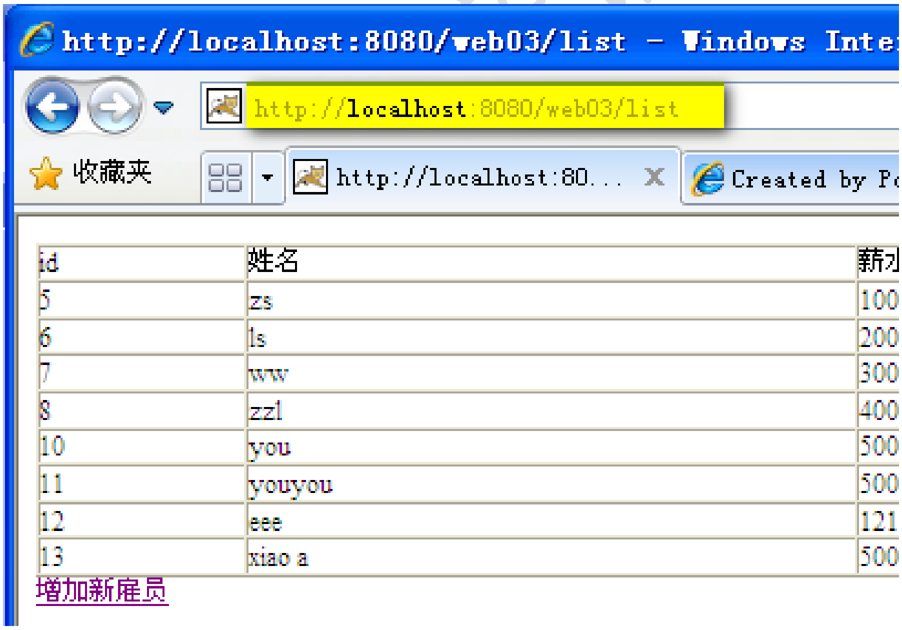
姓名: xiao a

薪水: 5000

年龄: 24

确认

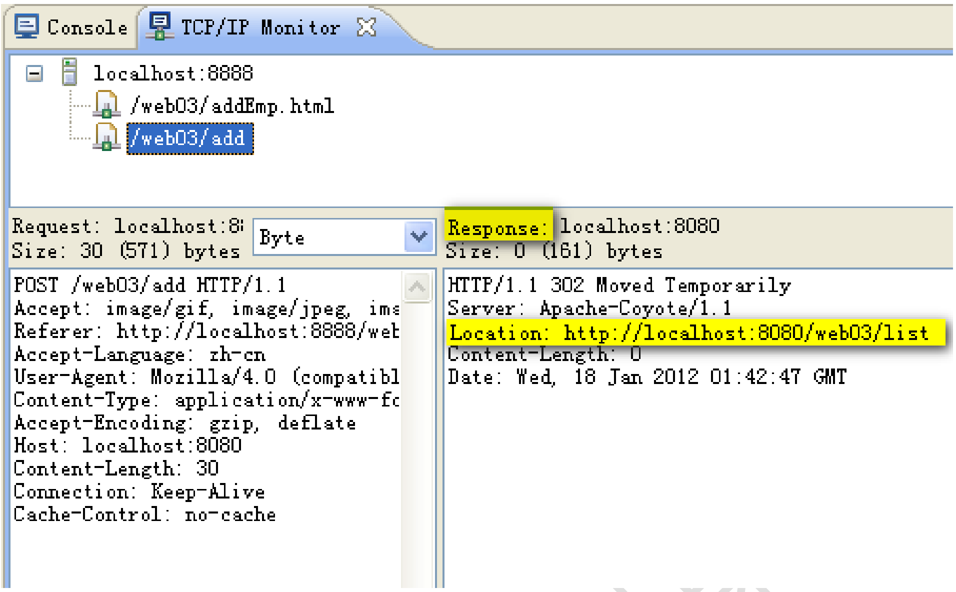
2) 页面直接跳转到“http://localhost:8080/web03/list”


http://localhost:8080/web03/list - Windows Inte
http://localhost:8080/web03/list
收藏夹
http://localhost:80...
Created by Po

id	姓名	薪水
5	zs	100
6	ls	200
7	ww	300
8	zzl	400
10	you	500
11	youyou	500
12	eee	121
13	xiao a	500

增加新雇员

3) TCP/IP Monitor 显示

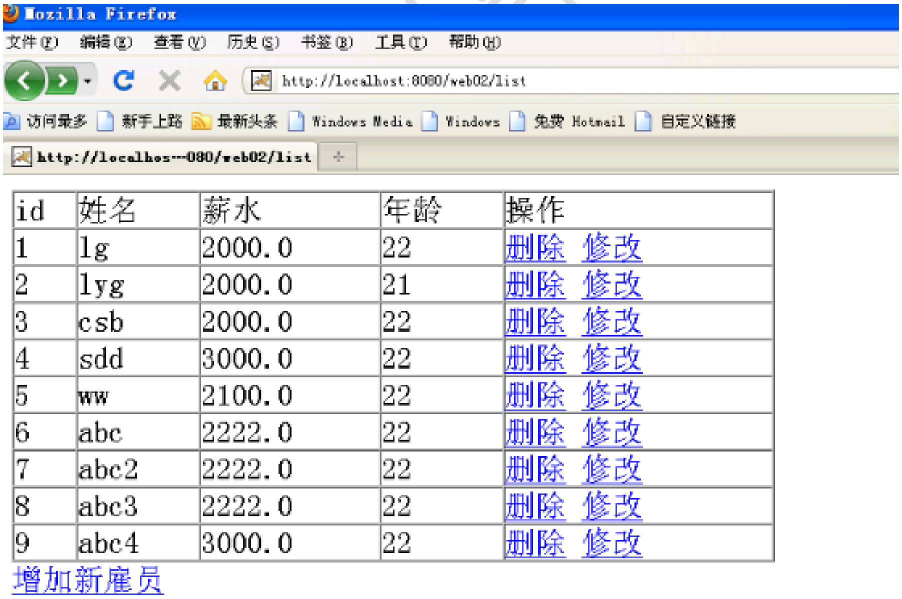


```
localhost:8888
    /web03/addEmp.html
    /web03/add
```

```
Request: localhost:8(
Size: 30 (571) bytes
POST /web03/add HTTP/1.1
Accept: image/gif, image/jpeg, ima
Referer: http://localhost:8888/web
Accept-Language: zh-cn
User-Agent: Mozilla/4.0 (compatibl
Content-Type: application/x-www-fo
Accept-Encoding: gzip, deflate
Host: localhost:8080
Content-Length: 30
Connection: Keep-Alive
Cache-Control: no-cache
```

```
Response: localhost:8080
Size: 0 (161) bytes
HTTP/1.1 302 Moved Temporarily
Server: Apache-Coyote/1.1
Location: http://localhost:8080/web03/list
Content-Length: 0
Date: Wed, 18 Jan 2012 01:42:47 GMT
```

3. 案例演示 2：雇员的删改 **


Mozilla Firefox
文件(F) 编辑(E) 查看(V) 历史(S) 书签(B) 工具(T) 帮助(H)
http://localhost:8080/web02/list
访问最多 新手上路 最新头条 Windows Media Windows 免费 Hotmail 自定义链接
http://localhos⋯080/web02/list

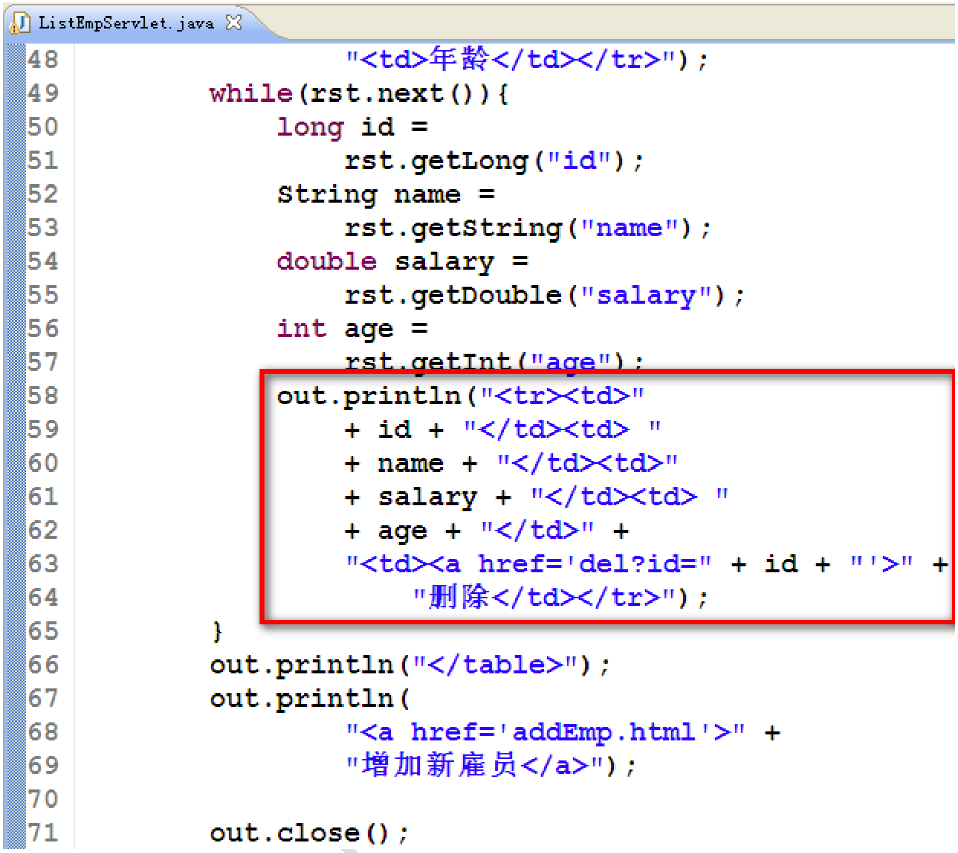
id	姓名	薪水	年龄	操作
1	lg	2000.0	22	删除 修改
2	lyg	2000.0	21	删除 修改
3	csb	2000.0	22	删除 修改
4	sdd	3000.0	22	删除 修改
5	ww	2100.0	22	删除 修改
6	abc	2222.0	22	删除 修改
7	abc2	2222.0	22	删除 修改
8	abc3	2222.0	22	删除 修改
9	abc4	3000.0	22	删除 修改

增加新雇员

3.1. 步骤演示

演示

1)　修改 ListEmpServlet.java 添加“删除”链接

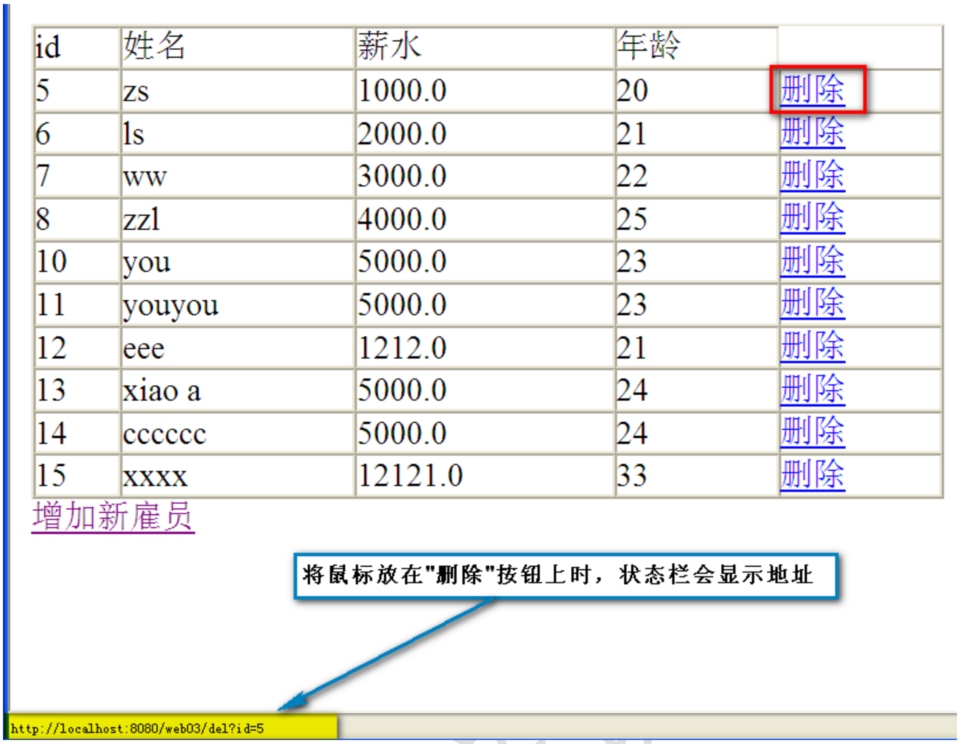


ListEmpServlet.java

```
                "<td>年龄</td></tr>");
        while(rst.next()){
            long id =
                rst.getLong("id");
            String name =
                rst.getString("name");
            double salary =
                rst.getDouble("salary");
            int age =
                rst.getInt("age");
            out.println("<tr><td>"
                + id + "</td><td> "
                + name + "</td><td>"
                + salary + "</td><td> "
                + age + "</td>" +
                "<td><a href='del?id=" + id + "'>" +
                    "删除</td></tr>");
        }
        out.println("</table>");
        out.println(
                "<a href='addEmp.html'>" +
                "增加新雇员</a>");

        out.close();
```

2)　测试

重新部署，并访问“http://localhost:8080/web03/list”



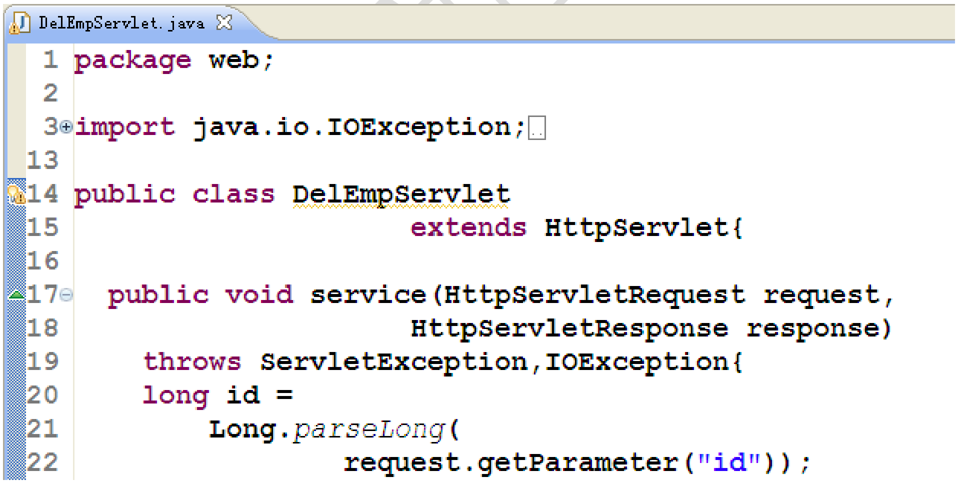
id	姓名	薪水	年龄	
5	zs	1000.0	20	删除
6	ls	2000.0	21	删除
7	ww	3000.0	22	删除
8	zzl	4000.0	25	删除
10	you	5000.0	23	删除
11	youyou	5000.0	23	删除
12	eee	1212.0	21	删除
13	xiao a	5000.0	24	删除
14	cccccc	5000.0	24	删除
15	xxxx	12121.0	33	删除

增加新雇员

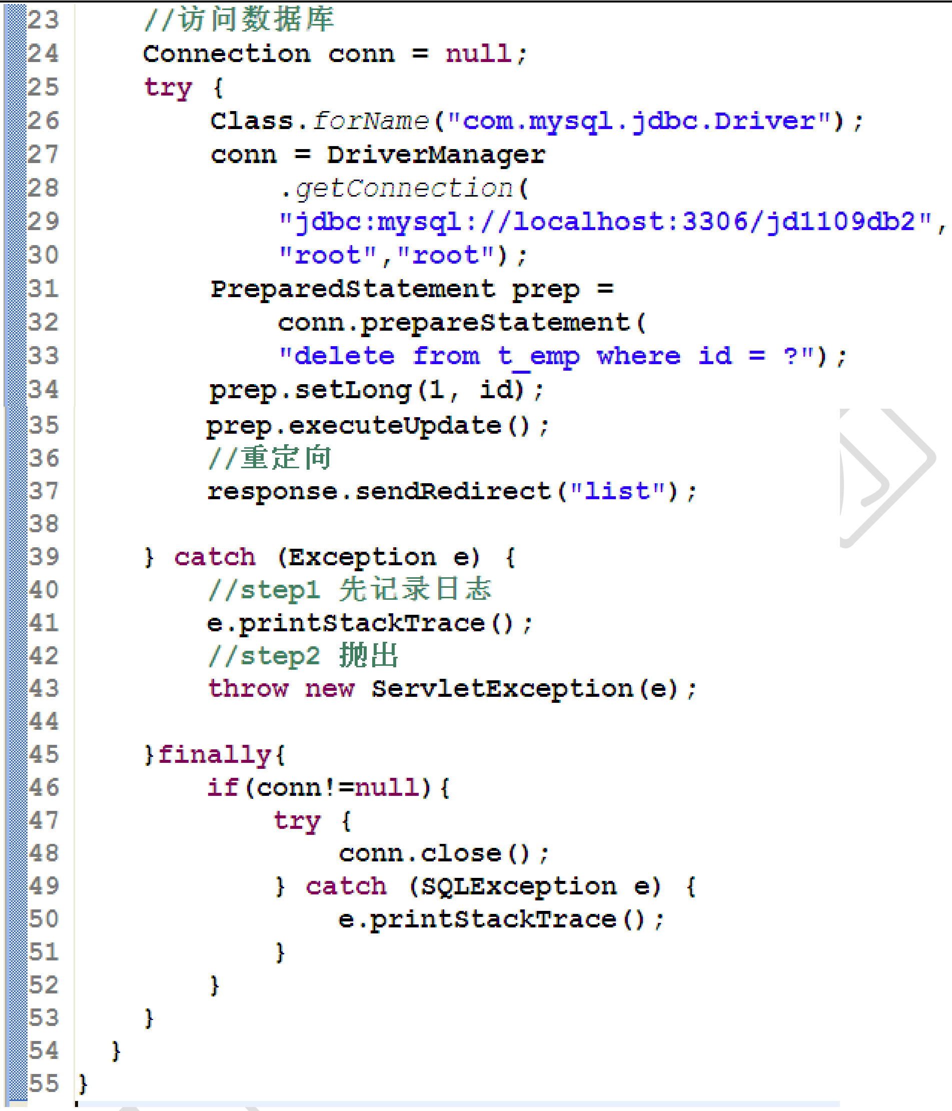
将鼠标放在"删除"按钮上时，状态栏会显示地址

http://localhost:8080/web03/del?id=5

3) 新建 DelEmpServlet.java

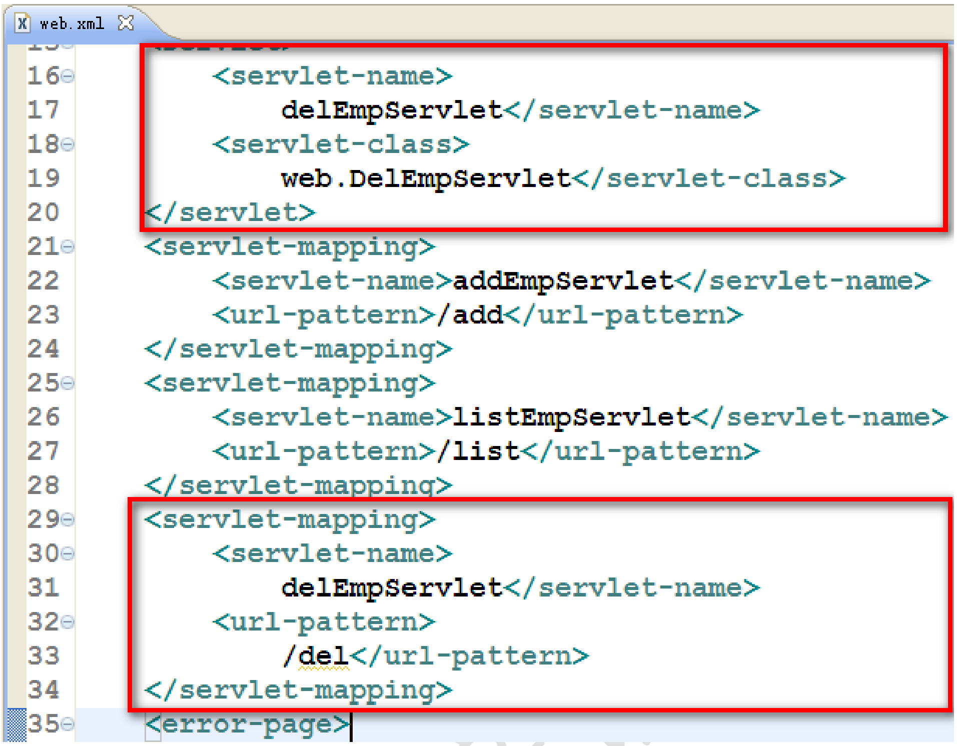


```
package web;

import java.io.IOException;

public class DelEmpServlet
                    extends HttpServlet{

  public void service(HttpServletRequest request,
                    HttpServletResponse response)
    throws ServletException,IOException{
    long id =
        Long.parseLong(
                request.getParameter("id"));
```



```
        //访问数据库
        Connection conn = null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager
                .getConnection(
                "jdbc:mysql://localhost:3306/jd1109db2",
                "root","root");
            PreparedStatement prep =
                conn.prepareStatement(
                "delete from t_emp where id = ?");
            prep.setLong(1, id);
            prep.executeUpdate();
            //重定向
            response.sendRedirect("list");

        } catch (Exception e) {
            //step1 先记录日志
            e.printStackTrace();
            //step2 抛出
            throw new ServletException(e);

        }finally{
            if(conn!=null){
                try {
                    conn.close();
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

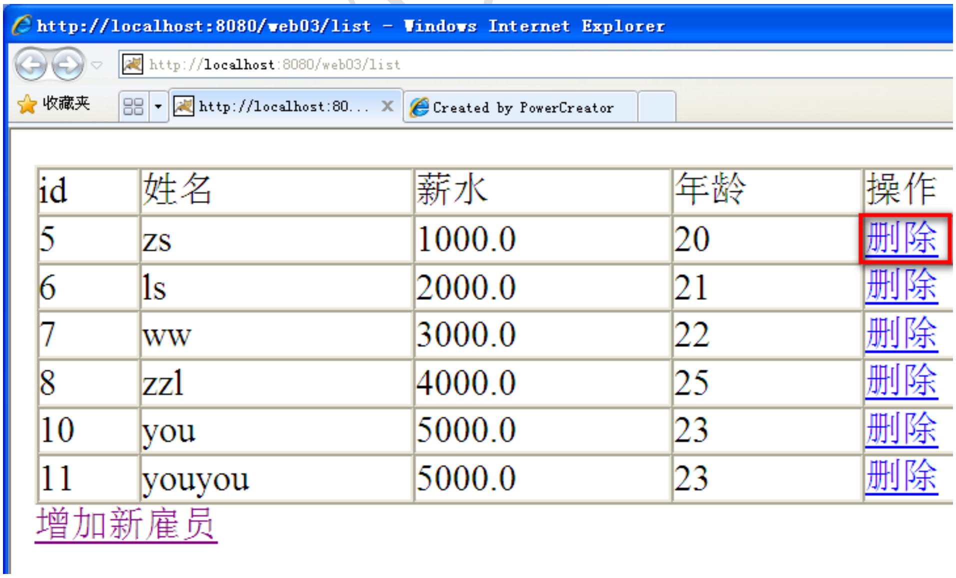
4) 配置 web.xml



```
    <servlet-name>
        delEmpServlet</servlet-name>
    <servlet-class>
        web.DelEmpServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>addEmpServlet</servlet-name>
    <url-pattern>/add</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>listEmpServlet</servlet-name>
    <url-pattern>/list</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>
        delEmpServlet</servlet-name>
    <url-pattern>
        /del</url-pattern>
</servlet-mapping>
<error-page>
```

5)　测试

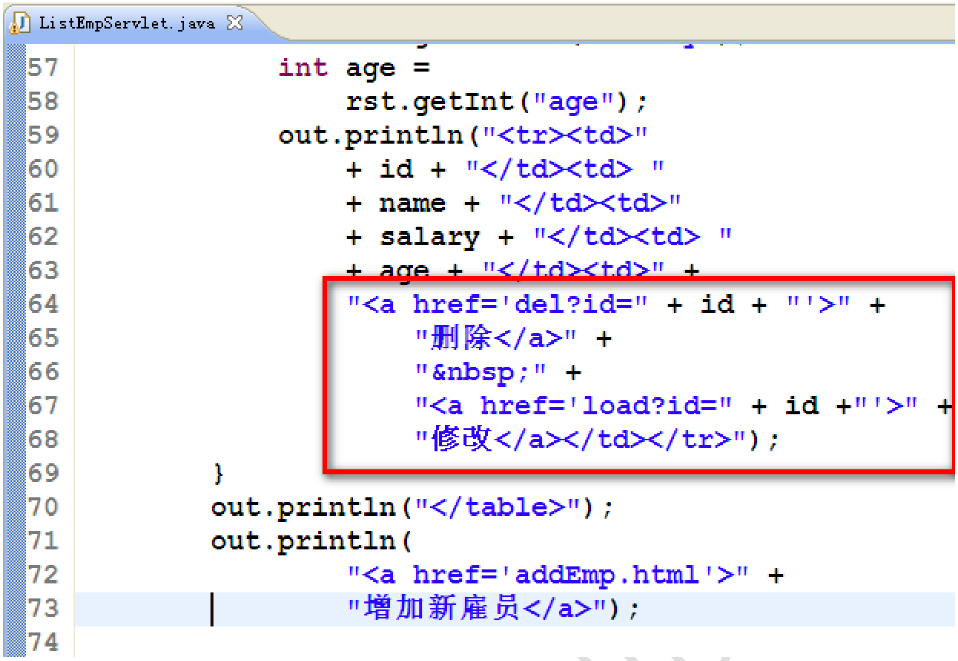
点击“删除”即可在数据库中删除该条数据


http://localhost:8080/web03/list - Windows Internet Explorer
http://localhost:8080/web03/list
收藏夹
http://localhost:80...
Created by PowerCreator

id	姓名	薪水	年龄	操作
5	zs	1000.0	20	删除
6	ls	2000.0	21	删除
7	ww	3000.0	22	删除
8	zzl	4000.0	25	删除
10	you	5000.0	23	删除
11	youyou	5000.0	23	删除

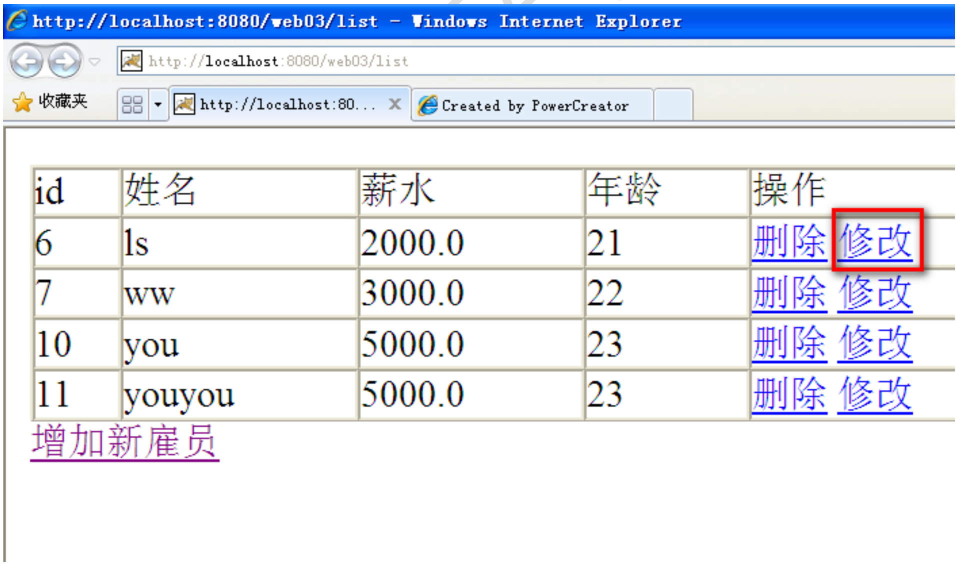
增加新雇员

6)　为 ListEmpServlet.java 添加“修改”链接



```
          int age =
              rst.getInt("age");
          out.println("<tr><td>"
              + id + "</td><td> "
              + name + "</td><td>"
              + salary + "</td><td> "
              + age + "</td><td>" +
              "<a href='del?id=" + id + "'>" +
                  "删除</a>" +
                  "&nbsp;" +
                  "<a href='load?id=" + id +"'>" +
                  "修改</a></td></tr>");
        }
        out.println("</table>");
        out.println(
              "<a href='addEmp.html'>" +
              "增加新雇员</a>");
```

7) 测试

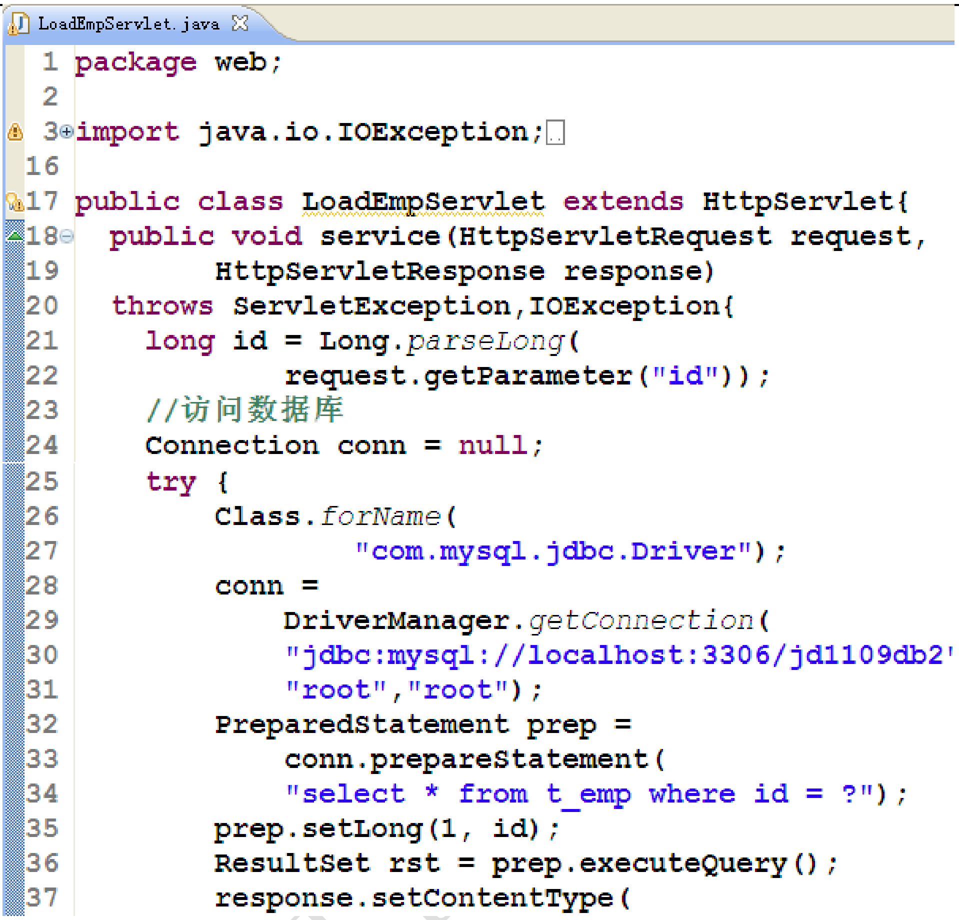


http://localhost:8080/web03/list - Windows Internet Explorer
http://localhost:8080/web03/list
收藏夹
http://localhost:80...
Created by PowerCreator

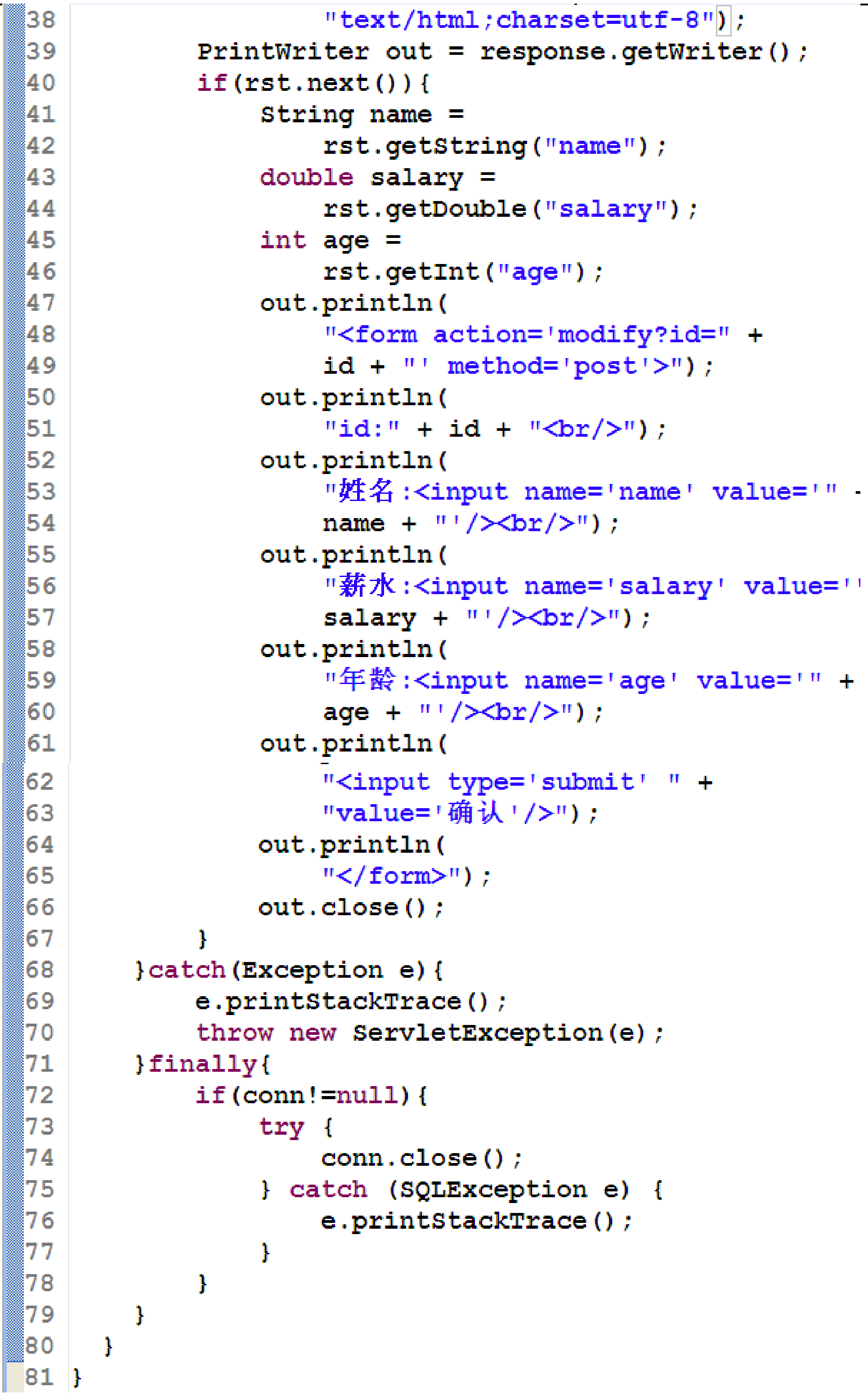
id	姓名	薪水	年龄	操作
6	ls	2000.0	21	删除 修改
7	ww	3000.0	22	删除 修改
10	you	5000.0	23	删除 修改
11	youyou	5000.0	23	删除 修改

增加新雇员

8) 新建 LoadEmpServlet.java

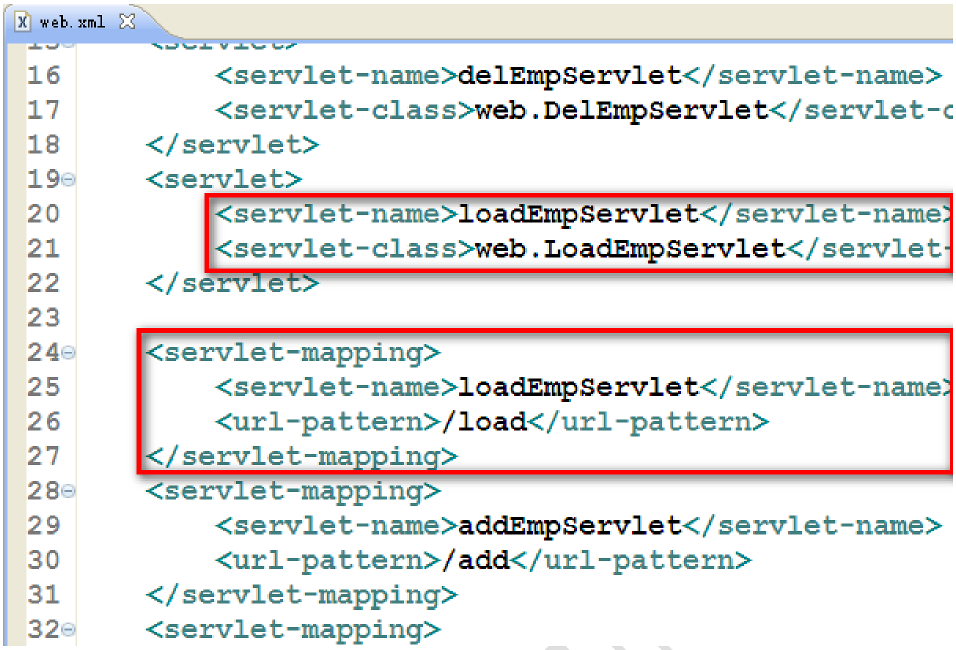


```
package web;

import java.io.IOException;□

public class LoadEmpServlet extends HttpServlet{
  public void service(HttpServletRequest request,
        HttpServletResponse response)
  throws ServletException,IOException{
    long id = Long.parseLong(
            request.getParameter("id"));
    //访问数据库
    Connection conn = null;
    try {
        Class.forName(
                "com.mysql.jdbc.Driver");
        conn =
            DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/jd1109db2"
            "root","root");
        PreparedStatement prep =
            conn.prepareStatement(
            "select * from t_emp where id = ?");
        prep.setLong(1, id);
        ResultSet rst = prep.executeQuery();
        response.setContentType(
```



```
                "text/html;charset=utf-8");
        PrintWriter out = response.getWriter();
        if(rst.next()){
            String name =
                rst.getString("name");
            double salary =
                rst.getDouble("salary");
            int age =
                rst.getInt("age");
            out.println(
                "<form action='modify?id=" +
                id + "' method='post'>");
            out.println(
                "id:" + id + "<br/>");
            out.println(
                "姓名:<input name='name' value='"
                name + "'/><br/>");
            out.println(
                "薪水:<input name='salary' value='
                salary + "'/><br/>");
            out.println(
                "年龄:<input name='age' value='" +
                age + "'/><br/>");
            out.println(
                "<input type='submit' " +
                "value='确认'/>");
            out.println(
                "</form>");
            out.close();
        }
    }catch(Exception e){
        e.printStackTrace();
        throw new ServletException(e);
    }finally{
        if(conn!=null){
            try {
                conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
  }
}
```

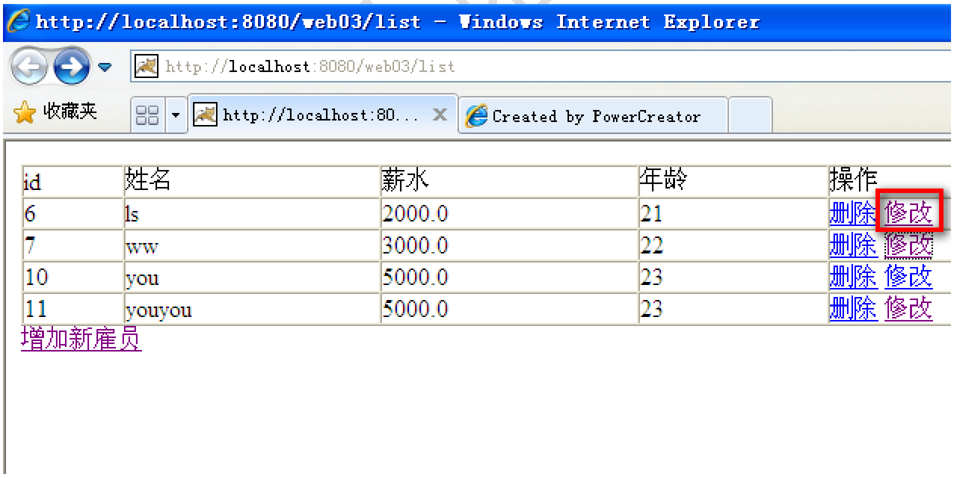
9) 修改 web.xml



```
    <servlet>
        <servlet-name>delEmpServlet</servlet-name>
        <servlet-class>web.DelEmpServlet</servlet-c
    </servlet>
    <servlet>
        <servlet-name>loadEmpServlet</servlet-name>
        <servlet-class>web.LoadEmpServlet</servlet-
    </servlet>

    <servlet-mapping>
        <servlet-name>loadEmpServlet</servlet-name>
        <url-pattern>/load</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>addEmpServlet</servlet-name>
        <url-pattern>/add</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
```

10) 测试

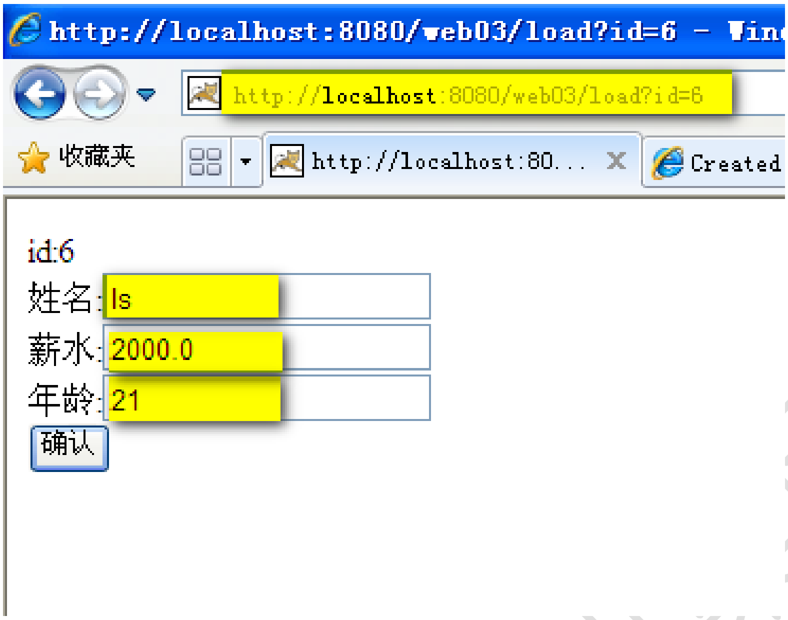
点击修改


http://localhost:8080/web03/list - Windows Internet Explorer
http://localhost:8080/web03/list
收藏夹
http://localhost:80...
Created by PowerCreator

id	姓名	薪水	年龄	操作
6	ls	2000.0	21	删除 修改
7	ww	3000.0	22	删除 修改
10	you	5000.0	23	删除 修改
11	youyou	5000.0	23	删除 修改

增加新雇员

跳转到 LoadEmpServlet.java


http://localhost:8080/web03/load?id=6 - Win
http://localhost:8080/web03/load?id=6
收藏夹
http://localhost:80...
Created

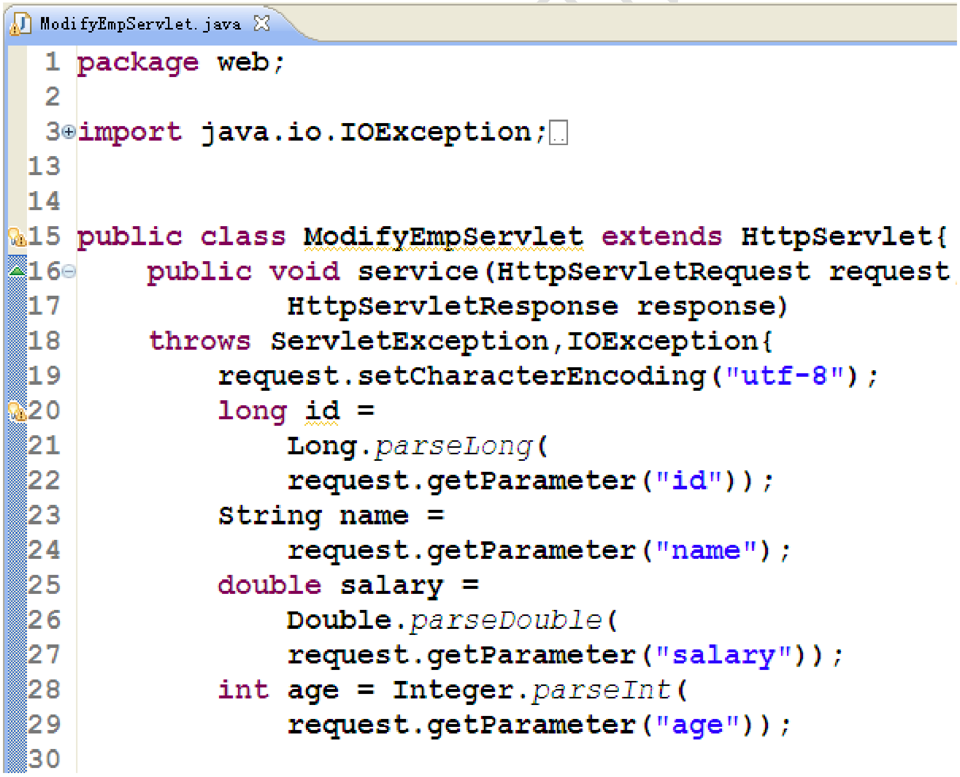
id:6

姓名: ls

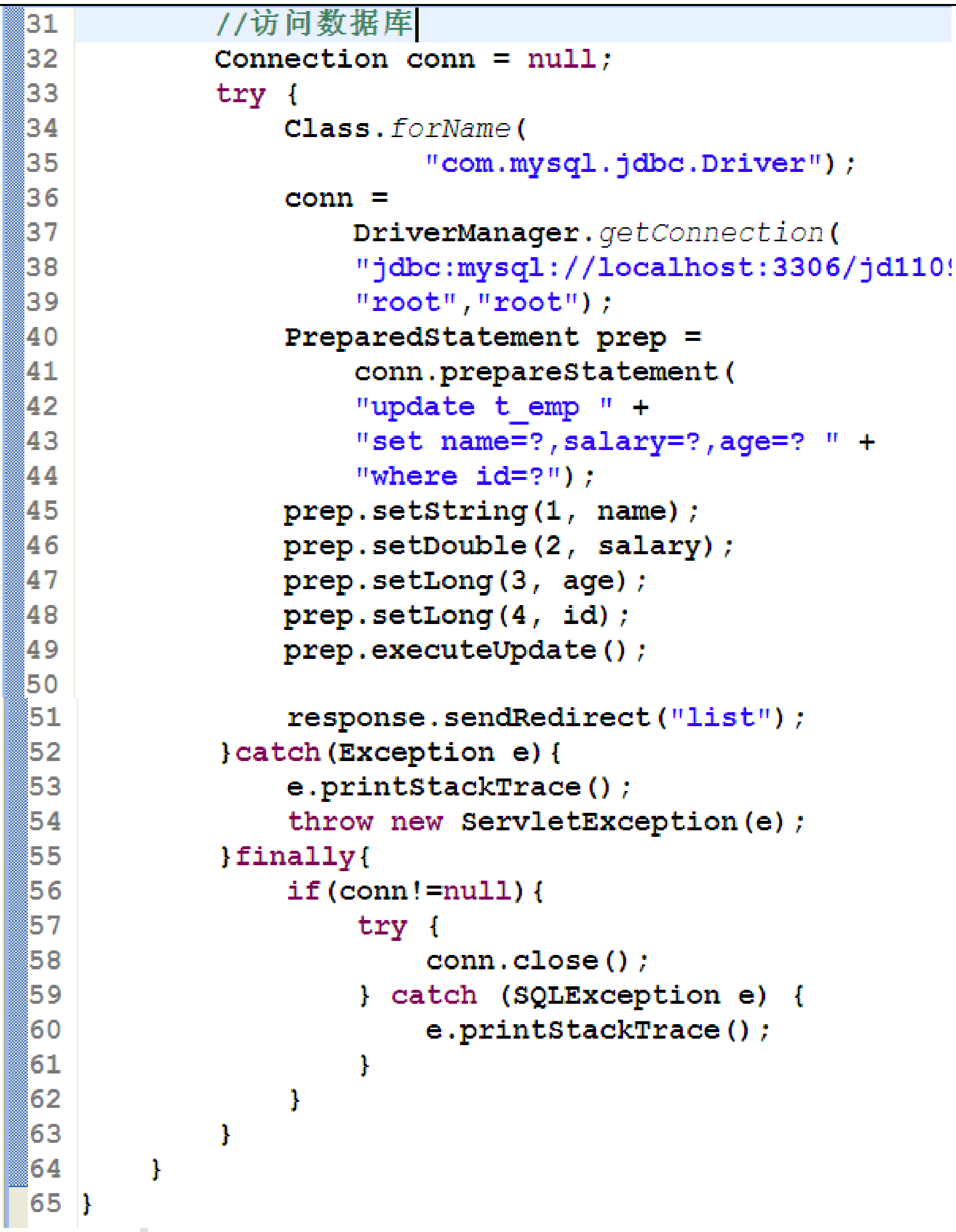
薪水: 2000.0

年龄: 21

11) 新建 ModifyEmpServlet.java



```
package web;

import java.io.IOException;□


public class ModifyEmpServlet extends HttpServlet{
    public void service(HttpServletRequest request
            HttpServletResponse response)
    throws ServletException,IOException{
        request.setCharacterEncoding("utf-8");
        long id =
            Long.parseLong(
            request.getParameter("id"));
        String name =
            request.getParameter("name");
        double salary =
            Double.parseDouble(
            request.getParameter("salary"));
        int age = Integer.parseInt(
            request.getParameter("age"));

```

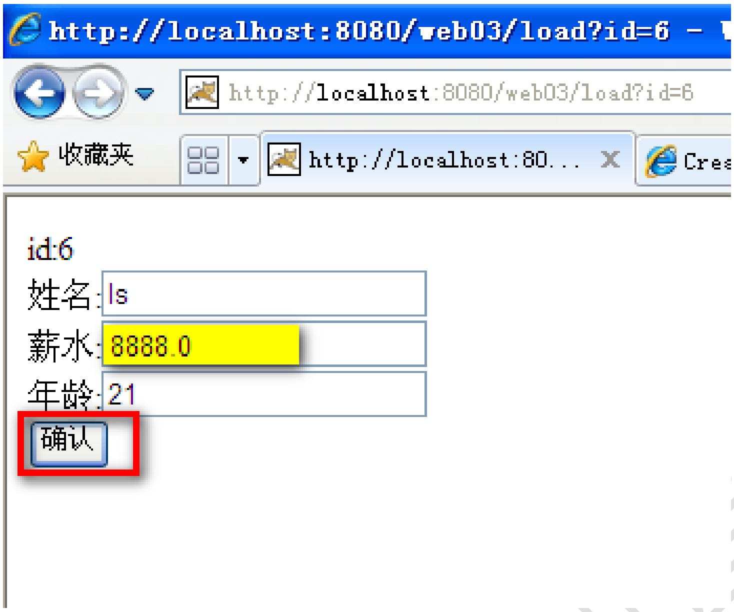


```
        //访问数据库
        Connection conn = null;
        try {
            Class.forName(
                    "com.mysql.jdbc.Driver");
            conn =
                DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/jd110
                "root","root");
            PreparedStatement prep =
                conn.prepareStatement(
                "update t_emp " +
                "set name=?,salary=?,age=? " +
                "where id=?");
            prep.setString(1, name);
            prep.setDouble(2, salary);
            prep.setLong(3, age);
            prep.setLong(4, id);
            prep.executeUpdate();

            response.sendRedirect("list");
        }catch(Exception e){
            e.printStackTrace();
            throw new ServletException(e);
        }finally{
            if(conn!=null){
                try {
                    conn.close();
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

12) 修改 web.xml（略）

13) 测试

将“ls”的薪水由 2000 修改为 8888



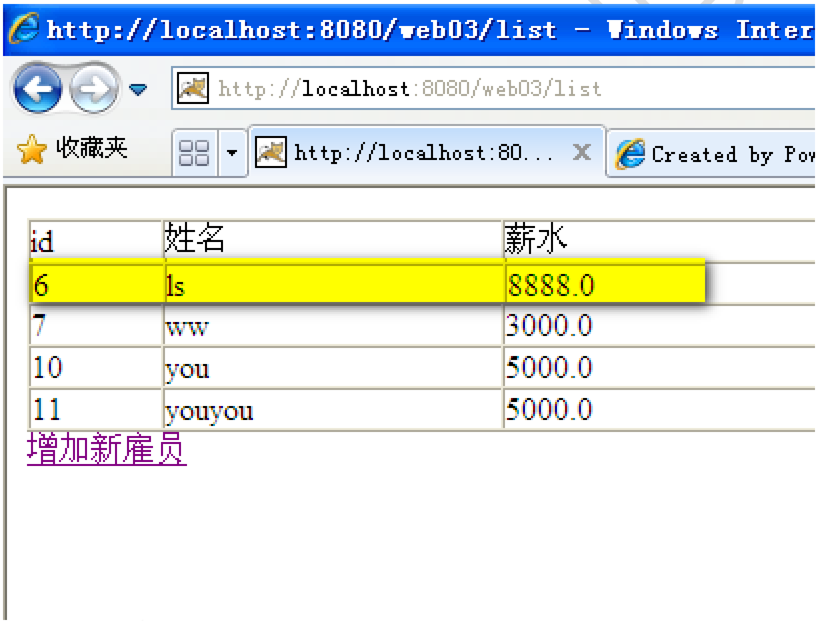
http://localhost:8080/web03/load?id=6

id:6

姓名: ls

薪水: 8888.0

年龄: 21



http://localhost:8080/web03/list

id	姓名	薪水
6	ls	8888.0
7	ww	3000.0
10	you	5000.0
11	youyou	5000.0

OK，演示结束。

【案例 2】雇员的增删改查

1) ListEmpServlet.java

```
package web;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ListEmpServlet extends HttpServlet{
  public void service(HttpServletRequest request,
                                   HttpServletResponse response)
     throws ServletException,IOException{
      //访问数据库
      Connection conn = null;
      try {
            Class.forName(
                        "com.mysql.jdbc.Driver");
            conn =
                  DriverManager.getConnection(
                  "jdbc:mysql://localhost:3306/jd1109db2",
                  "root","root");
            Statement stat =
                  conn.createStatement();
            ResultSet rst = stat.executeQuery(
                        "select * from t_emp");

            //使用查询得到的结果，生成一个表格
            response.setContentType(
                        "text/html;charset=utf-8");
            PrintWriter out =
    response.getWriter();
out.println(
        "<table border='1' " +
        "width='60%' " +
        "cellpadding='0' " +
        "cellspacing='0'>");
out.println("<tr>" +
        "<td>id</td>" +
        "<td>姓名</td>" +
        "<td>薪水</td>" +
        "<td>年龄</td>" +
        "<td>操作</td></tr>");
while(rst.next()){
    long id =
        rst.getLong("id");
    String name =
        rst.getString("name");
    double salary =
        rst.getDouble("salary");
    int age =
        rst.getInt("age");
    out.println("<tr><td>"
        + id + "</td><td> "
        + name + "</td><td>"
        + salary + "</td><td> "
        + age + "</td><td>" +
        "<a href='del?id=" + id + "'>" +
            "删除</a>" +
            "&nbsp;" +
            "<a href='load?id=" + id +"'>" +
            "修改</a></td></tr>");
}
out.println("</table>");
out.println(
        "<a href='addEmp.html'>" +
        "增加新雇员</a>");

out.close();
```

```
        } catch (Exception e) {
            e.printStackTrace();
            throw new ServletException(e);
        }finally{
            if(conn!=null){
                try {
                    conn.close();
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

2) DelEmpServlet.java

```
package web;

import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class DelEmpServlet
                    extends HttpServlet{

  public void service(HttpServletRequest request,
                    HttpServletResponse response)
    throws ServletException,IOException{
     long id =
        Long.parseLong(
                    request.getParameter("id"));
        //访问数据库
        Connection conn = null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager
                .getConnection(
                "jdbc:mysql://localhost:3306/jd1109db2",
                "root","root");
            PreparedStatement prep =
                conn.prepareStatement(
                "delete from t_emp where id = ?");
            prep.setLong(1, id);
            prep.executeUpdate();
            //重定向
            response.sendRedirect("list");

        } catch (Exception e) {
            //step1 先记录日志
            e.printStackTrace();
            //step2 抛出
            throw new ServletException(e);

        }finally{
            if(conn!=null){
                try {
                    conn.close();
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

3) LoadEmpServlet.java

```
package web;
```

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class LoadEmpServlet extends HttpServlet{
  public void service(HttpServletRequest request,
        HttpServletResponse response)
  throws ServletException,IOException{
    long id = Long.parseLong(
            request.getParameter("id"));
    //访问数据库
    Connection conn = null;
    try {
        Class.forName(
                "com.mysql.jdbc.Driver");
        conn =
            DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/jd1109db2",
            "root","root");
        PreparedStatement prep =
            conn.prepareStatement(
            "select * from t_emp where id = ?");
        prep.setLong(1, id);
        ResultSet rst = prep.executeQuery();
        response.setContentType(
                "text/html;charset=utf-8");
        PrintWriter out = response.getWriter();
        if(rst.next()){
            String name =
                rst.getString("name");
            double salary =
                rst.getDouble("salary");
            int age =
                rst.getInt("age");
            out.println(
                "<form action='modify?id=" +
                id + "' method='post'>");
            out.println(
                "id:" + id + "<br/>");
            out.println(
                "姓名:<input name='name' value='" +
                name + "'/><br/>");
            out.println(
                "薪水:<input name='salary' value='" +
                salary + "'/><br/>");
            out.println(
                "年龄:<input name='age' value='" +
                age + "'/><br/>");
            out.println(
                "<input type='submit' " +
                "value='确认'/>");
            out.println(
                "</form>");
            out.close();
        }
    }catch(Exception e){
        e.printStackTrace();
        throw new ServletException(e);
    }finally{
        if(conn!=null){
            try {
                conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
  }
}
```

4) ModifyEmpServlet.java

```
package web;

import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


public class ModifyEmpServlet extends HttpServlet{
    public void service(HttpServletRequest request,
            HttpServletResponse response)
    throws ServletException,IOException{
        request.setCharacterEncoding("utf-8");
        long id =
            Long.parseLong(
            request.getParameter("id"));
        String name =
            request.getParameter("name");
        double salary =
            Double.parseDouble(
            request.getParameter("salary"));
        int age = Integer.parseInt(
            request.getParameter("age"));

        //访问数据库
        Connection conn = null;
        try {
            Class.forName(
                    "com.mysql.jdbc.Driver");
            conn =
                DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/jd1109db2",
                "root","root");
            PreparedStatement prep =
                conn.prepareStatement(
                "update t_emp " +
                "set name=?,salary=?,age=? " +
                "where id=?");
            prep.setString(1, name);
            prep.setDouble(2, salary);
            prep.setLong(3, age);
            prep.setLong(4, id);
            prep.executeUpdate();

            response.sendRedirect("list");
        }catch(Exception e){
            e.printStackTrace();
            throw new ServletException(e);
        }finally{
            if(conn!=null){
                try {
                    conn.close();
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

5) AddEmpServlet.java

```
package web;
```

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class AddEmpServlet extends HttpServlet{

  public void service(HttpServletRequest request,
                      HttpServletResponse response)
    throws ServletException,IOException{

     //这行代码要放在 getParameter()执行之前。
     request.setCharacterEncoding("utf-8");

     String name = request.getParameter("name");
     double salary = Double.parseDouble(
                      request.getParameter("salary"));
     int age = Integer.parseInt(
              request.getParameter("age"));
     System.out.println("name:" + name);
     System.out.println("salary:" + salary);
     System.out.println("age:" + age);

     //访问数据库
     Connection conn = null;
     try {
          Class.forName("com.mysql.jdbc.Driver");
          conn = DriverManager
               .getConnection(
               "jdbc:mysql://localhost:3306/jd1109db2",
              "root","root");
          PreparedStatement prep =
              conn.prepareStatement(
              "insert into t_emp(name,salary,age) " +
              "values(?,?,?)");
          prep.setString(1, name);
          prep.setDouble(2, salary);
          prep.setInt(3, age);
          prep.executeUpdate();

          //1
          response.setContentType(
                  "text/html;charset=utf-8");
          PrintWriter out = response.getWriter();
          out.println("添加雇员成功");
          out.println(
                  "<a href='list'>" +
                  "查看雇员</a>");
          //out.close();

          //2. 重定向
          response.sendRedirect("list");

     } catch (Exception e) {
          //step1 先记录日志
          e.printStackTrace();
          //step2 抛出
          throw new ServletException(e);

     }finally{
          if(conn!=null){
              try {
                  conn.close();
              } catch (SQLException e) {
                  e.printStackTrace();
              }
          }
     }
```

```
    }
}
```

6) addEmp.html

```
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=utf-8">
<title>addEmp</title>
</head>
<body style="font-size:30px;">
    <form action="add" method="post">
        <fieldset>
            <legend>添加雇员</legend>
            姓名:<input name="name"/><br/>
            薪水:<input name="salary"/><br/>
            年龄:<input name="age"/><br/>
            <input type="submit" value="确认"/>
        </fieldset>
    </form>
</body>
</html>
```

7) error.html

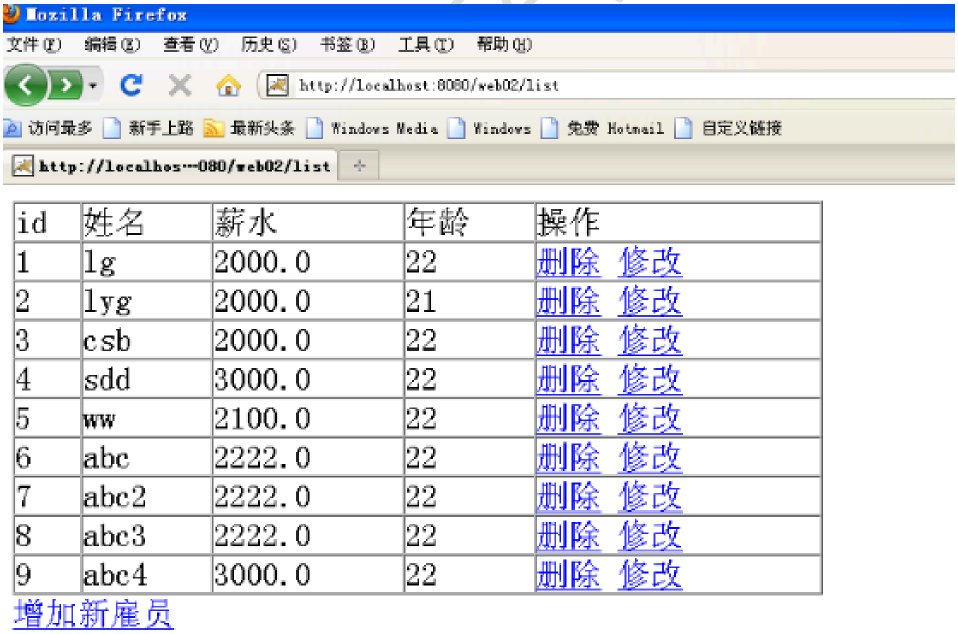
```
<html>
<head>
<meta http-equiv="Content-Type"
    content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body style="font-size:30px;color:red;">
    发生了系统错误，请稍后
    <a href="addEmp.html">重试</a>
</body>
</html>
```

8) web.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4"
    xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
    <servlet>
        <servlet-name>addEmpServlet</servlet-name>
        <servlet-class>web.AddEmpServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>listEmpServlet</servlet-name>
        <servlet-class>web.ListEmpServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>delEmpServlet</servlet-name>
        <servlet-class>web.DelEmpServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>loadEmpServlet</servlet-name>
        <servlet-class>web.LoadEmpServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>modifyEmpServlet</servlet-name>
        <servlet-class>web.ModifyEmpServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>addEmpServlet</servlet-name>
        <url-pattern>/add</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>listEmpServlet</servlet-name>
        <url-pattern>/list</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>delEmpServlet</servlet-name>
        <url-pattern>/del</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>loadEmpServlet</servlet-name>
        <url-pattern>/load</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>modifyEmpServlet</servlet-name>
        <url-pattern>/modify</url-pattern>
    </servlet-mapping>
    <error-page>
        <error-code>500</error-code>
        <location>/error.html</location>
    </error-page>
</web-app>
```

结果演示

访问 http://localhost:8080/web03/list


Mozilla Firefox
文件(F) 编辑(E) 查看(V) 历史(S) 书签(B) 工具(T) 帮助(H)
http://localhost:8080/web02/list
访问最多 新手上路 最新头条 Windows Media Windows 免费 Hotmail 自定义链接
http://localhos···080/web02/list

id	姓名	薪水	年龄	操作
1	lg	2000.0	22	删除 修改
2	lyg	2000.0	21	删除 修改
3	csb	2000.0	22	删除 修改
4	sdd	3000.0	22	删除 修改
5	ww	2100.0	22	删除 修改
6	abc	2222.0	22	删除 修改
7	abc2	2222.0	22	删除 修改
8	abc3	2222.0	22	删除 修改
9	abc4	3000.0	22	删除 修改

增加新雇员

4. DAO **

1) **什么是 DAO?**

data access object（DAO），封装了**数据访问逻辑**，调用者(一般是业务逻辑模块)不需要了解底层的数据访问细节就可以实现数据访问。这样一来，当底层的数据访问细节发生改变，不会影响到调用者。

2) **如何写一个 dao**

a. **实体类**

为了方便处理数据库中的记录，可以定义与记录对应的实体类，即可以将数据库中的某条记录转换成一个实体类的实例。

比如 Employee 类,提供 id,name,salary,age 属性及相关的 get/set 方法，我们可以将 t_emp 表中的记录转化成一个 Employee 对象。

b. **dao 接口**

声明一些数据访问方法，这些方法不要涉及具体的技术。

比如:

EmployeeDAO 接口，声明了：

public void save(Employee e) throws **SQLException**;

这种方式不好,因为出现了 jdbc 的异常类，这就要求实现该方法只能使用 jdbc。

换一种方式来声明，更好

public void save(Employee e) throws **Exception**;

public List<Employee> findAll() throws Exception;

c. **dao 实现类**

采用具体的技术实现 dao 接口。

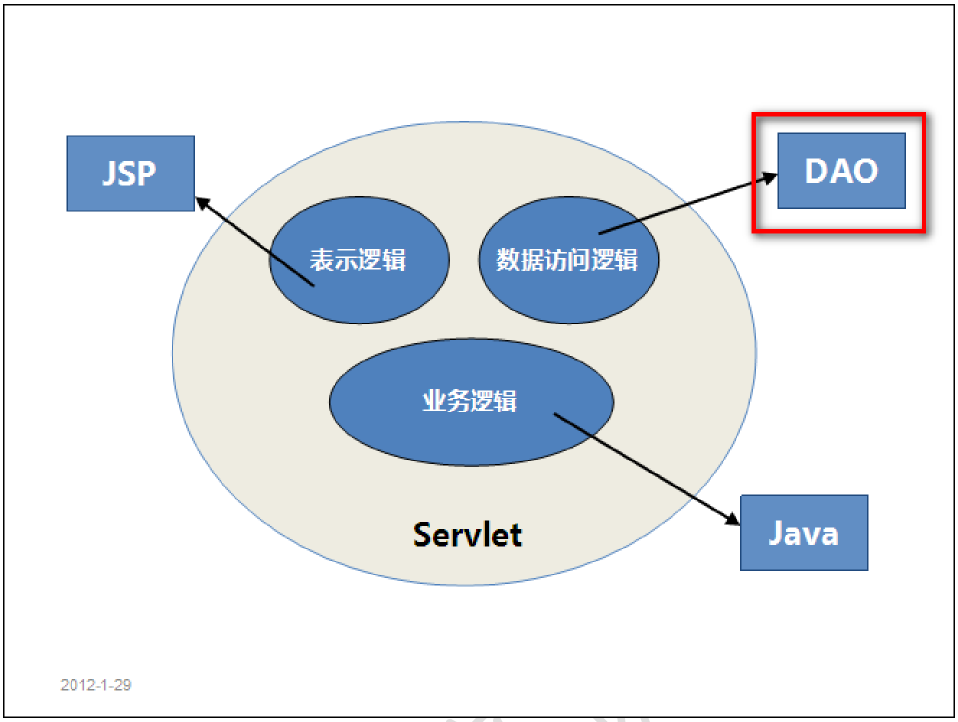
比如:

使用 jdbc 实现 EmployeeDAO 接口,写一个类 EmployeeDAOJdbcImpl。

d. **dao 工厂(工厂模式)**

工厂类:封装了对象的创建细节，为调用者提供符合要求的对象。

4.1. 什么是 DAO *



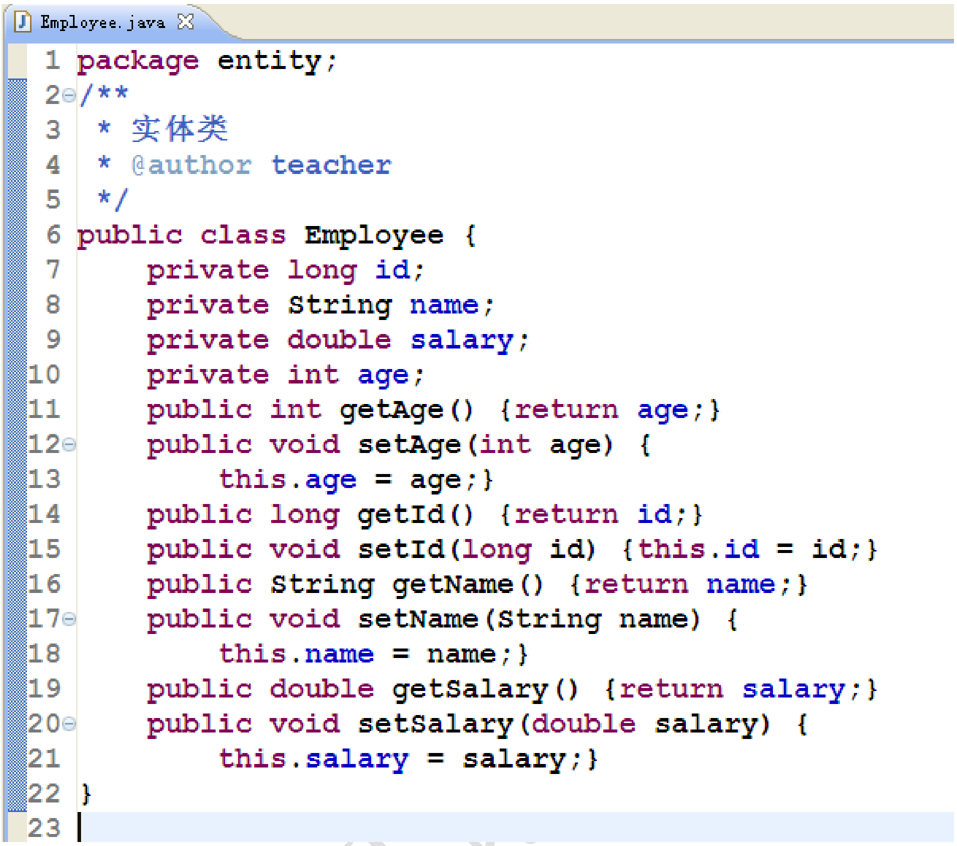
2012-1-29

- ✓ 数据访问逻辑：DAO（Data Access Object）
- ✓ 表示逻辑：和界面相关的，比如 out.print()

4.2.如何写一个 DAO **

步骤 1

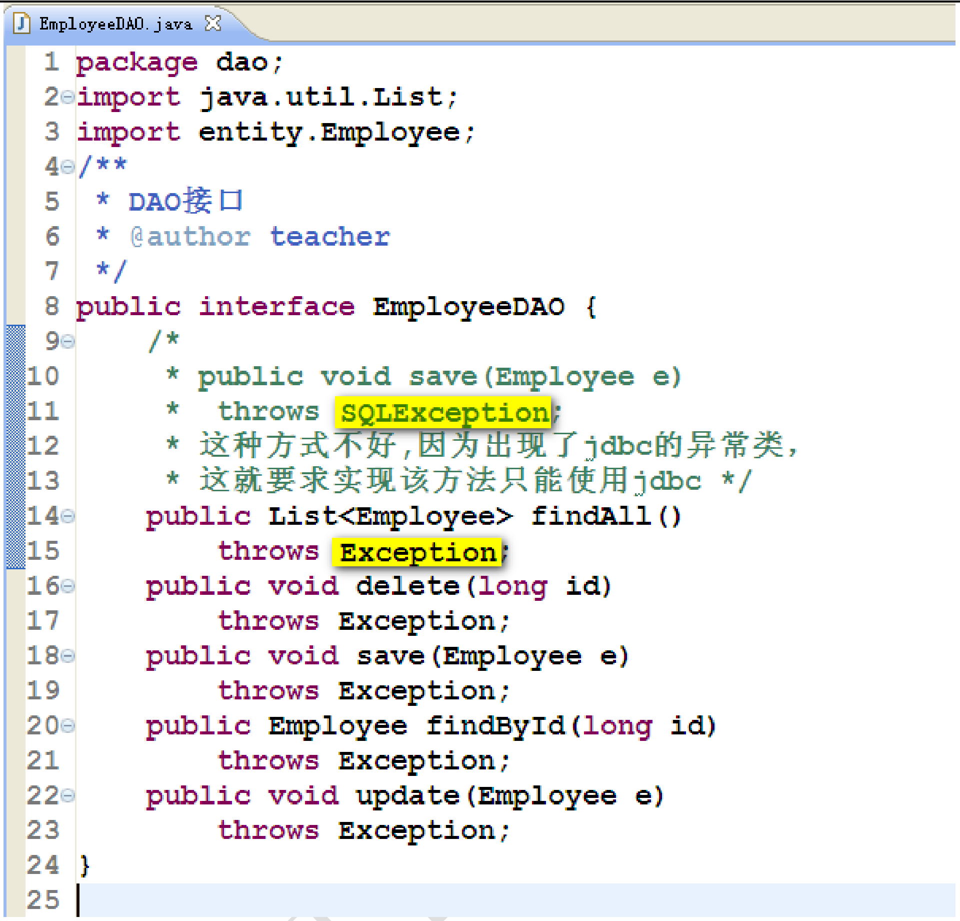
实体类 Employee.java



```java
package entity;
/**
 * 实体类
 * @author teacher
 */
public class Employee {
    private long id;
    private String name;
    private double salary;
    private int age;
    public int getAge() {return age;}
    public void setAge(int age) {
        this.age = age;}
    public long getId() {return id;}
    public void setId(long id) {this.id = id;}
    public String getName() {return name;}
    public void setName(String name) {
        this.name = name;}
    public double getSalary() {return salary;}
    public void setSalary(double salary) {
        this.salary = salary;}
}
```

步骤 2

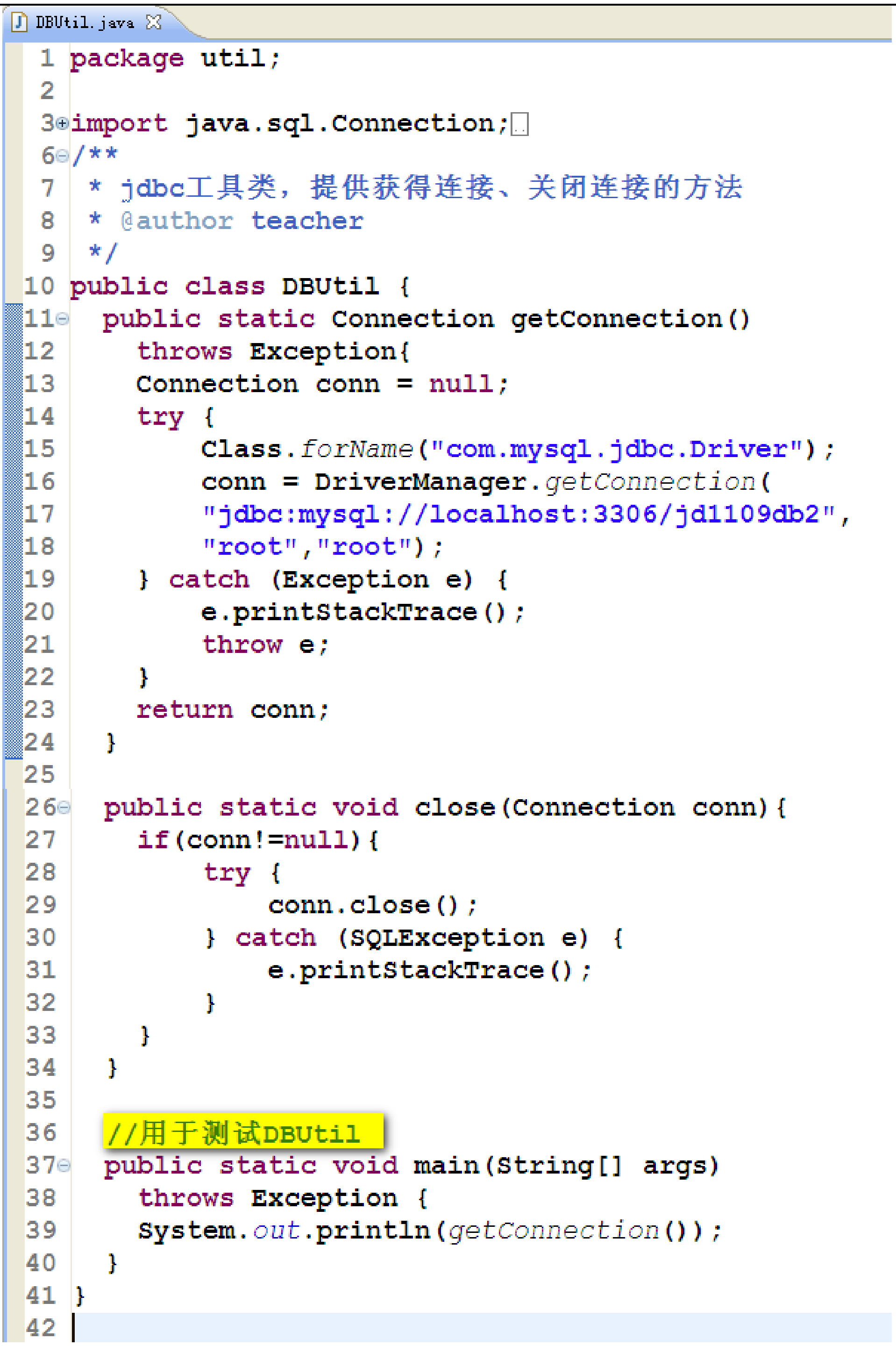
dao 接口 EmployeeDAO.java



```java
package dao;
import java.util.List;
import entity.Employee;
/**
 * DAO接口
 * @author teacher
 */
public interface EmployeeDAO {
    /*
     * public void save(Employee e)
     *   throws SQLException;
     * 这种方式不好,因为出现了jdbc的异常类,
     * 这就要求实现该方法只能使用jdbc */
    public List<Employee> findAll()
        throws Exception;
    public void delete(long id)
        throws Exception;
    public void save(Employee e)
        throws Exception;
    public Employee findById(long id)
        throws Exception;
    public void update(Employee e)
        throws Exception;
}

```

步骤 3

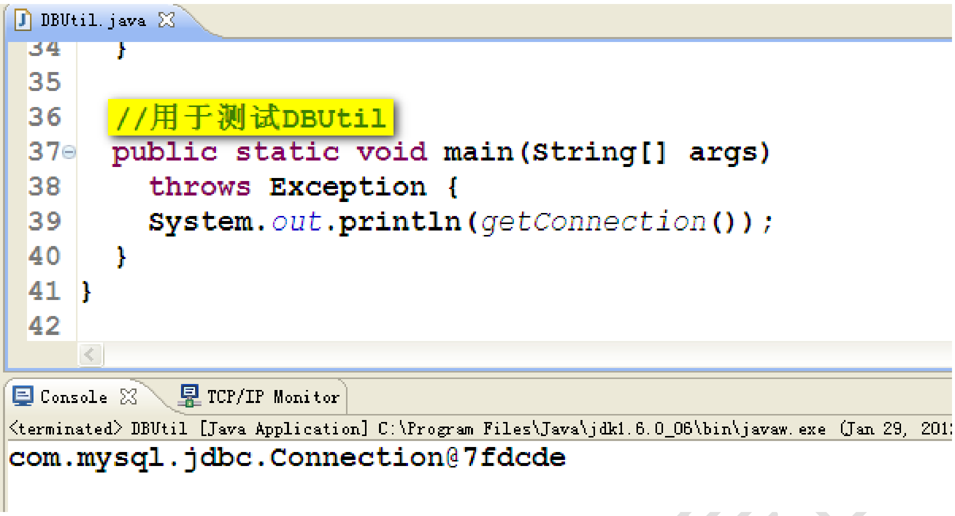
工具类 DBUtil.java



```java
package util;

import java.sql.Connection;
/**
 * jdbc工具类，提供获得连接、关闭连接的方法
 * @author teacher
 */
public class DBUtil {
  public static Connection getConnection()
    throws Exception{
    Connection conn = null;
    try {
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(
        "jdbc:mysql://localhost:3306/jd1109db2",
        "root","root");
    } catch (Exception e) {
        e.printStackTrace();
        throw e;
    }
    return conn;
  }

  public static void close(Connection conn){
    if(conn!=null){
        try {
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
  }

  //用于测试DBUtil
  public static void main(String[] args)
    throws Exception {
    System.out.println(getConnection());
  }
}

```

步骤 4

测试 DBUtil

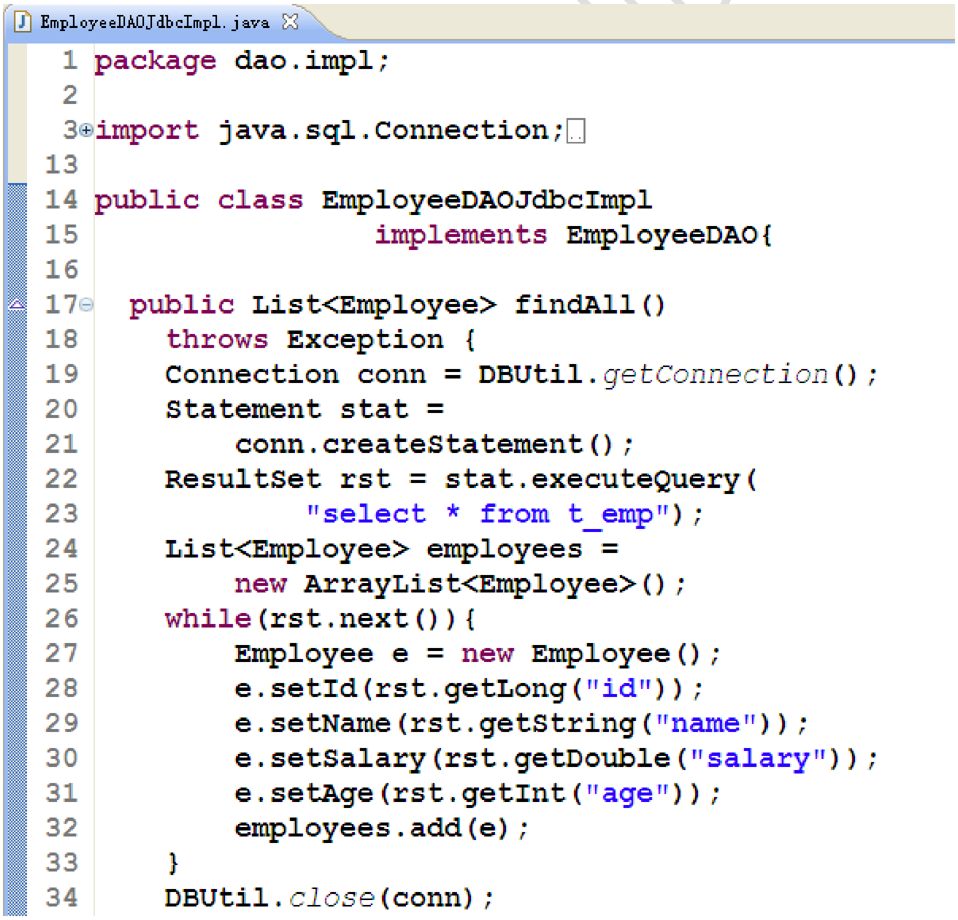


```java
    }

    //用于测试DBUtil
    public static void main(String[] args)
      throws Exception {
      System.out.println(getConnection());
    }
}

```

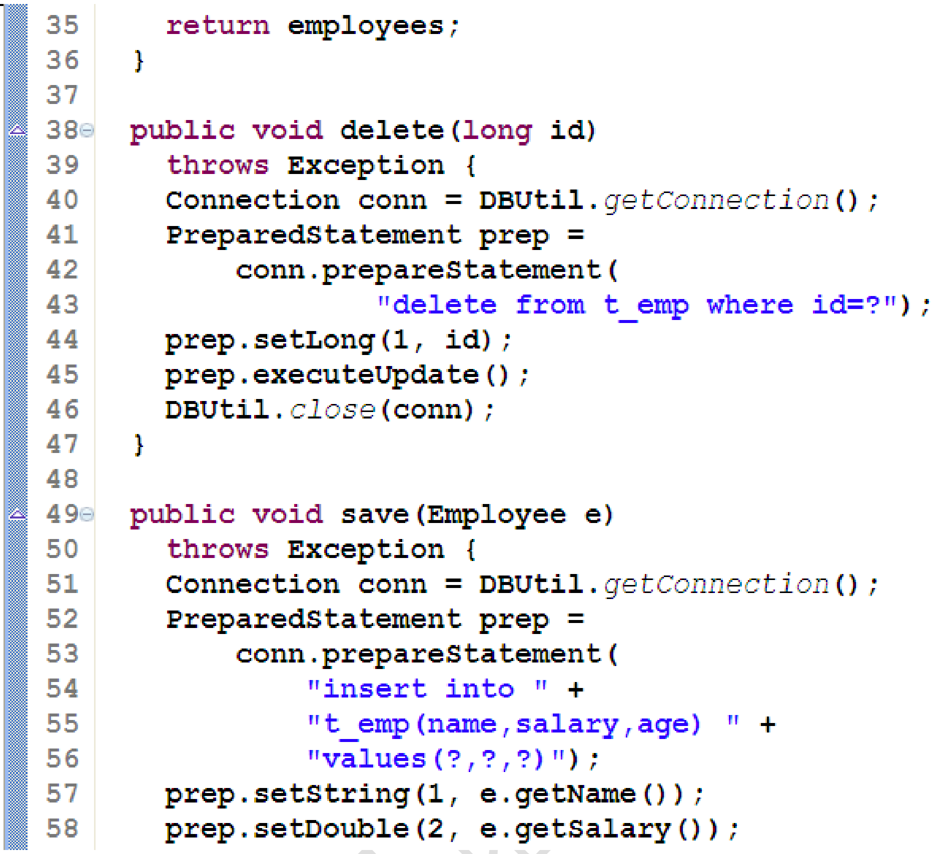
```
<terminated> DBUtil [Java Application] C:\Program Files\Java\jdk1.6.0_06\bin\javaw.exe (Jan 29, 201
com.mysql.jdbc.Connection@7fdcde
```

步骤 5

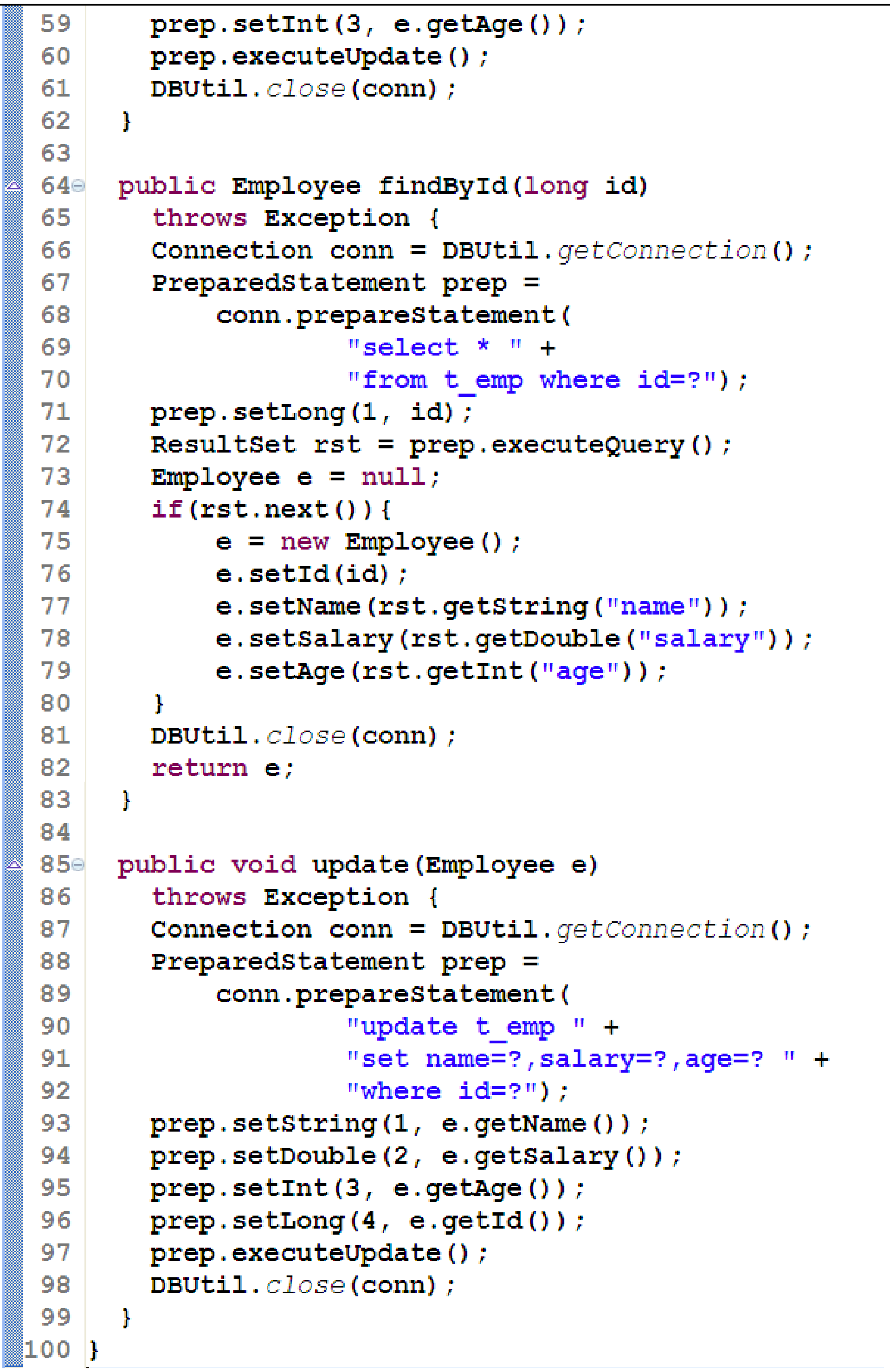
dao 实现类



```java
package dao.impl;

import java.sql.Connection;

public class EmployeeDAOJdbcImpl
                implements EmployeeDAO{

  public List<Employee> findAll()
    throws Exception {
    Connection conn = DBUtil.getConnection();
    Statement stat =
        conn.createStatement();
    ResultSet rst = stat.executeQuery(
            "select * from t_emp");
    List<Employee> employees =
        new ArrayList<Employee>();
    while(rst.next()){
        Employee e = new Employee();
        e.setId(rst.getLong("id"));
        e.setName(rst.getString("name"));
        e.setSalary(rst.getDouble("salary"));
        e.setAge(rst.getInt("age"));
        employees.add(e);
    }
    DBUtil.close(conn);
```



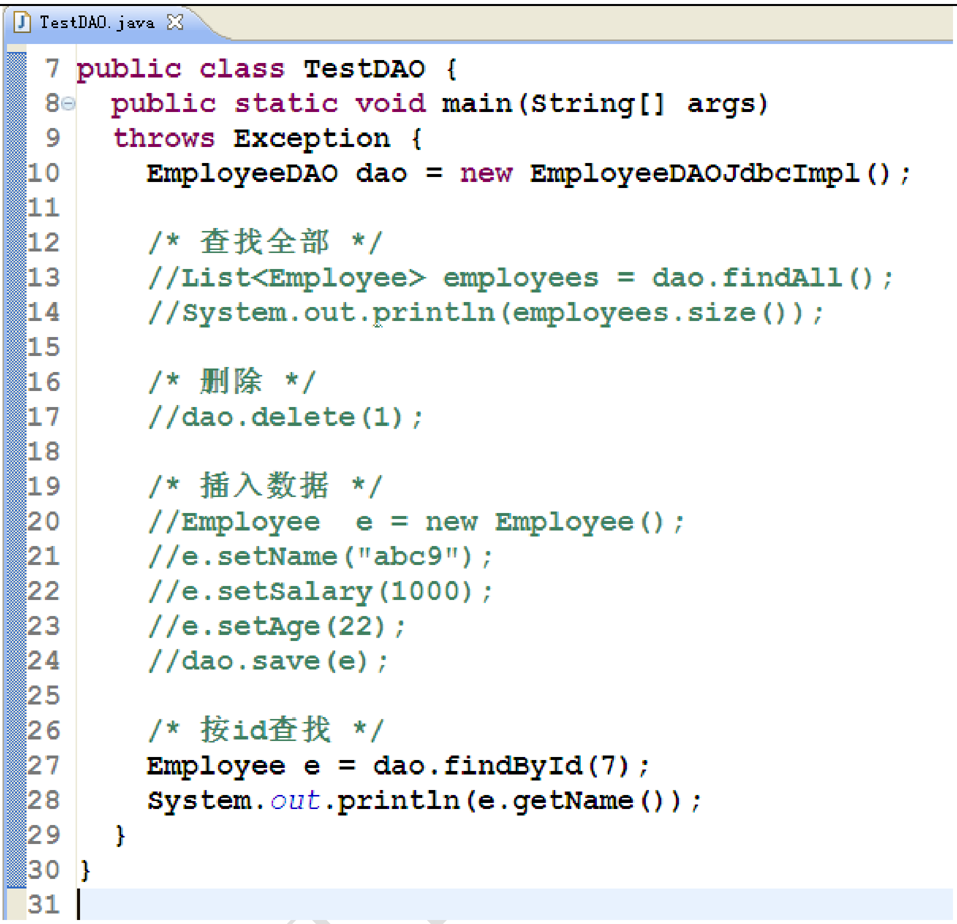
```
    return employees;
  }

  public void delete(long id)
    throws Exception {
    Connection conn = DBUtil.getConnection();
    PreparedStatement prep =
        conn.prepareStatement(
                "delete from t_emp where id=?");
    prep.setLong(1, id);
    prep.executeUpdate();
    DBUtil.close(conn);
  }

  public void save(Employee e)
    throws Exception {
    Connection conn = DBUtil.getConnection();
    PreparedStatement prep =
        conn.prepareStatement(
            "insert into " +
            "t_emp(name,salary,age) " +
            "values(?,?,?)");
    prep.setString(1, e.getName());
    prep.setDouble(2, e.getSalary());
```



```
    prep.setInt(3, e.getAge());
    prep.executeUpdate();
    DBUtil.close(conn);
  }

  public Employee findById(long id)
    throws Exception {
    Connection conn = DBUtil.getConnection();
    PreparedStatement prep =
        conn.prepareStatement(
                "select * " +
                "from t_emp where id=?");
    prep.setLong(1, id);
    ResultSet rst = prep.executeQuery();
    Employee e = null;
    if(rst.next()){
        e = new Employee();
        e.setId(id);
        e.setName(rst.getString("name"));
        e.setSalary(rst.getDouble("salary"));
        e.setAge(rst.getInt("age"));
    }
    DBUtil.close(conn);
    return e;
  }

  public void update(Employee e)
    throws Exception {
    Connection conn = DBUtil.getConnection();
    PreparedStatement prep =
        conn.prepareStatement(
                "update t_emp " +
                "set name=?,salary=?,age=? " +
                "where id=?");
    prep.setString(1, e.getName());
    prep.setDouble(2, e.getSalary());
    prep.setInt(3, e.getAge());
    prep.setLong(4, e.getId());
    prep.executeUpdate();
    DBUtil.close(conn);
  }
}
```

步骤 6

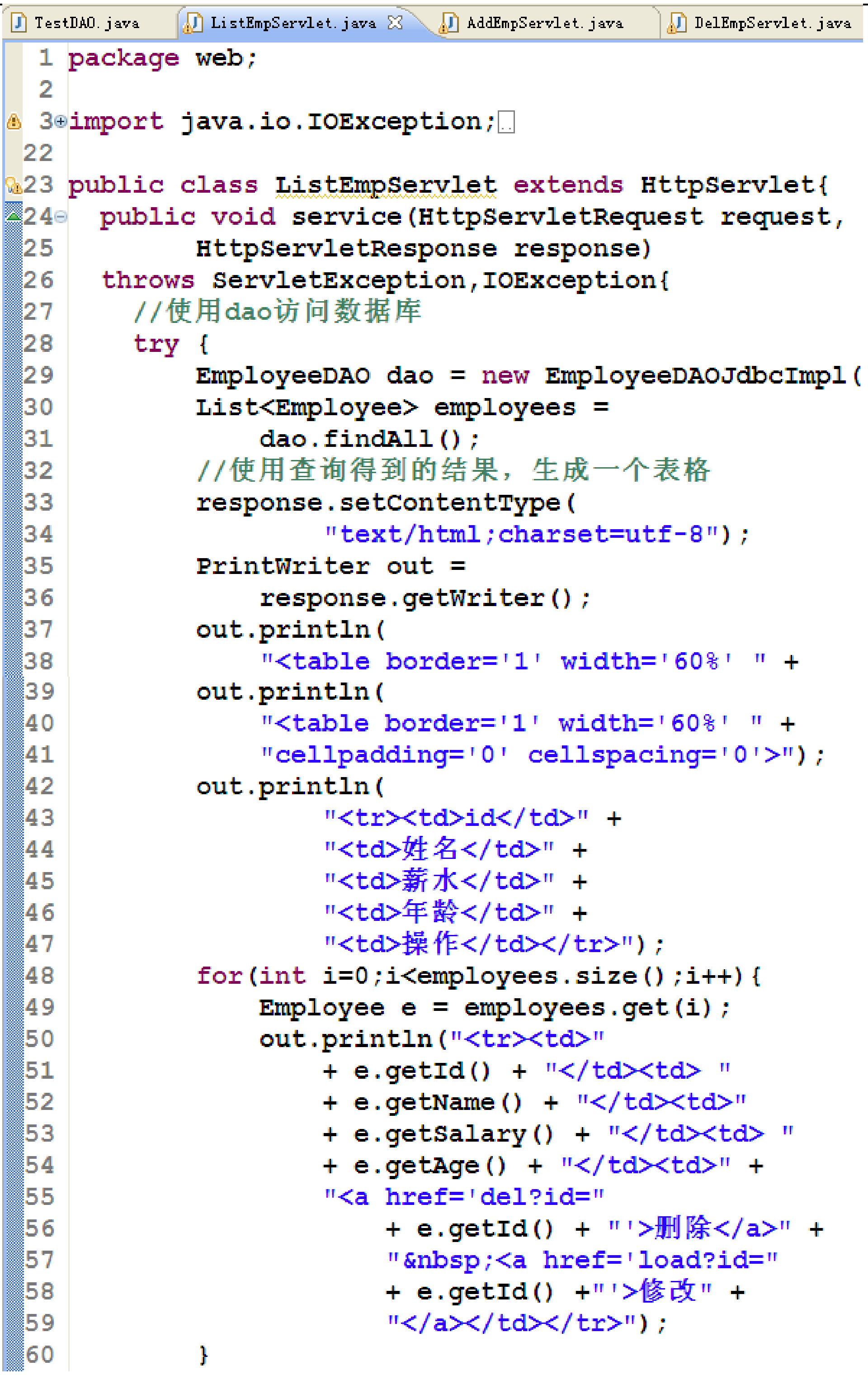
测试 dao 实现类



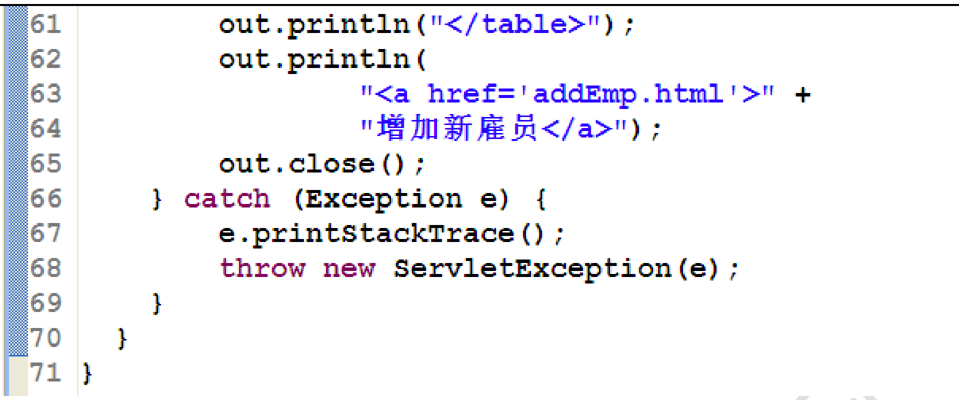
```
public class TestDAO {
    public static void main(String[] args)
    throws Exception {
        EmployeeDAO dao = new EmployeeDAOJdbcImpl();

        /* 查找全部 */
        //List<Employee> employees = dao.findAll();
        //System.out.println(employees.size());

        /* 删除 */
        //dao.delete(1);

        /* 插入数据 */
        //Employee  e = new Employee();
        //e.setName("abc9");
        //e.setSalary(1000);
        //e.setAge(22);
        //dao.save(e);

        /* 按id查找 */
        Employee e = dao.findById(7);
        System.out.println(e.getName());
    }
}
```

步骤 7.1

修改 ListEmpServlet.java



```
package web;

import java.io.IOException;

public class ListEmpServlet extends HttpServlet{
  public void service(HttpServletRequest request,
        HttpServletResponse response)
  throws ServletException,IOException{
    //使用dao访问数据库
    try {
        EmployeeDAO dao = new EmployeeDAOJdbcImpl(
        List<Employee> employees =
            dao.findAll();
        //使用查询得到的结果，生成一个表格
        response.setContentType(
                "text/html;charset=utf-8");
        PrintWriter out =
            response.getWriter();
        out.println(
            "<table border='1' width='60%' " +
        out.println(
            "<table border='1' width='60%' " +
            "cellpadding='0' cellspacing='0'>");
        out.println(
                "<tr><td>id</td>" +
                "<td>姓名</td>" +
                "<td>薪水</td>" +
                "<td>年龄</td>" +
                "<td>操作</td></tr>");
        for(int i=0;i<employees.size();i++){
            Employee e = employees.get(i);
            out.println("<tr><td>"
                + e.getId() + "</td><td> "
                + e.getName() + "</td><td>"
                + e.getSalary() + "</td><td> "
                + e.getAge() + "</td><td>" +
                "<a href='del?id="
                    + e.getId() + "'>删除</a>" +
                    "&nbsp;<a href='load?id="
                    + e.getId() +"'>修改" +
                    "</a></td></tr>");
        }
```

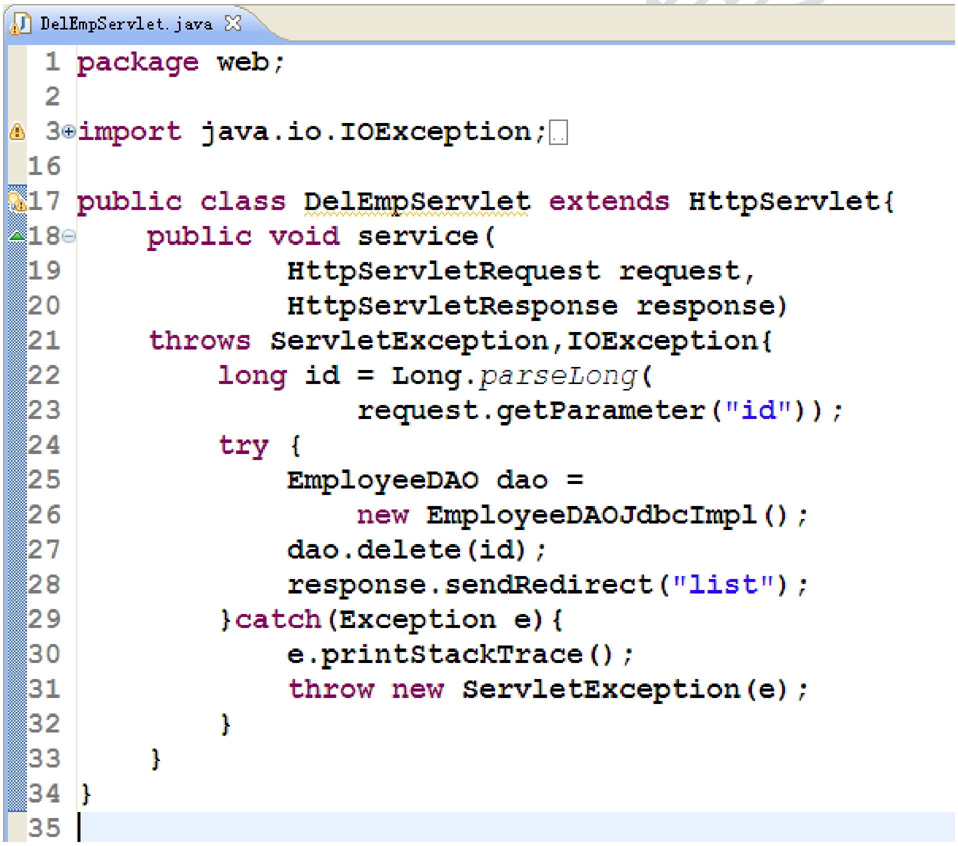


```
        out.println("</table>");
        out.println(
                "<a href='addEmp.html'>" +
                "增加新雇员</a>");
        out.close();
    } catch (Exception e) {
        e.printStackTrace();
        throw new ServletException(e);
    }
  }
}
```

测试（略）

步骤 7.2

修改 DelEmpServlet.java

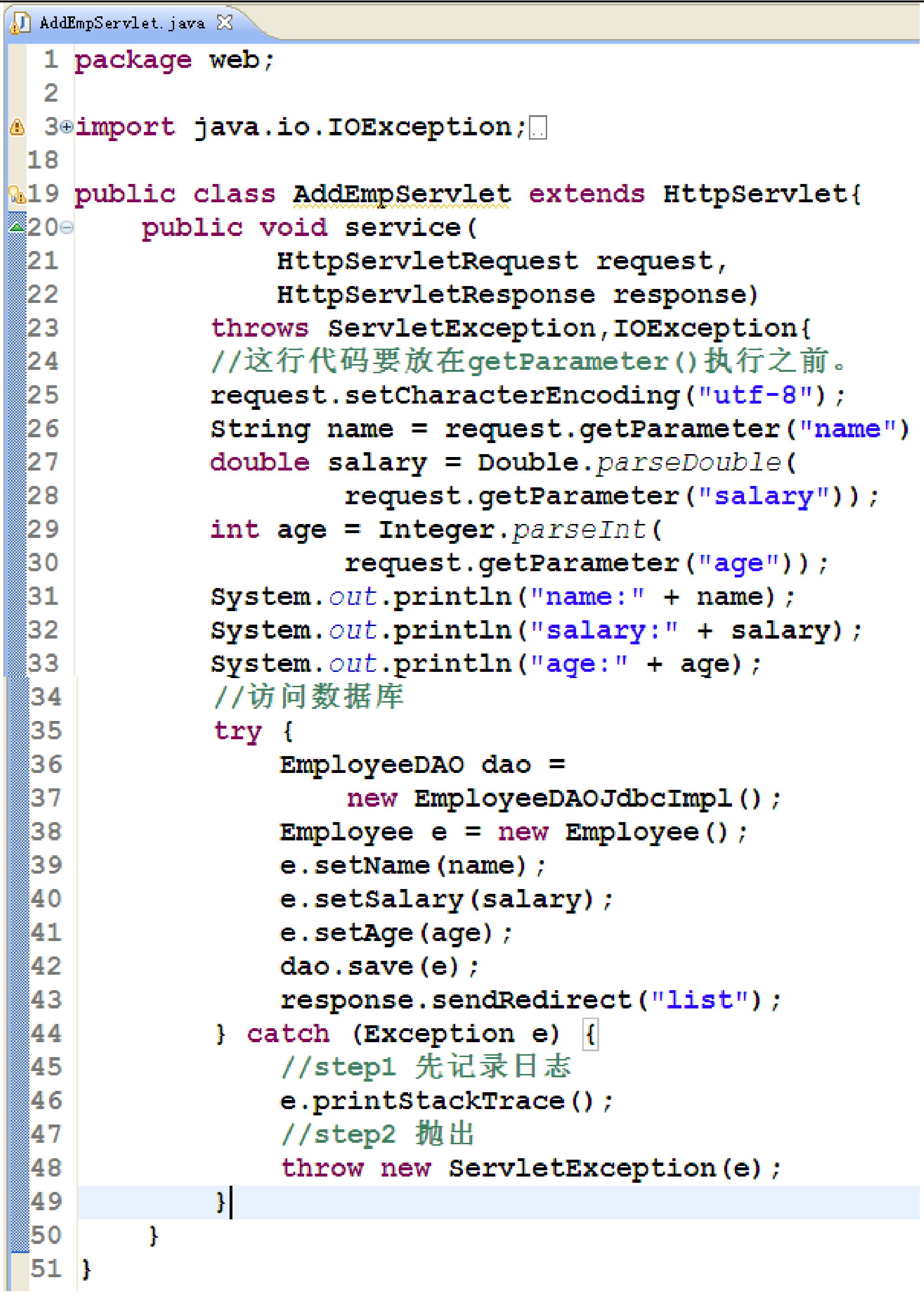


```
package web;

import java.io.IOException;□

public class DelEmpServlet extends HttpServlet{
    public void service(
            HttpServletRequest request,
            HttpServletResponse response)
    throws ServletException,IOException{
        long id = Long.parseLong(
                request.getParameter("id"));
        try {
            EmployeeDAO dao =
                new EmployeeDAOJdbcImpl();
            dao.delete(id);
            response.sendRedirect("list");
        }catch(Exception e){
            e.printStackTrace();
            throw new ServletException(e);
        }
    }
}

```

测试

步骤 7.3

修改 AddEmpServlet.java

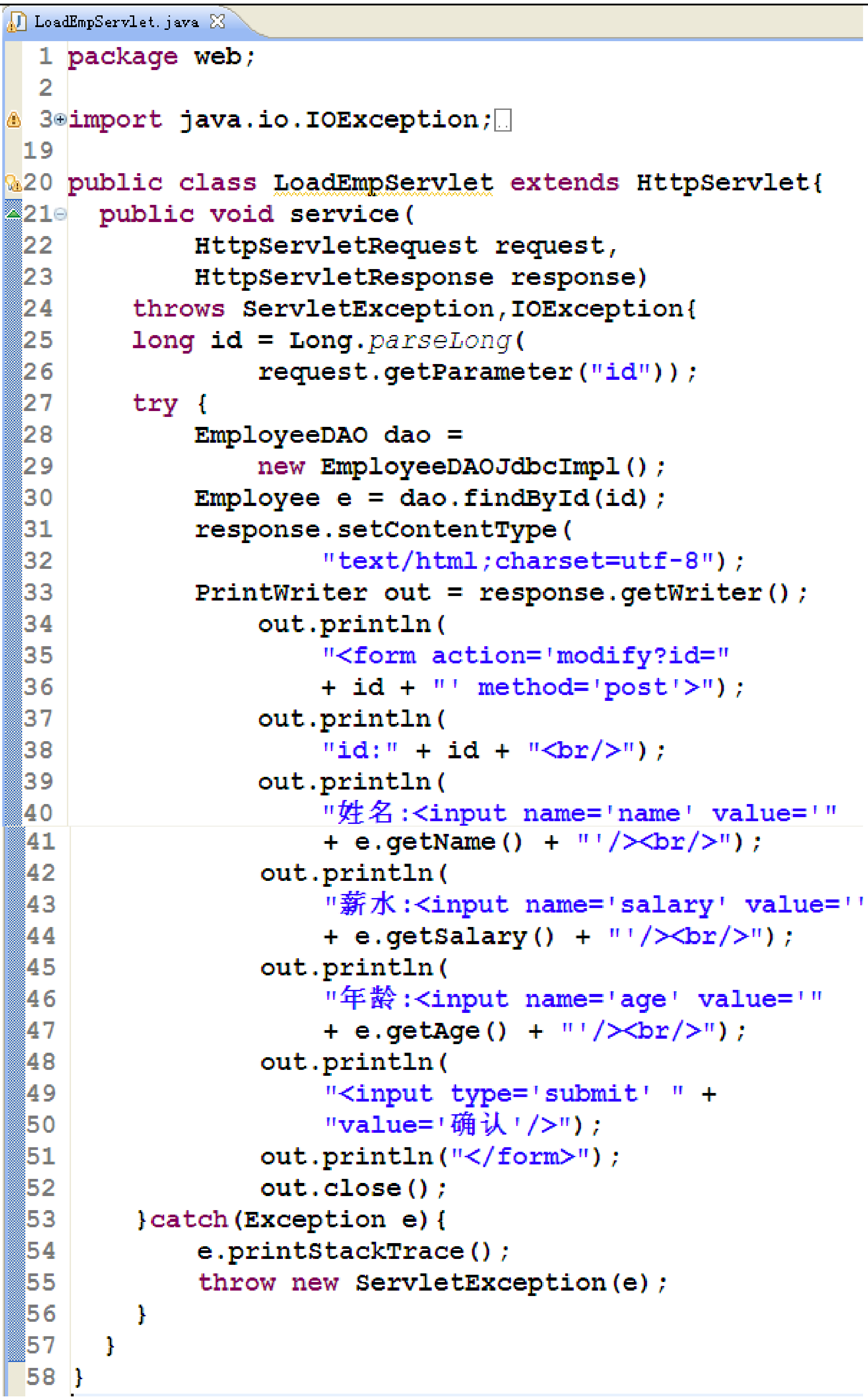


```java
package web;

import java.io.IOException;

public class AddEmpServlet extends HttpServlet{
    public void service(
            HttpServletRequest request,
            HttpServletResponse response)
        throws ServletException,IOException{
        //这行代码要放在getParameter()执行之前。
        request.setCharacterEncoding("utf-8");
        String name = request.getParameter("name")
        double salary = Double.parseDouble(
                request.getParameter("salary"));
        int age = Integer.parseInt(
                request.getParameter("age"));
        System.out.println("name:" + name);
        System.out.println("salary:" + salary);
        System.out.println("age:" + age);
        //访问数据库
        try {
            EmployeeDAO dao =
                new EmployeeDAOJdbcImpl();
            Employee e = new Employee();
            e.setName(name);
            e.setSalary(salary);
            e.setAge(age);
            dao.save(e);
            response.sendRedirect("list");
        } catch (Exception e) {
            //step1 先记录日志
            e.printStackTrace();
            //step2 抛出
            throw new ServletException(e);
        }
    }
}
```

测试

步骤 7.4

修改 LoadEmpServlet.java

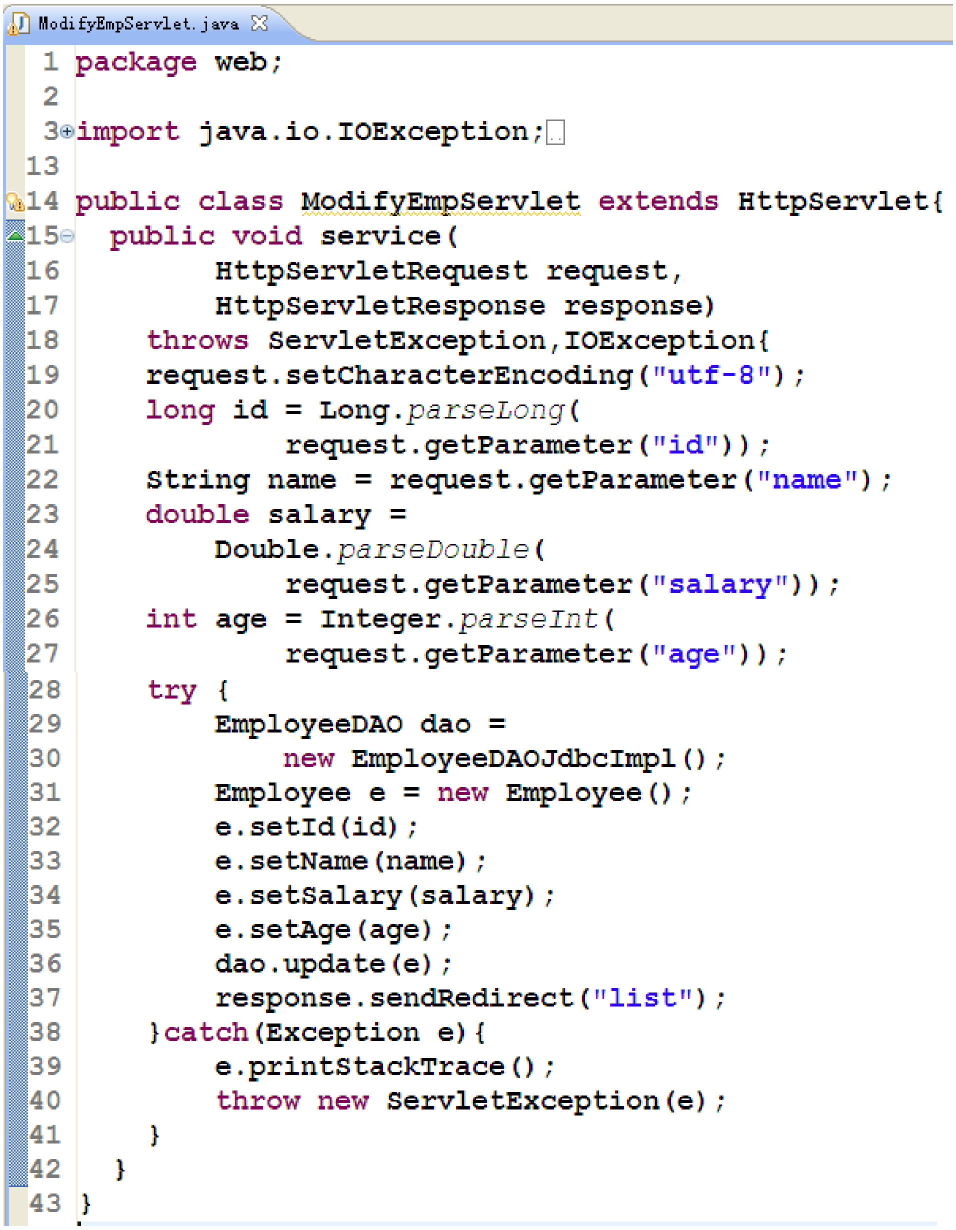


```java
package web;

import java.io.IOException;

public class LoadEmpServlet extends HttpServlet{
  public void service(
        HttpServletRequest request,
        HttpServletResponse response)
    throws ServletException,IOException{
    long id = Long.parseLong(
            request.getParameter("id"));
    try {
        EmployeeDAO dao =
            new EmployeeDAOJdbcImpl();
        Employee e = dao.findById(id);
        response.setContentType(
                "text/html;charset=utf-8");
        PrintWriter out = response.getWriter();
            out.println(
                "<form action='modify?id="
                + id + "' method='post'>");
            out.println(
                "id:" + id + "<br/>");
            out.println(
                "姓名:<input name='name' value='"
                + e.getName() + "'/><br/>");
            out.println(
                "薪水:<input name='salary' value='"
                + e.getSalary() + "'/><br/>");
            out.println(
                "年龄:<input name='age' value='"
                + e.getAge() + "'/><br/>");
            out.println(
                "<input type='submit' " +
                "value='确认'/>");
            out.println("</form>");
            out.close();
    }catch(Exception e){
        e.printStackTrace();
        throw new ServletException(e);
    }
  }
}
```

测试

步骤 7.5

修改 ModifyEmpServlet.java



```java
package web;

import java.io.IOException;

public class ModifyEmpServlet extends HttpServlet{
  public void service(
        HttpServletRequest request,
        HttpServletResponse response)
    throws ServletException,IOException{
    request.setCharacterEncoding("utf-8");
    long id = Long.parseLong(
            request.getParameter("id"));
    String name = request.getParameter("name");
    double salary =
        Double.parseDouble(
            request.getParameter("salary"));
    int age = Integer.parseInt(
            request.getParameter("age"));
    try {
        EmployeeDAO dao =
            new EmployeeDAOJdbcImpl();
        Employee e = new Employee();
        e.setId(id);
        e.setName(name);
        e.setSalary(salary);
        e.setAge(age);
        dao.update(e);
        response.sendRedirect("list");
    }catch(Exception e){
        e.printStackTrace();
        throw new ServletException(e);
    }
  }
}
```

测试

完成。

4.3. dao 工厂(工厂模式) **

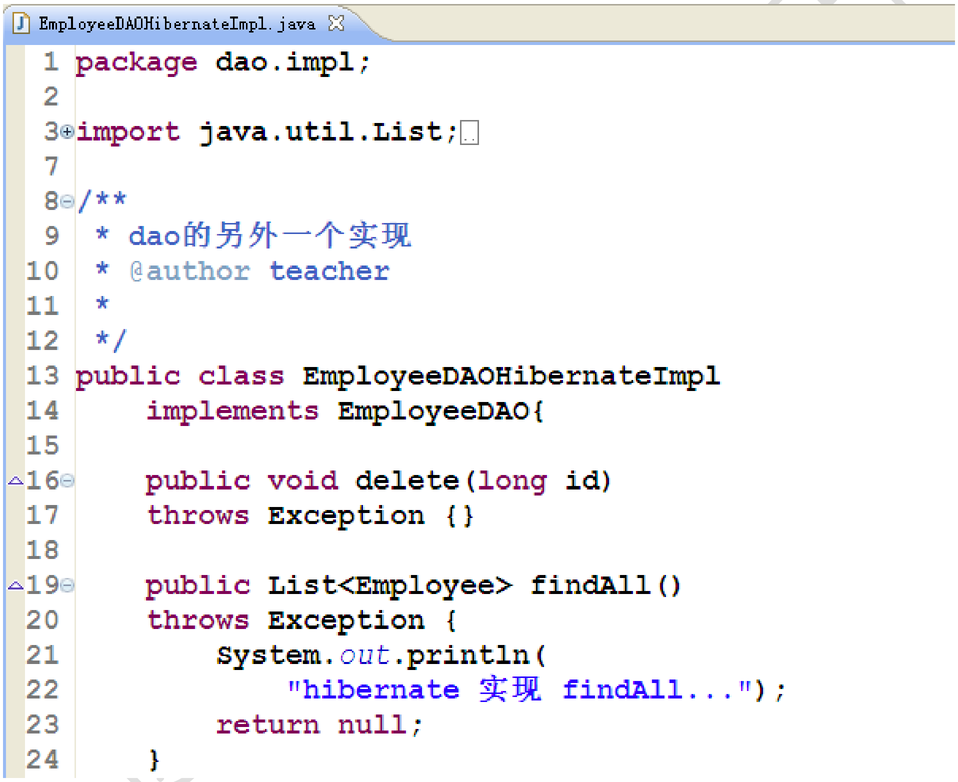
工厂类:封装了对象的创建细节，为调用者提供符合要求的对象。

现在做这样一个假设，假如我们的底层数据访问不再使用 JDBC，而采用 Hibernate（框架技术，后续会学习，此处做了解）。

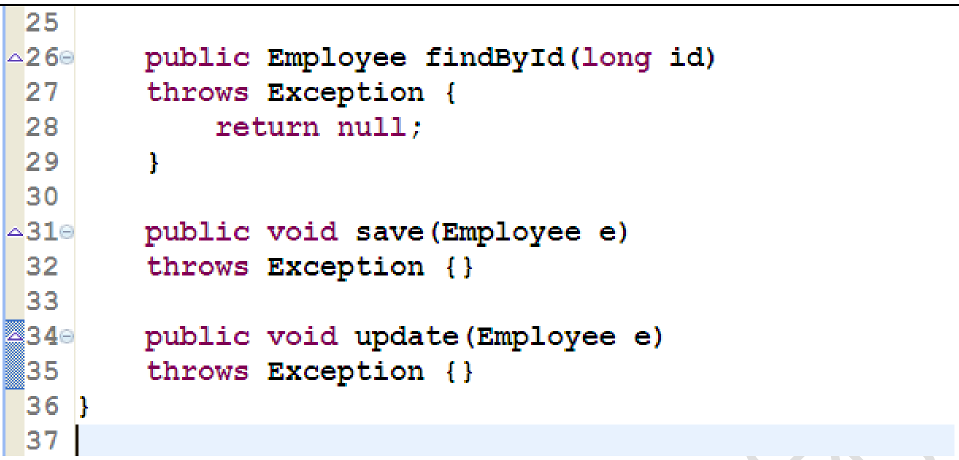
那么我们需要为 **EmployeeDAO** 增加一个实现 **EmployeeDAOHibernateImpl.java**

并且修改之前的代码，DelEmpServlet、AddEmpServlet、ListEmpServlet、LoadEmpServlet、ModifyEmpServlet 都要做相应修改，如下所示：

1) EmployeeDAOHibernateImpl.java（模拟实现）

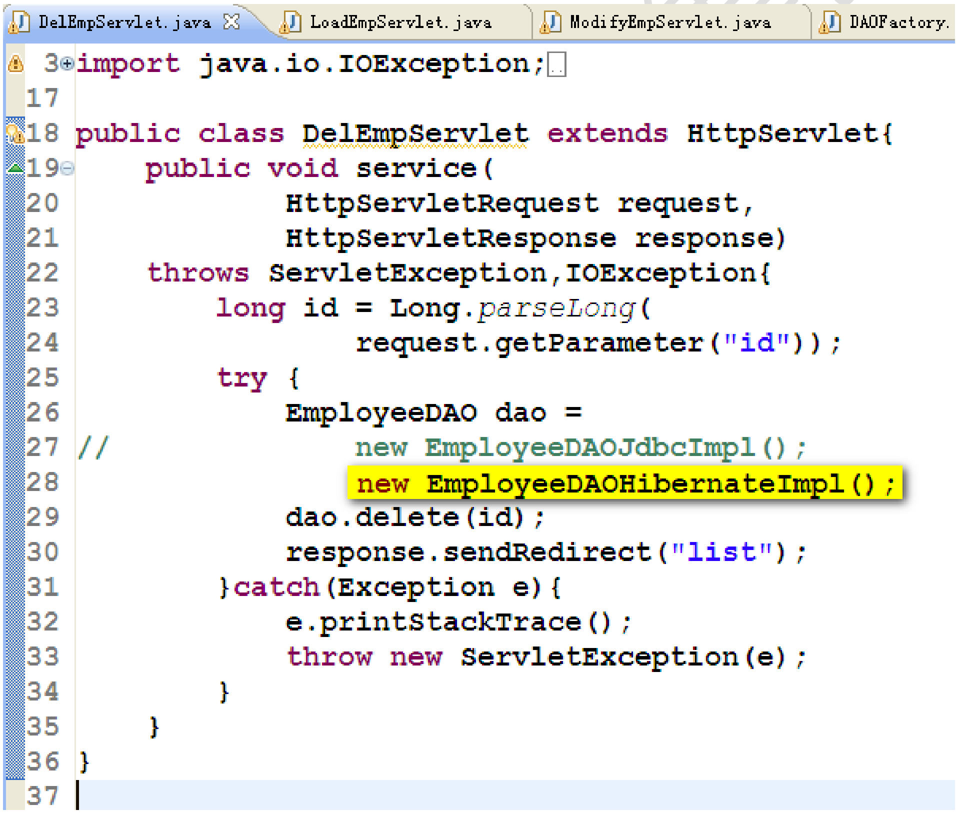


```
package dao.impl;

import java.util.List;

/**
 * dao的另外一个实现
 * @author teacher
 *
 */
public class EmployeeDAOHibernateImpl
    implements EmployeeDAO{

    public void delete(long id)
    throws Exception {}

    public List<Employee> findAll()
    throws Exception {
        System.out.println(
            "hibernate 实现 findAll...");
        return null;
    }
```



```

    public Employee findById(long id)
    throws Exception {
        return null;
    }

    public void save(Employee e)
    throws Exception {}

    public void update(Employee e)
    throws Exception {}
}

```

2)　**DelEmpServlet.java**



```
import java.io.IOException;□

public class DelEmpServlet extends HttpServlet{
    public void service(
            HttpServletRequest request,
            HttpServletResponse response)
    throws ServletException,IOException{
        long id = Long.parseLong(
                request.getParameter("id"));
        try {
            EmployeeDAO dao =
//              new EmployeeDAOJdbcImpl();
                new EmployeeDAOHibernateImpl();
            dao.delete(id);
            response.sendRedirect("list");
        }catch(Exception e){
            e.printStackTrace();
            throw new ServletException(e);
        }
    }
}

```

3)　**AddEmpServlet.java（略）**

4)　**ListEmpServlet（略）**

5)　**LoadEmpServlet（略）**

6)　**ModifyEmpServlet（略）**

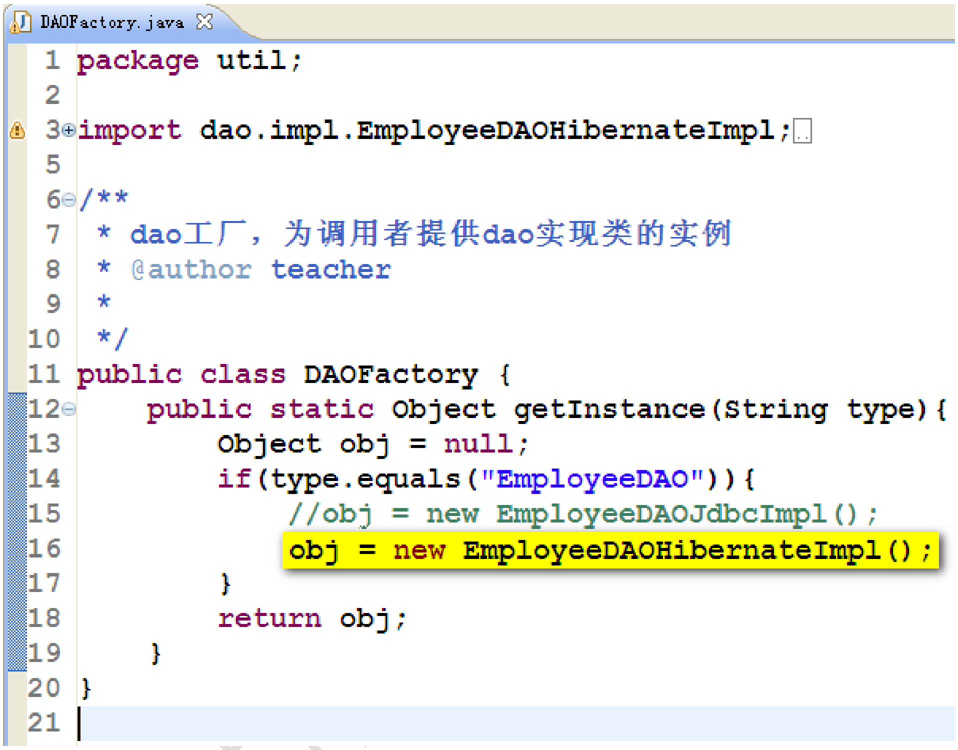
然而，我们希望**尽量不修改写好的代码（Servlet 代码）**

此时，我们可以使用**工厂模式**

演示：工厂模式

步骤 1

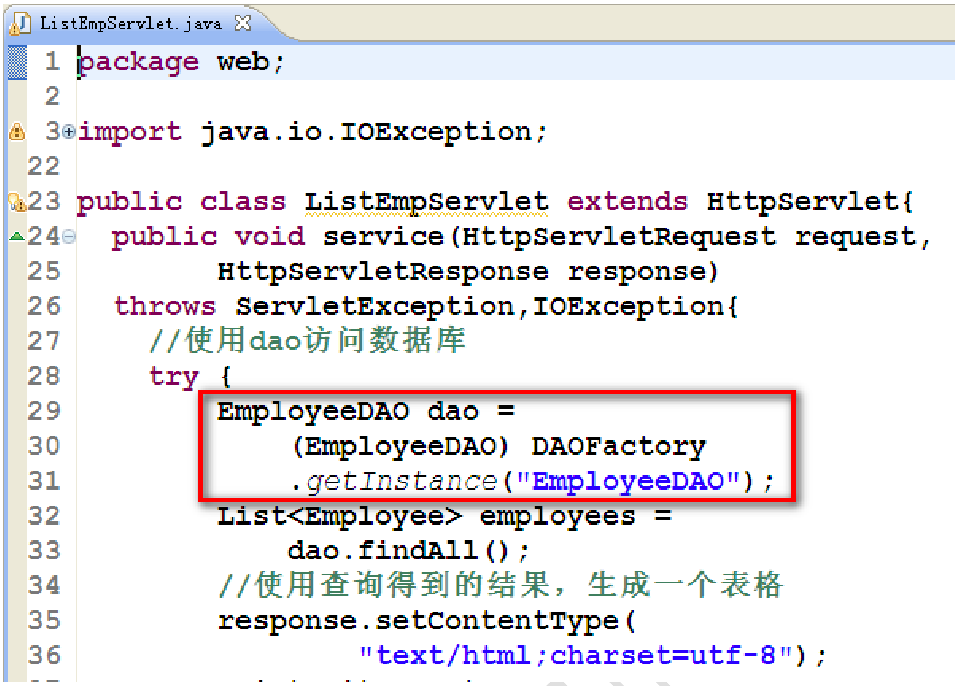
新建 DAOFactory.java，在 DAOFactory 中确定 EmployeeDAO 接口的具体实现类



```
package util;

import dao.impl.EmployeeDAOHibernateImpl;

/**
 * dao工厂，为调用者提供dao实现类的实例
 * @author teacher
 *
 */
public class DAOFactory {
    public static Object getInstance(String type){
        Object obj = null;
        if(type.equals("EmployeeDAO")){
            //obj = new EmployeeDAOJdbcImpl();
            obj = new EmployeeDAOHibernateImpl();
        }
        return obj;
    }
}

```

步骤 2

Servlet 中增删改查操作都使用 DAOFactory 生成具体实例



```
package web;

import java.io.IOException;

public class ListEmpServlet extends HttpServlet{
  public void service(HttpServletRequest request,
        HttpServletResponse response)
  throws ServletException,IOException{
    //使用dao访问数据库
    try {
        EmployeeDAO dao =
            (EmployeeDAO) DAOFactory
            .getInstance("EmployeeDAO");
        List<Employee> employees =
            dao.findAll();
        //使用查询得到的结果，生成一个表格
        response.setContentType(
                "text/html;charset=utf-8");
```

OK

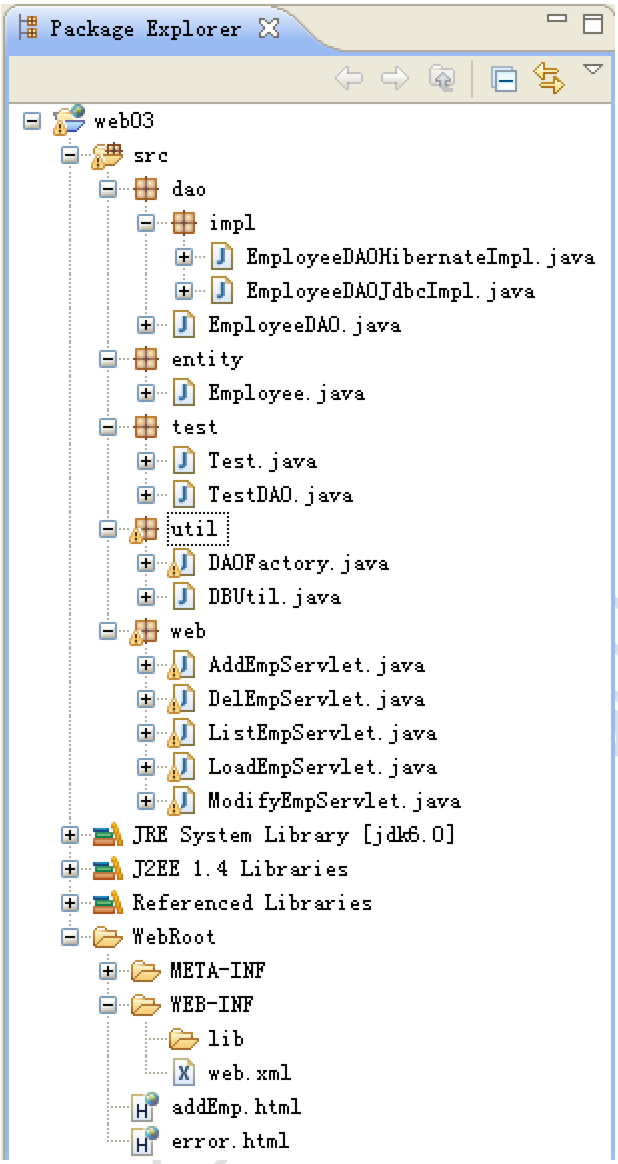
这样我们就不用修改每个 Servlet 了，查询数据库需要不同实现时，只要修改工厂类 DAOFacory 即可。

提问：

这样的方式已经使代码的可扩展性提高很多，但是我们还需要修改 DAOFactory，如果我们不想修改 DAOFacotry 该如何实现？

【案例 3】DAO 及工厂模式

项目结构



```
web03
  src
    dao
      impl
        EmployeeDAOHibernateImpl.java
        EmployeeDAOJdbcImpl.java
      EmployeeDAO.java
    entity
      Employee.java
    test
      Test.java
      TestDAO.java
    util
      DAOFactory.java
      DBUtil.java
    web
      AddEmpServlet.java
      DelEmpServlet.java
      ListEmpServlet.java
      LoadEmpServlet.java
      ModifyEmpServlet.java
  JRE System Library [jdk6.0]
  J2EE 1.4 Libraries
  Referenced Libraries
  WebRoot
    META-INF
    WEB-INF
      lib
      web.xml
    addEmp.html
    error.html
```

参考代码

1) Employee.java

```
package entity;
/**
 * 实体类
 * @author teacher
 */
public class Employee {
	private long id;
```

```
    private String name;
    private double salary;
    private int age;
    public int getAge() {return age;}
    public void setAge(int age) {
        this.age = age;}
    public long getId() {return id;}
    public void setId(long id) {this.id = id;}
    public String getName() {return name;}
    public void setName(String name) {
        this.name = name;}
    public double getSalary() {return salary;}
    public void setSalary(double salary) {
        this.salary = salary;}
}
```

2) EmployeeDAO.java

```
package dao;
import java.util.List;
import entity.Employee;
/**
 * DAO 接口
 * @author teacher
 */
public interface EmployeeDAO {
    /*
     * public void save(Employee e)
     *     throws SQLException;
     * 这种方式不好,因为出现了 jdbc 的异常类，
     * 这就要求实现该方法只能使用 jdbc */
    public List<Employee> findAll()
        throws Exception;
    public void delete(long id)
        throws Exception;
    public void save(Employee e)
        throws Exception;
    public Employee findById(long id)
```

```
            throws Exception;
        public void update(Employee e)
            throws Exception;
}
```

3) EmployeeDAOJdbcImpl.java

```
package dao.impl;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

import util.DBUtil;
import dao.EmployeeDAO;
import entity.Employee;

public class EmployeeDAOJdbcImpl
                    implements EmployeeDAO{

  public List<Employee> findAll()
      throws Exception {
      Connection conn = DBUtil.getConnection();
      Statement stat =
          conn.createStatement();
      ResultSet rst = stat.executeQuery(
              "select * from t_emp");
      List<Employee> employees =
          new ArrayList<Employee>();
      while(rst.next()){
          Employee e = new Employee();
          e.setId(rst.getLong("id"));
          e.setName(rst.getString("name"));
          e.setSalary(rst.getDouble("salary"));
          e.setAge(rst.getInt("age"));
```

```
            employees.add(e);
        }
        DBUtil.close(conn);
        return employees;
    }

    public void delete(long id)
        throws Exception {
        Connection conn = DBUtil.getConnection();
        PreparedStatement prep =
            conn.prepareStatement(
                    "delete from t_emp where id=?");
        prep.setLong(1, id);
        prep.executeUpdate();
        DBUtil.close(conn);
    }

    public void save(Employee e)
        throws Exception {
        Connection conn = DBUtil.getConnection();
        PreparedStatement prep =
            conn.prepareStatement(
                "insert into " +
                "t_emp(name,salary,age) " +
                "values(?,?,?)");
        prep.setString(1, e.getName());
        prep.setDouble(2, e.getSalary());
        prep.setInt(3, e.getAge());
        prep.executeUpdate();
        DBUtil.close(conn);
    }

    public Employee findById(long id)
        throws Exception {
        Connection conn = DBUtil.getConnection();
        PreparedStatement prep =
            conn.prepareStatement(
                    "select * " +
                          "from t_emp where id=?");
        prep.setLong(1, id);
        ResultSet rst = prep.executeQuery();
        Employee e = null;
        if(rst.next()){
            e = new Employee();
            e.setId(id);
            e.setName(rst.getString("name"));
            e.setSalary(rst.getDouble("salary"));
            e.setAge(rst.getInt("age"));
        }
        DBUtil.close(conn);
        return e;
    }

    public void update(Employee e)
        throws Exception {
        Connection conn = DBUtil.getConnection();
        PreparedStatement prep =
            conn.prepareStatement(
                          "update t_emp " +
                          "set name=?,salary=?,age=? " +
                          "where id=?");
        prep.setString(1, e.getName());
        prep.setDouble(2, e.getSalary());
        prep.setInt(3, e.getAge());
        prep.setLong(4, e.getId());
        prep.executeUpdate();
        DBUtil.close(conn);
    }
}
```

4) EmployeeDAOHibernateImpl.java

```
package dao.impl;

import java.util.List;
import dao.EmployeeDAO;
import entity.Employee;

/**
 * dao 的另外一个实现
 * @author teacher
 *
 */
public class EmployeeDAOHibernateImpl
	implements EmployeeDAO{

	public void delete(long id)
	throws Exception {}

	public List<Employee> findAll()
	throws Exception {
		System.out.println(
			"hibernate 实现 findAll...");
		return null;
	}

	public Employee findById(long id)
	throws Exception {
		return null;
	}

	public void save(Employee e)
	throws Exception {}

	public void update(Employee e)
	throws Exception {}
}
```

5) TestDAO.java

```
package test;

import java.util.List;

import dao.EmployeeDAO;
import dao.impl.EmployeeDAOJdbcImpl;
import entity.Employee;

public class TestDAO {
   public static void main(String[] args)
   throws Exception {
       EmployeeDAO dao = new EmployeeDAOJdbcImpl();

       /* 查找全部 */
       List<Employee> employees = dao.findAll();
       System.out.println(employees.size());

       /* 删除 */
       //dao.delete(1);

       /* 插入数据 */
       //Employee   e = new Employee();
       //e.setName("abc9");
       //e.setSalary(1000);
       //e.setAge(22);
       //dao.save(e);

       /* 按 id 查找 */
       //Employee e = dao.findById(7);
       //System.out.println(e.getName());
   }
}
```

6) DAOFactory.java

```
package util;

import dao.impl.EmployeeDAOHibernateImpl;
import dao.impl.EmployeeDAOJdbcImpl;

/**
```

```
 * dao 工厂，为调用者提供 dao 实现类的实例
 * @author teacher
 *
 */
public class DAOFactory {
    public static Object getInstance(String type){
        Object obj = null;
        if(type.equals("EmployeeDAO")){
            //obj = new EmployeeDAOJdbcImpl();
            obj = new EmployeeDAOHibernateImpl();
        }
        return obj;
    }
}
```

7) DBUtil.java

```
package util;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
/**
 * jdbc 工具类，提供获得连接、关闭连接的方法
 * @author teacher
 */
public class DBUtil {
  public static Connection getConnection()
    throws Exception{
    Connection conn = null;
    try {
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(
        "jdbc:mysql://localhost:3306/jd1109db2",
        "root","root");
    } catch (Exception e) {
        e.printStackTrace();
        throw e;
```

```
            }
            return conn;
        }

        public static void close(Connection conn){
            if(conn!=null){
                    try {
                            conn.close();
                    } catch (SQLException e) {
                            e.printStackTrace();
                    }
            }
        }

        //用于测试 DBUtil
        public static void main(String[] args)
            throws Exception {
            System.out.println(getConnection());
        }
}
```

8) AddEmpServlet.java

```
package web;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import dao.EmployeeDAO;
import dao.impl.EmployeeDAOJdbcImpl;
import entity.Employee;

public class AddEmpServlet extends HttpServlet{
    public void service(
                HttpServletRequest request,
                HttpServletResponse response)
        throws ServletException,IOException{
        //这行代码要放在 getParameter()执行之前。
        request.setCharacterEncoding("utf-8");
        String name = request.getParameter("name");
        double salary = Double.parseDouble(
                    request.getParameter("salary"));
        int age = Integer.parseInt(
                    request.getParameter("age"));
        System.out.println("name:" + name);
        System.out.println("salary:" + salary);
        System.out.println("age:" + age);
        //访问数据库
        try {
            EmployeeDAO dao =
                new EmployeeDAOJdbcImpl();
            Employee e = new Employee();
            e.setName(name);
            e.setSalary(salary);
            e.setAge(age);
            dao.save(e);
            response.sendRedirect("list");
        } catch (Exception e) {
            //step1  先记录日志
            e.printStackTrace();
            //step2  抛出
            throw new ServletException(e);
        }
    }
}
```

9) DelEmpServlet.java

```
package web;

import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import dao.EmployeeDAO;
import dao.impl.EmployeeDAOHibernateImpl;
import dao.impl.EmployeeDAOJdbcImpl;

public class DelEmpServlet extends HttpServlet{
    public void service(
            HttpServletRequest request,
            HttpServletResponse response)
    throws ServletException,IOException{
        long id = Long.parseLong(
                request.getParameter("id"));
        try {
            EmployeeDAO dao =
//              new EmployeeDAOJdbcImpl();
                new EmployeeDAOHibernateImpl();
            dao.delete(id);
            response.sendRedirect("list");
        }catch(Exception e){
            e.printStackTrace();
            throw new ServletException(e);
        }
    }
}
```

10) ListEmpServlet.java

```
package web;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import util.DAOFactory;

import dao.EmployeeDAO;
import dao.impl.EmployeeDAOJdbcImpl;
import entity.Employee;

public class ListEmpServlet extends HttpServlet{
   public void service(HttpServletRequest request,
              HttpServletResponse response)
   throws ServletException,IOException{
       //使用 dao 访问数据库
       try {
              EmployeeDAO dao =
                     (EmployeeDAO) DAOFactory
                     .getInstance("EmployeeDAO");
              List<Employee> employees =
                     dao.findAll();
              //使用查询得到的结果，生成一个表格
              response.setContentType(
                            "text/html;charset=utf-8");
```

```
        PrintWriter out =
            response.getWriter();
        out.println(
            "<table border='1' width='60%' " +
            "cellpadding='0' cellspacing='0'>");
        out.println(
                "<tr><td>id</td>" +
                "<td>姓名</td>" +
                "<td>薪水</td>" +
                "<td>年龄</td>" +
                "<td>操作</td></tr>");
        for(int i=0;i<employees.size();i++){
            Employee e = employees.get(i);
            out.println("<tr><td>"
                + e.getId() + "</td><td> "
                + e.getName() + "</td><td>"
                + e.getSalary() + "</td><td> "
                + e.getAge() + "</td><td>" +
                "<a href='del?id="
                    + e.getId() + "'>删除</a>" +
                    "&nbsp;<a href='load?id="
                    + e.getId() +"'>修改" +
                    "</a></td></tr>");
        }
        out.println("</table>");
        out.println(
                "<a href='addEmp.html'>" +
                "增加新雇员</a>");
        out.close();
    } catch (Exception e) {
        e.printStackTrace();
        throw new ServletException(e);
    }
  }
}
```

11) LoadEmpServlet.java

```
package web;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import dao.EmployeeDAO;
import dao.impl.EmployeeDAOJdbcImpl;
import entity.Employee;

public class LoadEmpServlet extends HttpServlet{
  public void service(
        HttpServletRequest request,
        HttpServletResponse response)
    throws ServletException,IOException{
    long id = Long.parseLong(
            request.getParameter("id"));
    try {
        EmployeeDAO dao =
            new EmployeeDAOJdbcImpl();
        Employee e = dao.findById(id);
        response.setContentType(
                "text/html;charset=utf-8");
        PrintWriter out = response.getWriter();
            out.println(
                "<form action='modify?id="
                + id + "' method='post'>");
            out.println(
                "id:" + id + "<br/>");
            out.println(
                "姓名:<input name='name' value='"
                + e.getName() + "'/><br/>");
            out.println(
                "薪水:<input name='salary' value='"
                + e.getSalary() + "'/><br/>");
            out.println(
                "年龄:<input name='age' value='"
                + e.getAge() + "'/><br/>");
            out.println(
                "<input type='submit' " +
                "value='确认'/>");
            out.println("</form>");
            out.close();
    }catch(Exception e){
        e.printStackTrace();
        throw new ServletException(e);
    }
  }
}
```

12) ModifyEmpServlet.java

```
package web;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import dao.EmployeeDAO;
import dao.impl.EmployeeDAOJdbcImpl;
import entity.Employee;

public class ModifyEmpServlet extends HttpServlet{
  public void service(
        HttpServletRequest request,
        HttpServletResponse response)
    throws ServletException,IOException{
    request.setCharacterEncoding("utf-8");
    long id = Long.parseLong(
            request.getParameter("id"));
    String name = request.getParameter("name");
    double salary =
        Double.parseDouble(
            request.getParameter("salary"));
    int age = Integer.parseInt(
            request.getParameter("age"));
    try {
        EmployeeDAO dao =
            new EmployeeDAOJdbcImpl();
        Employee e = new Employee();
        e.setId(id);
        e.setName(name);
        e.setSalary(salary);
        e.setAge(age);
        dao.update(e);
        response.sendRedirect("list");
    }catch(Exception e){
        e.printStackTrace();
        throw new ServletException(e);
    }
  }
}
```

13) web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4"
    xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
    <servlet>
```

```
        <servlet-name>addEmpServlet</servlet-name>
        <servlet-class>web.AddEmpServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>listEmpServlet</servlet-name>
        <servlet-class>web.ListEmpServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>delEmpServlet</servlet-name>
        <servlet-class>web.DelEmpServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>loadEmpServlet</servlet-name>
        <servlet-class>web.LoadEmpServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>modifyEmpServlet</servlet-name>
        <servlet-class>web.ModifyEmpServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>addEmpServlet</servlet-name>
        <url-pattern>/add</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>listEmpServlet</servlet-name>
        <url-pattern>/list</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>delEmpServlet</servlet-name>
        <url-pattern>/del</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>loadEmpServlet</servlet-name>
        <url-pattern>/load</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>modifyEmpServlet</servlet-name>
        <url-pattern>/modify</url-pattern>
    </servlet-mapping>
    <error-page>
        <error-code>500</error-code>
        <location>/error.html</location>
    </error-page>
</web-app>
```

14) addEmp.html

```
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=utf-8">
<title>addEmp</title>
</head>
<body style="font-size:30px;">
    <form action="add" method="post">
        <fieldset>
            <legend>添加雇员</legend>
            姓名:<input name="name"/><br/>
            薪水:<input name="salary"/><br/>
            年龄:<input name="age"/><br/>
            <input type="submit" value="确认"/>
        </fieldset>
    </form>
</body>
</html>
```

15) error.html

```
<html>
<head>
<meta http-equiv="Content-Type"
    content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body style="font-size:30px;color:red;">
    发生了系统错误，请稍后
    <a href="addEmp.html">重试</a>
</body>
</html>
```