

## 知识点列表

编号	名称	描述	级别
1	转发_图示演示	理解转发的流程	*
2	转发_案例演示	通过案例掌握转发的使用	**
3	Include 指令	了解 include 指令的常用使用	**
4	表单中文处理步骤	按步骤处理中文，在今后的项目中一般没有问题	**
5	在 MyEclipse 中访问 MySql	掌握工具的新用法	**
6	相对路径&绝对路径	理解绝对路径和相对路径	*
7	注册与登陆模块	要求需要熟练掌握，在给定的 html 文件下熟练完成。	***

注：    "\*\*\*"理解级别    "\*\*\*\*"掌握级别    "\*\*\*\*\*"应用级别

## 目录

1. 转发*	错误！未定义书签。
1.1. 图示演示 *	3
1.2. 案例演示 **	4
1.3. Include 指令 **	9
1. 转发 *	3
1.1. 图示演示 *	4
1.2. 案例演示 **	5
1.3. Include 指令 **	10
2. 表单中文问题 **	12
2.1. 表单中文处理步骤 **	12
【案例 1】雇员的增删改查 **	13
2.2. 在 MyEclipse 中访问 MySql **	13
3. 路径问题 *	19
【案例 2】相对路径&&绝对路径 *	20
4. 课堂练习：注册与登录模块 ***	29
【案例 3】注册与登录模块 ***	30

1.	转发 *	4
1.1.	图示演示 *	4
1.2.	案例演示 **	5
1.3.	Include 指令 **	10
2.	表单中文问题 **	12
2.1.	表单中文处理步骤 **	12
	【案例 1】雇员的增删改查 **	13
2.2.	在 MyEclipse 中访问 MySql **	13
3.	路径问题 *	19
	【案例 2】相对路径&&绝对路径 *	20
4.	课堂练习：注册与登录模块 ***	29
	【案例 3】注册与登录模块 ***	30
1.	转发 *	4
1.1.	图示演示 *	5
1.2.	案例演示 **	6
1.3.	Include 指令 **	11
2.	表单中文问题 **	13

2.1.	表单中文处理步骤 **	13
	【案例 1】雇员的增删改查 **	14
2.2.	在 MyEclipse 中访问 MySql **	14
3.	路径问题 *	20
	【案例 2】相对路径&&绝对路径 *	21
4.	课堂练习：注册与登录模块 ***	30
	【案例 3】注册与登录模块 ***	31
1.	转发 *	5
1.1.	图示演示 *	6
1.2.	案例演示 **	7
1.3.	Include 指令 **	12
2.	表单中文问题 **	14
2.1.	表单中文处理步骤 **	14
	【案例 1】雇员的增删改查 **	15
2.2.	在 MyEclipse 中访问 MySql **	15
3.	路径问题 *	21
	【案例 2】相对路径&&绝对路径 *	22
4.	课堂练习：注册与登录模块 ***	31

【案例 3】注册与登录模块 *** .....	32
-------------------------	----

## 1. 转发 \*

### 1) 什么是转发?

一个 web 组件(jsp/servlet)将未完成的处理转交给另一个 web 组件继续处理。转发的各个组件会共享 request 和 response 对象。

### 2) 如何转发?

#### step1

先绑定数据

```
request.setAttribute(String name,Object obj);
```

跟绑定相关的另外两个方法

```
//如果 name 对应的值不存在，返回 null。
```

```
Object request.getAttribute(String name);
```

```
request.removeAttribute(String name);
```

#### step2

获得转发器，转发

```
RequestDispatcher rd = request.getRequestDispatcher(String uri);
```

```
rd.forward(request,response);
```

### 3) 编程中要注意的问题

转发之前，不能够执行 out.close 或者 out.flush。

转发之前，会将 response 中缓存的数据先清空。

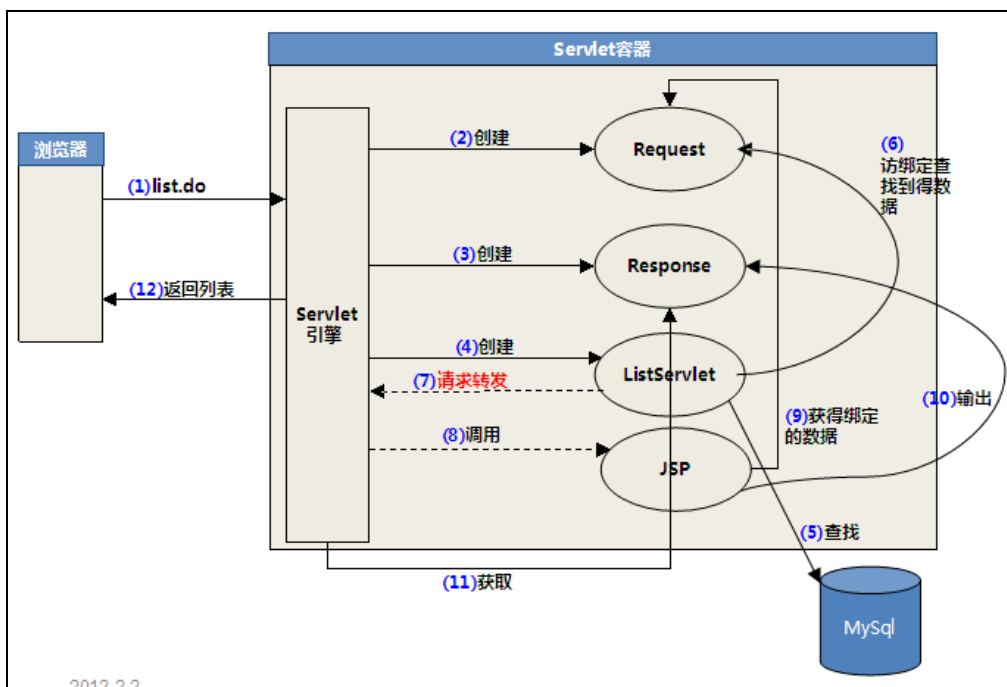
### 4) 转发的特点

- a. 转发的目的地只能是同一个应用内部的某个组件
- b. 转发之后，浏览器地址栏的地址没有变化
- c. 转发所涉及的各个组件可以共享同一个 request , response 对象

## 1.1. 图示演示 \*

### 转发

- 1) 浏览器访问 list.do
- 2) Servlet 引擎创建 Request 对象
- 3) Servlet 引擎创建 Response 对象
- 4) Servlet 引擎调用 ListServlet 对象
- 5) ListServlet 访问数据库，查找数据
- 6) ListServlet 绑定查找到的数据到 Request 对象上
- 7) 不能使用重定向，ListServlet 只能向 Servlet 引擎请求**转发**
- 8) Servlet 引擎调用 JSP
- 9) JSP 从 Request 对象上获取绑定的数据，并生成页面
- 10) JSP 将生成的页面数据输入到 Response 对象
- 11) Servlet 引擎从 Response 对象中获取页面数据
- 12) Servlet 引擎将获取的页面数据发送给浏览器



思考：转发与重定向的区别？

## 1.2. 案例演示 \*\*

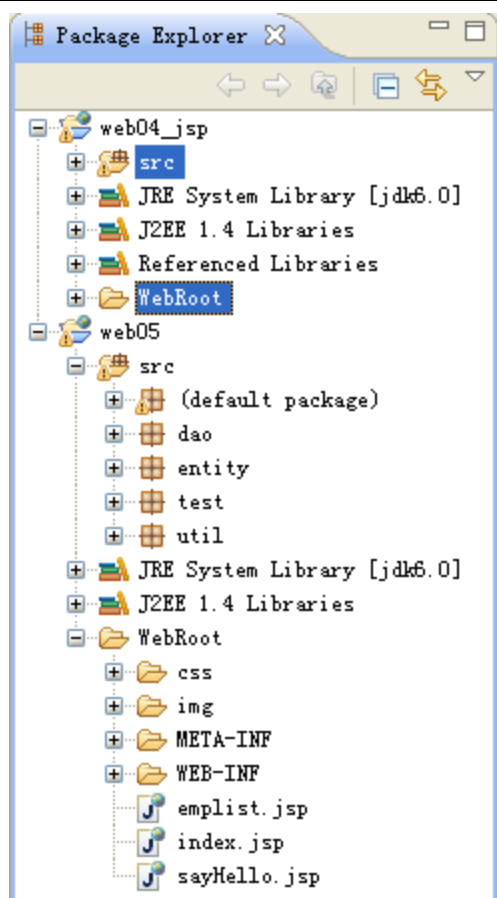
重构 day04 的【案例 4】JSP 实现员工增删改查

由 Servlet 负责处理数据库操作，JSP 负责显示

1) 新建项目 web05

将 web04\_jsp 下的 **src** 目录、**WebRoot** 目录及之下所有内容拷贝到 web05 项目下

请下载 web04\_jsp.zip



2) 导入 mysql 数据库驱动的 Jar 包

3) 拷贝项目 web04\_2 中的 **web.ActionServlet** 和 **web.xml** 到 web05 项目中

请下载 web04\_2.zip

查询雇员信息

4) 修改 ActionServlet 的内容

因为 Servlet 不擅长生成页面，所以**转发**给 JSP 来处理页面



```

ActionServlet.java X
17
18 public class ActionServlet extends HttpServlet {
19     public void service(
20         HttpServletRequest request,
21         HttpServletResponse response)
22         throws ServletException, IOException {
23         request.setCharacterEncoding("utf-8");
24         response.setContentType(
25             "text/html;charset=utf-8");
26         PrintWriter out = response.getWriter();
27         String uri = request.getRequestURI();
28         String path =
29             uri.substring(
30                 uri.lastIndexOf("/"),
31                 uri.lastIndexOf("."));
32         if (path.equals("/list")) {
33             // 使用dao访问数据库
34             try {
35                 EmployeeDAO dao =
36                     (EmployeeDAO) DAOFactory
37                         .getInstance("EmployeeDAO");
38                 List<Employee> employees =
39                     dao.findAll();

```

```

40 // 使用查询得到的结果，生成一个表格
41 /*
42  * 因为Servlet不擅长生成页面，所以
43  */
44 //1. 将数据绑定到Request上
45 request.setAttribute(
46     "employees", employees);
47 //2. 转发
48 //2.1 获得一个转发器
49 RequestDispatcher rd =
50     request.getRequestDispatcher(
51         "emplist.jsp");
52 //2.2 转发
53 rd.forward(request, response);
54
55 } catch (Exception e) {
56     e.printStackTrace();
57     throw new ServletException(e);
58 }
59 } else if (path.equals("/add")) {
60     String name = request.getParameter("name")
61     double salary =
62     Double.parseDouble(

```

#### 5) 修改 emplist.jsp

```

emplist.jsp
34      欢迎
35      </h1>
36      <table class="table">
37          <tr class="table_header">
38              <td>ID</td>
39              <td>姓名</td>
40              <td>薪水</td>
41              <td>年龄</td>
42              <td>操作</td>
43          </tr>
44      <%
45          //访问数据库
46          //从request对象当中获得绑定的数据
47          List<Employee> employees =
48              (List<Employee>) request
49                  .getAttribute("employees");
50
51          for(int i=0;i<employees.size();i++){
52              Employee e = employees.get(i);
53          }
54          <tr class="row<%= (i%2+1) %>">
55              <td>
56                  <%=e.getId() %>
57              </td>
58              <td>

```

6) 访问 <http://localhost:8080/web05/list.do>  
完成员工的查询！

main2009/11/20

欢迎

ID	姓名	薪水	年龄	操作
6	ls	Y 8887.0	21	删除 更新
7	ww	Y 3001.0	22	删除 更新
10	you	Y 5000.0	23	删除 更新
12	wwq	Y 20000.0	30	删除 更新

添加雇员

ABC@126.com

7) 增删改类似完成。

### 1.3.Include 指令 \*\*

对于页面的公共部分，我们可以使用相同的 jsp 文件，并使用 include 指令导入，如此可以实现代码的优化。

比如 emplist.jsp 和 addEmp.jsp 的“头部”和“尾部”相同

1) emplist.jsp

main2009/11/20

欢迎

ID	姓名	薪水	年龄	操作
7	ww	3001.0	22	删除 更新
10	you	5000.0	23	删除 更新
12	wwq	20000.0	30	删除 更新

添加雇员

ABC@126.com

2) addEmp.jsp

Main
2009/11/20

添加雇员:
  
姓名: 
  
薪水: 
  
年龄:

ABC@126.com

## 演示

### 1) 修改 emplist.jsp

删除注释部分，加入语句`<%@ include file="head.jsp" %>`

```

emplist.jsp  head.jsp
10  <body>
11  <div id="wrap">
12      <!--
13          <div id="top_content">
14              <div id="header">
15                  <div id="righthead" >
16                      <p>
17                          2009/11/20
18                      <br />
19                      </p>
20              </div>
21              <div id="topheader">
22                  <h1 id="title">
23                      <a href="#">main</a>
24                  </h1>
25              </div>
26              <div id="navigation">
27              </div>
28          </div>
29      -->
30      <%@include file="head.jsp" %>
31
32  <div id="content">
33      <p id="whereami">
34      </p>

```

## 2) 新建 head.jsp

```
head.jsp
1<div id="header">
2    <div id="righthead">
3        <p>
4            2009/11/20
5        <br />
6    </p>
7    </div>
8    <div id="topheader">
9        <h1 id="title">
10            <a href="#">main</a>
11        </h1>
12    </div>
13    <div id="navigation">
14    </div>
15 </div>
16
```

## 3) 同理修改 addEmp.jsp 页面及其他页面的实现代码的优化

## 2. 表单中文问题 \*\*

### 2.1. 表单中文处理步骤 \*\*

#### step1

jsp 文件，要添加

```
<%@page pageEncoding="utf-8"
contentType="text/html;charset=utf-8"%>
```

表单设置 method="post"。

#### step2

在 servlet 类当中，添加 request.setCharacterEncoding("utf-8");

#### step3

如果要访问数据库:

- a. 保证数据库能够正常地保存中文。  
对于 mysql 数据库  
create database dbname default character set utf8;
- b. 使用 jdbc 访问数据库时，必须要保证 jdbc 驱动程序能够识别数据库中保存数据的编码。

jdbc:mysql://localhost:3306/jd1109db2?  
useUnicode=true&characterEncoding=utf8

### 【案例 1】雇员的增删改查 \*\*

实现员工的增删改查业务。

请下载 web05.zip 并访问 <http://localhost:8080/web05/list.do>

本案例涉及的知识点包括：

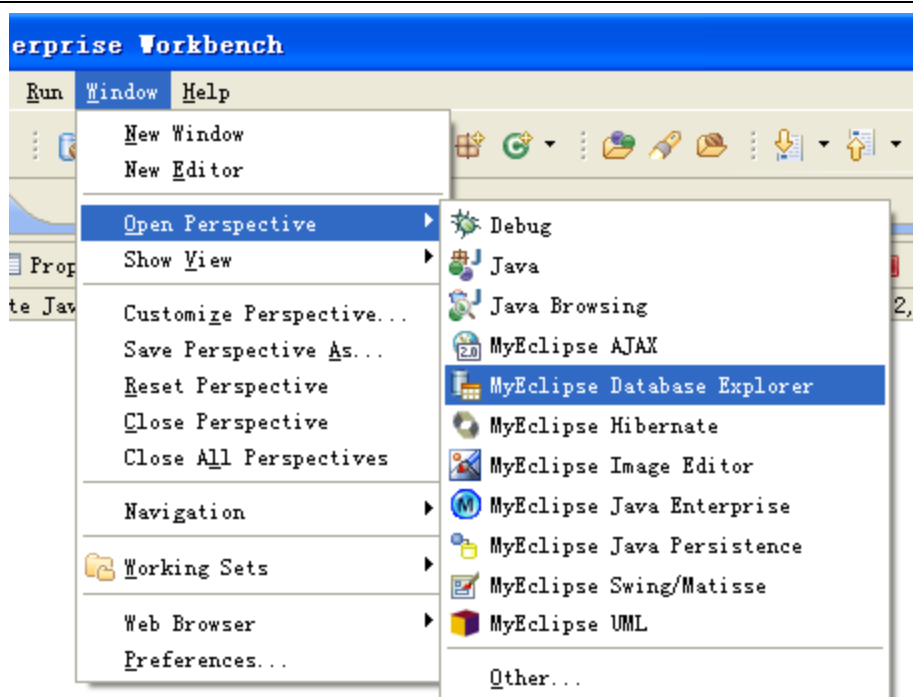
- 1) 转发
  - a) 转发的步骤
  - b) 转发需要注意的问题  
转发之前，不能够执行 out.close 或者 out.flush。  
转发之前，会将 response 中缓存的数据先清空。
- 2) Include 指令
- 3) 表单的中文处理

## 2.2. 在 MyEclipse 中访问 MySql \*\*

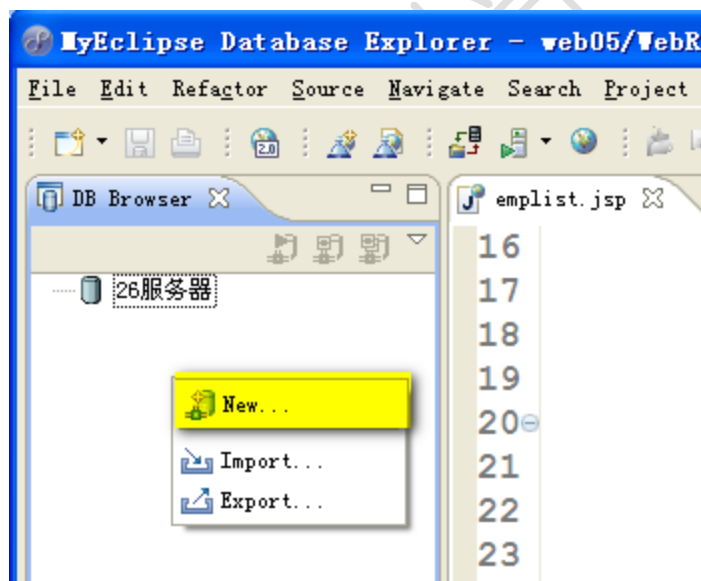
### 1) Window 选项卡

选择 Open PerSpective

选择 MyEclipse Database Explorer，打开数据库视图



2) 在 DB Browser 中单击右键，选择 New ...

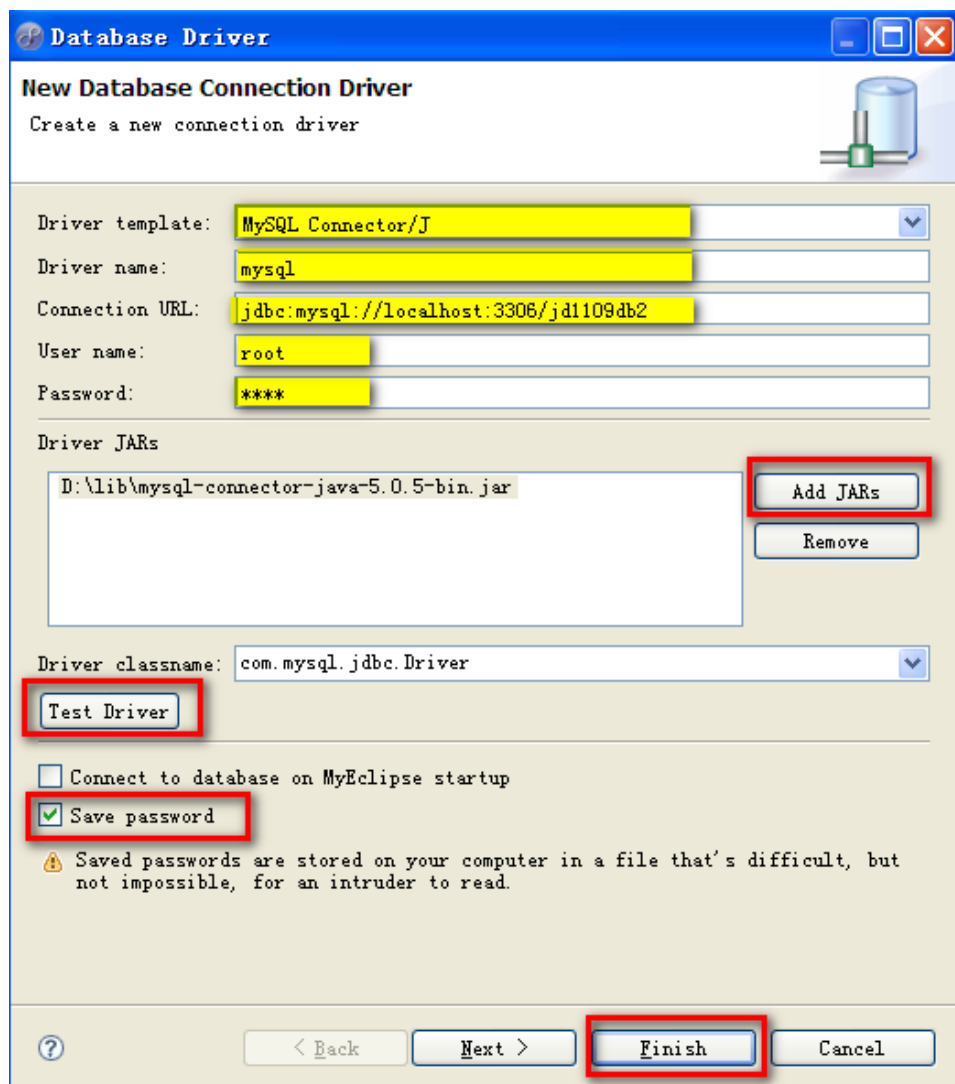


3) 在 Database Driver 界面配置

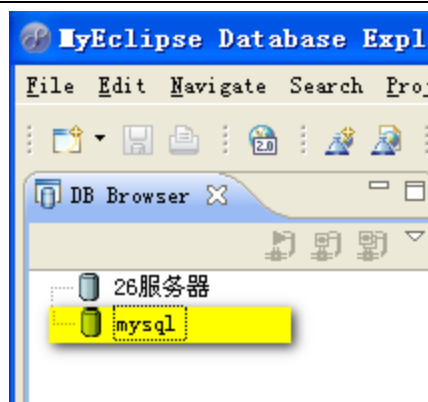
- a. Driver template: 连接的数据库模板
- b. Driver name: 自定义的名字
- c. Connection URL : URL



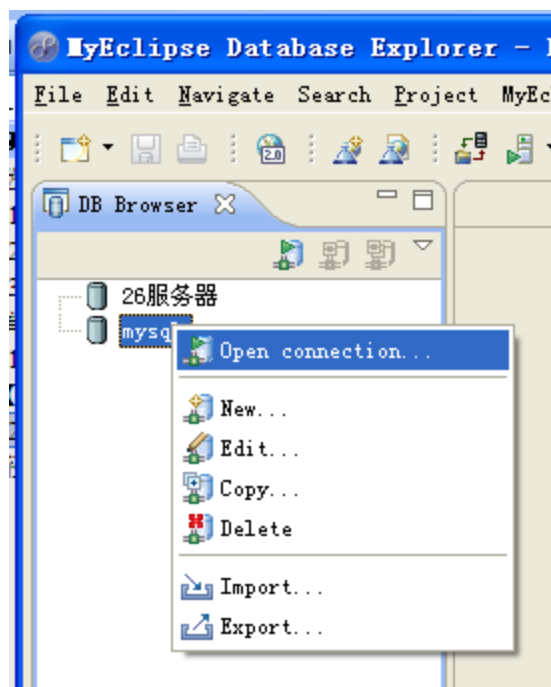
- d. User name : 用户名
- e. Passwd: 密码
- f. 【Add JARs】 : 添加 Mysql 的 Jar 包
- g. 【Test Driver】 : 测试一下驱动 ( 可选 )
- h. 【Save Passwd】 : 保存密码下次访问更方便



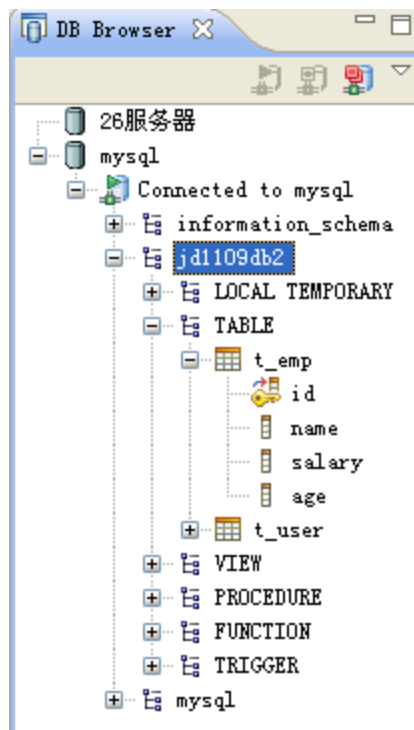
4) 点击【Finish】后会出现配置好的数据库连接 “mysql”



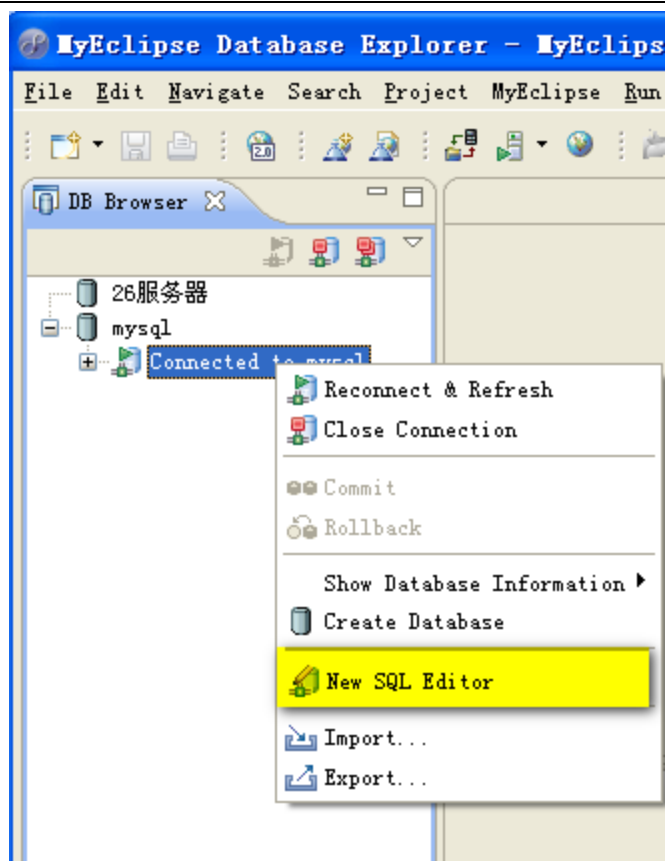
5) 在新建的数据库连接 **mysql** 上点击右键，选择 **open connection...**



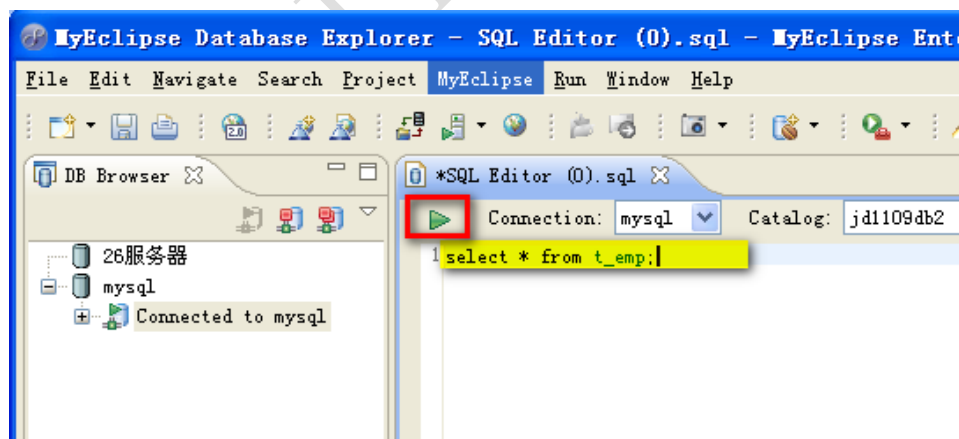
6) 展开可以看到数据库及表结构



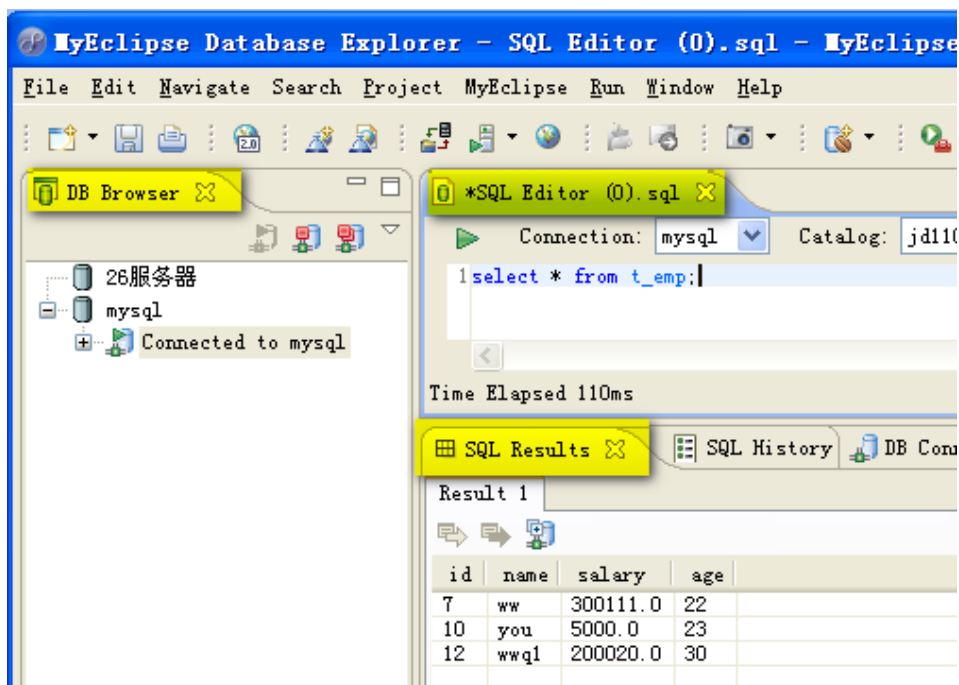
7) 在“Connected to mysql”上点击右键，选择“new SQL Editor”



8) 在出现的“SQL Editor”中输入 SQL 语句，点击“绿色小三角”执行



9) 显示结果



### 3. 路径问题 \*

#### 1) 链接地址、表单提交、重定向、转发的路径应该如何写？

- 链接地址：`<a href=""></a>`
- 表单提交：`<form action="">`
- 重定向：`response.sendRedirect("");`
- 转发：`getRequestDispatcher("");`

#### 2) 相对路径与绝对路径

相对路径：

不以"/"开头的路径，比如：`<a href="del.do"></a>`

在当前路径下进行跳转

绝对路径：

以"/"开头的路径，比如：`<a href="/appname/del.do"></a>`

**建议：因为相对路径较易出错，建议在实际开发中尽量使用绝对路径**

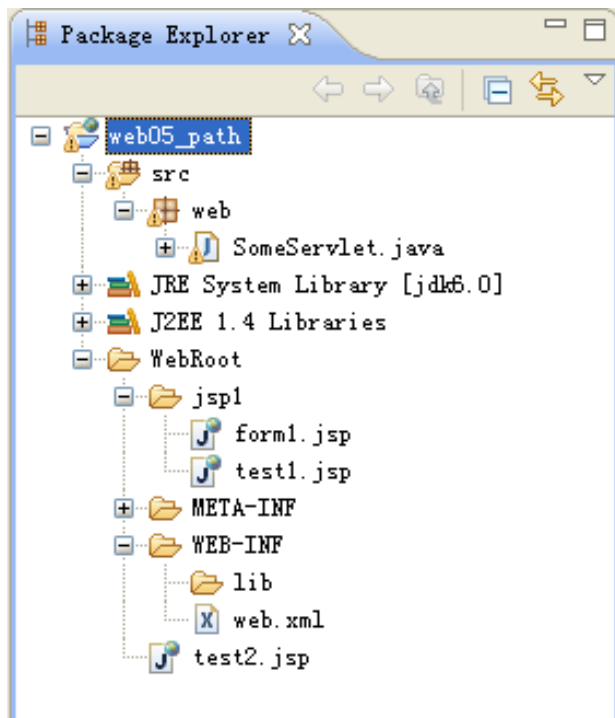
#### 3) 怎样写绝对路径？\*\*\*

链接、表单提交、重定向，绝对路径要从应用名开始写。

转发要从应用名之后开始写。

## 【案例 2】相对路径&&绝对路径 \*

### 1) 项目结构



## ● 链接

### 2) jsp1/test1.jsp

```
<%@page pageEncoding="utf-8"
    contentType="text/html; charset=utf-8"%>
<html>
    <head></head>
    <body style="font-size:30px;">

        test1页面<br/>

        <h3>相对路径</h3>
        <a href=" ../test2.jsp">访问test2</a>

        <h3>绝对路径演示一</h3>
        <a href="/web05_path/test2.jsp">
            访问test2</a>
```

```
<h3>绝对路径演示二</h3>
<a href="<%=request.getContextPath()%>/test2.jsp">
    访问test2</a>
</body>
</html>
```

### 3) test2.jsp

```
<%@page pageEncoding="utf-8"
    contentType="text/html; charset=utf-8"%>
<html>
    <head></head>
    <body style="font-size:30px;">

        test2页面<br/>

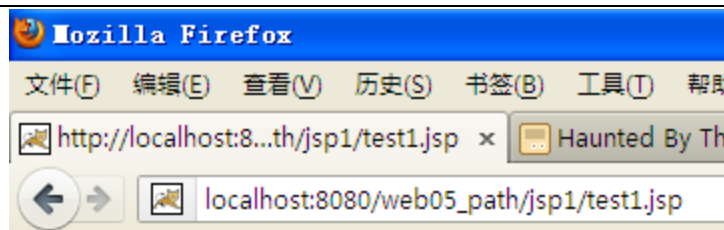
        <h3>相对路径</h3>
        <a href="jsp1/test1.jsp">
            访问test1页面</a>

        <h3>绝对路径演示一</h3>
        <a href="/web05_path/jsp1/test1.jsp">
            访问test1页面</a>

        <h3>绝对路径演示二</h3>
        <a href="<%=request.getContextPath()%>/jsp1/test1.jsp">
            访问test1页面</a>

    </body>
</html>
```

### 4) 访问 http://localhost:8080/web05\_path/jsp1/test1.jsp



test1页面

## 相对路径

[访问test2](#)

## 绝对路径演示一

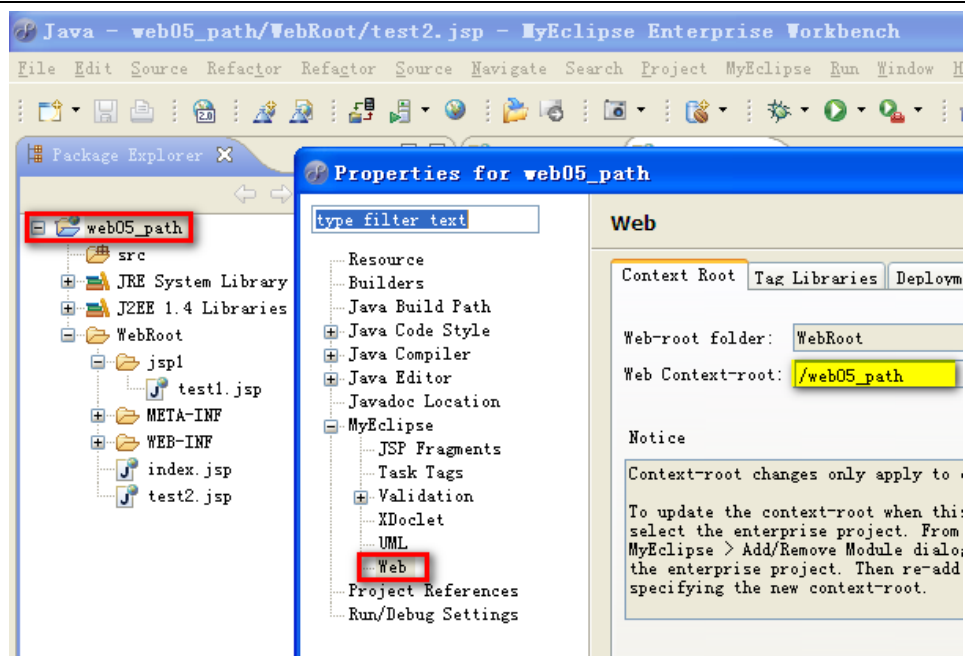
[访问test2](#)

## 绝对路径演示二

[访问test2](#)

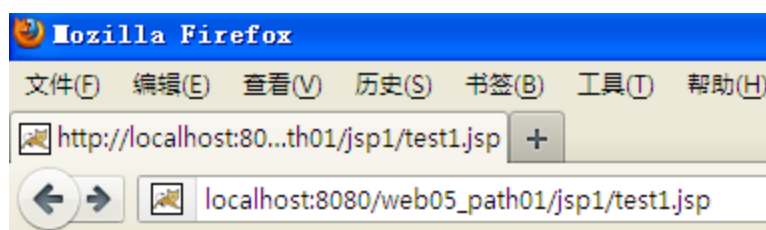
- 5) 小技巧 给 web 项目的应用改名
- 在项目上点右键 “Prpoerties”
  - 找到 web 选项，将 “Web Context-root” 中默认的名字更改  
比如改为 web05\_path01





c. 重新部署

d. 访问 [http://localhost:8080/web05\\_path01/jsp1/test1.jsp](http://localhost:8080/web05_path01/jsp1/test1.jsp)



test1页面

## 相对路径

[访问test2](#)

### 绝对路径演示一

点击会报404

[访问test2](#)

### 绝对路径演示二

[访问test2](#)

依然OK

- 表单

6) 新建 jsp1/form1.jsp

```
form1.jsp
1 <%@page pageEncoding="utf-8"
2   contentType="text/html; charset=utf-8"%>
3 <html>
4   <head></head>
5   <body style="font-size:30px;"
6     <!-- 相对路径
7     <form action=
8       "../some">
9       username:<input name="username"/>
10      <input type="submit"/>
11    </form>
12    -->
13
14    <!-- 绝对路径 -->
15    <form action=
16      "<%=request.getContextPath()%>/some">
17      username:<input name="username"/>
18      <input type="submit"/>
19    </form>
20  </body>
21 </html>
22
```

## 7) 新建 web.SomeServlet

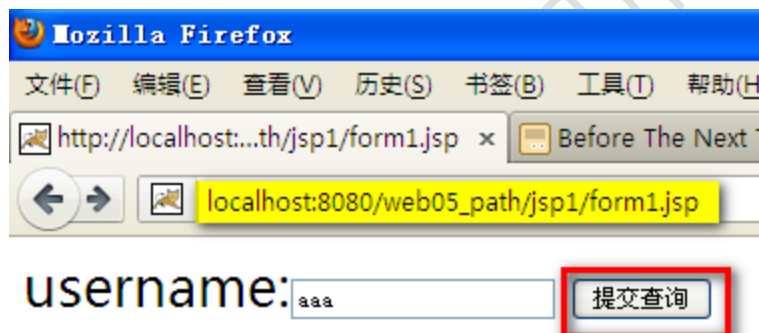
```
SomeServlet.java
1 package web;
2
3 import java.io.IOException;
4
5
6
7
8
9
10
11 public class SomeServlet extends HttpServlet {
12
13   public void service(HttpServletRequest request,
14     HttpServletResponse response)
15     throws ServletException, IOException {
16     String username =
17       request.getParameter("username");
18     System.out.println("username:" + username);
19   }
20 }
21
```

## 8) web.xml

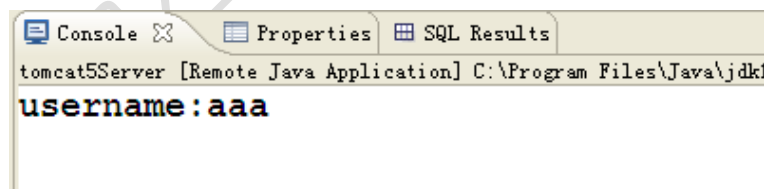
```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="2.4"
3     xmlns="http://java.sun.com/xml/ns/j2ee"
4     xmlns:xsi="http://www.w3.org/2001/XMLSchema-in:
5     xsi:schemaLocation="http://java.sun.com/xml/ns,
6     http://java.sun.com/xml/ns/j2ee/web-app_2_4.xs
7 <servlet>
8     <servlet-name>SomeServlet</servlet-name>
9     <servlet-class>web.SomeServlet</servlet-class>
10 </servlet>
11
12 <servlet-mapping>
13     <servlet-name>SomeServlet</servlet-name>
14     <url-pattern>/some</url-pattern>
15 </servlet-mapping>
16 </web-app>
  
```

#### 9) 访问



#### 控制台打印用户输入的信息



- 重定向

#### 10) 修改 web.SomeServlet

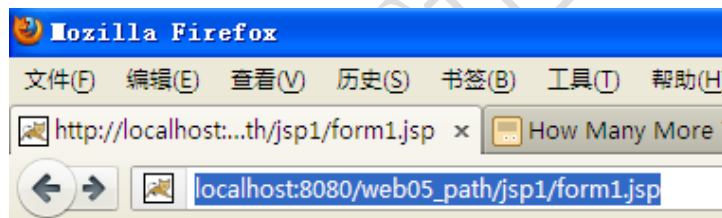
重定向到 jsp1/test1.jsp

```

1 package web;
2
3 import java.io.IOException;
10
11 public class SomeServlet extends HttpServlet {
12
13     public void service(HttpServletRequest request,
14                         HttpServletResponse response)
15         throws ServletException, IOException {
16         String username =
17             request.getParameter("username");
18         System.out.println("username:" + username);
19
20         //重定向
21         response.sendRedirect(
22             request.getContextPath()
23             + "/jsp1/test1.jsp");
24     }
25 }
26

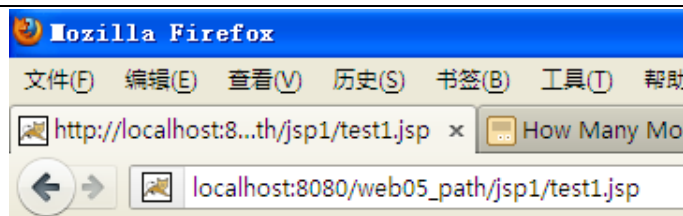
```

11) 访问 [http://localhost:8080/web05\\_path/jsp1/form1.jsp](http://localhost:8080/web05_path/jsp1/form1.jsp)



username:

用户点击提交按钮后，访问地址改变了



test1页面

## 相对路径

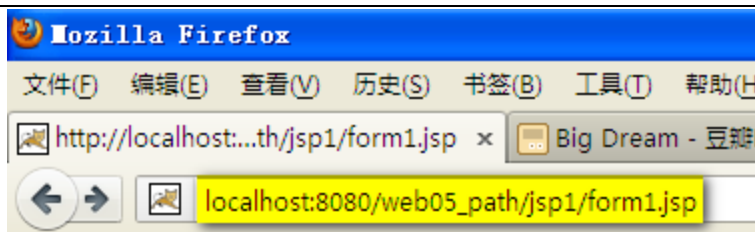
- 转发

12) 修改 web.SomeServlet



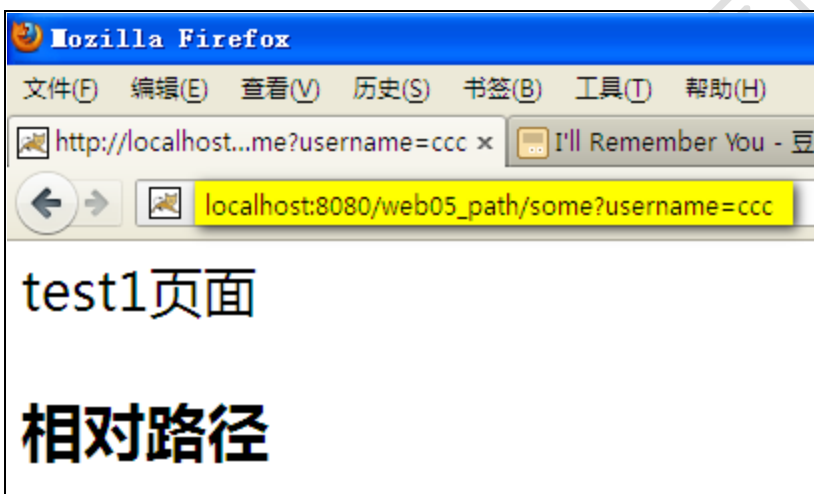
13) 重新部署

14) 访问 [http://localhost:8080/web05\\_path/jsp1/form1.jsp](http://localhost:8080/web05_path/jsp1/form1.jsp)



username:

#### 15) 用户点击提交按钮，访问地址没有改变



## 4. 课堂练习：注册与登录模块 \*\*\*

### 注册

将表单中的用户数据插入到数据库。在插入之前，要查询用户名是否已经存在，如果存在，则跳转到注册页面，提示用户“用户名已经存在，重新输入”。注册成功，跳转到登录页面。

### 步骤说明

#### step1 建表

```
create table t_user(
    id bigint primary key auto_increment,
```

```
username varchar(50) unique,  
name varchar(50),  
pwd varchar(20),  
gendar char(1));
```

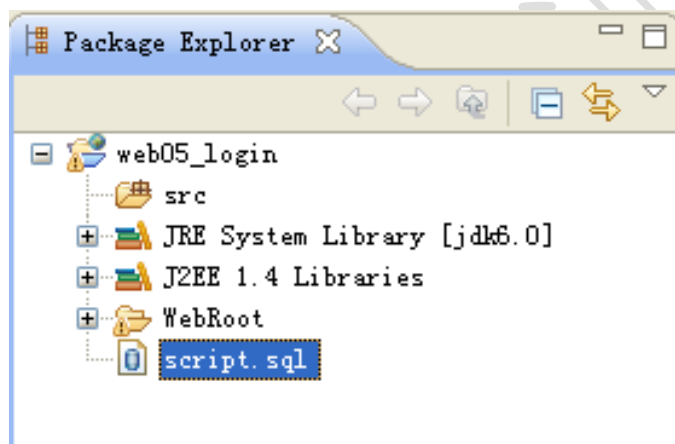
**step2** 创建一个 web 工程

- a. User 实体类
- b. dao 类
- c. regist.jsp
- d. ActionServlet

**【案例 3】注册与登录模块 \*\*\***

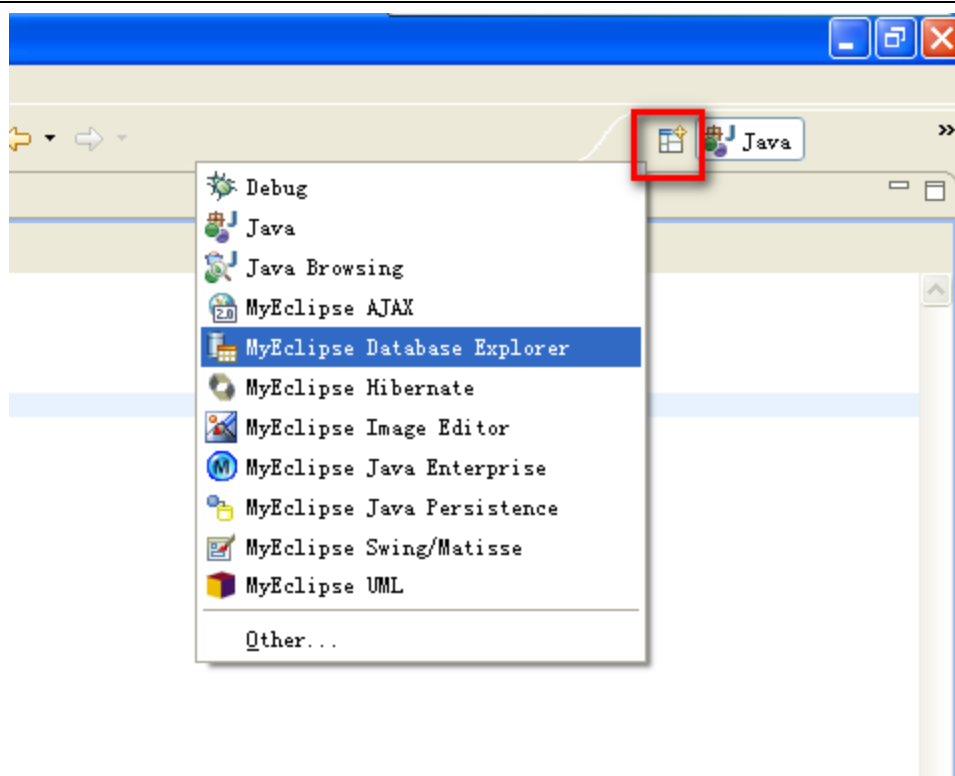
**1) 建数据库表**

**a. 新建脚本文件 script.sql**

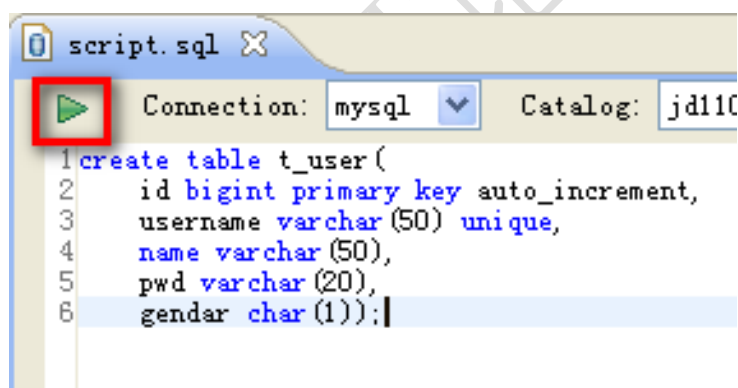


**b. 切换到 MyEclipse Database Explorer 视图中打开数据库连接**



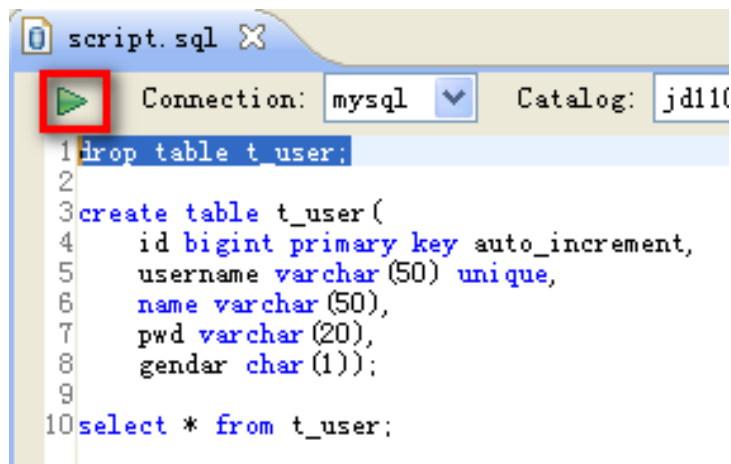


### c. 执行 SQL 脚本



### d. 小技巧

将要执行的代码“选中”后，点击右键，则只执行选中部分的代码



#### e. script.sql

```
drop table t_user;

create table t_user(
    id bigint primary key auto_increment,
    username varchar(50) unique,
    name varchar(50),
    pwd varchar(20),
    gendar char(1));

select * from t_user;
```

#### 2) 新建实体类 entity.User

```
package entity;

public class User {
    private long id;
    private String username;
    private String name;
    private String pwd;
    private String gendar;
    public User() {
    }
    public User(String username, String name,
        String pwd, String gendar) {
        this.username = username;
    }
}
```

```
        this.name = name;
        this.pwd = pwd;
        this.gendar = gendar;
    }

    public String getGendar() {
        return gendar;
    }

    public void setGendar(String gendar) {
        this.gendar = gendar;
    }

    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 String getPwd() {
        return pwd;
    }

    public void setPwd(String pwd) {
        this.pwd = pwd;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }
}
```

### 3) 拷贝 util.DBUtil 到项目下

```
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?" +
                "useUnicode=true&characterEncoding=utf8",
                "root","1234");
        } 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();
            }
        }
    }
}
```

```
/**
 * @param args
 * @throws Exception
 */
public static void main(String[] args) throws Exception {
    // TODO Auto-generated method stub
    System.out.println(getConnection());
}
}
```

#### 4) 新建 dao.UserDAO

```
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

import util.DBUtil;
import entity.User;

public class UserDAO {
    public void save(User user) throws Exception{
        Connection conn = DBUtil.getConnection();
        PreparedStatement prep =
            conn.prepareStatement(
                "insert into t_user(" +
                "username,name,pwd,gendar)" +
                " values(?,?,?,?)");
        prep.setString(1, user.getUsername());
        prep.setString(2, user.getName());
        prep.setString(3, user.getPwd());
        prep.setString(4, user.getGendar());
        prep.executeUpdate();
        DBUtil.close(conn);
    }
}
```

```
public User findByUsername(String username)
throws Exception{
    User user = null;
    Connection conn = DBUtil.getConnection();
    PreparedStatement prep =
        conn.prepareStatement(
            "select * " +
            "from t_user " +
            "where username=?");
    prep.setString(1, username);
    ResultSet rst = prep.executeQuery();
    if(rst.next()){
        user = new User();
        user.setName(rst.getString("name"));
        user.setUsername(username);
        user.setGendar(rst.getString("gendar"));
        user.setPwd(rst.getString("pwd"));
        user.setId(rst.getLong("id"));
    }
    DBUtil.close(conn);
    return user;
}
}
```

## 5) 新建 TestDAO

```
package test;

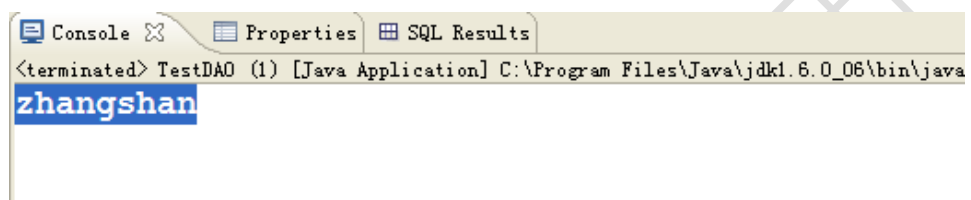
import dao.UserDAO;
import entity.User;

public class TestDAO {
    public static void main(String[] args)
    throws Exception {
        // TODO Auto-generated method stub
        UserDAO dao = new UserDAO();
        User user =
            new User("zs","zhangshan","test","m");
    }
}
```

```
        dao.save(user);  
        User user2 = dao.findByUsername("zs");  
        System.out.println(user2.getName());  
    }  
  
}
```

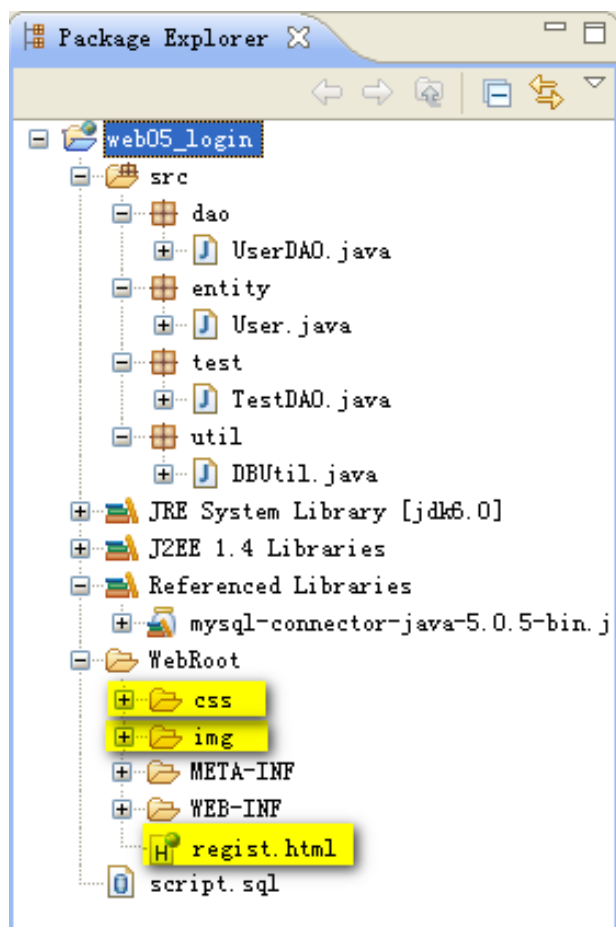
## 6) 测试

### 运行 TestDAO.java



## 7) 拷贝 css、img 目录、regist.html 到项目下

请参照 day04 资料下载 emsdemo.zip



#### 8) 将注册页面 **regist.html** 修改为 **regist.jsp**

```
<%@page pageEncoding="utf-8"
    contentType="text/html; charset=utf-8"%>
<html>
    <head>
        <title>regist</title>
        <meta http-equiv="Content-Type"
            content="text/html; charset=UTF-8">
        <link rel="stylesheet" type="text/css"
            href="css/style.css" />
    </head>
    <body>
        <div id="wrap">
            <div id="top_content">
```



```

<div id="header">
    <div id="righthead">
        <p>
            2009/11/20
        <br />
        </p>
    </div>
    <div id="topheader">
        <h1 id="title">
            <a href="#">main</a>
        </h1>
    </div>
    <div id="navigation"></div>
</div>
<div id="content">
    <p id="whereami"></p>
    <h1>
        注册
    </h1>
    <form action="regist.do" method="post">
        <table cellpadding="0"
            cellspacing="0"
            border="0"
            class="form_table">
            <tr>
                <td valign="middle" align="right">
                    用户名:
                </td>
                <td valign="middle" align="left">
                    <input
                        type="text"
                        class="inputgri"
                        name="username" />
                <%
                    String msg =
                        (String)request
                            .getAttribute("regist_error");
                %>

```

```

        <span style="color: red; font-style: italic;">
        <%= (msg == null ? "" : msg)%>
        </span>
    </td>
</tr>
<tr>
    <td valign="middle" align="right">
        真实姓名:
    </td>
    <td valign="middle" align="left">
        <input type="text"
            class="inputgri"
            name="name" />
    </td>
</tr>
<tr>
    <td valign="middle" align="right">
        密码:
    </td>
    <td valign="middle" align="left">
        <input type="password"
            class="inputgri"
            name="pwd" />
    </td>
</tr>
<tr>
    <td valign="middle" align="right">
        性别:
    </td>
    <td valign="middle" align="left">
        男
        <input type="radio"
            class="inputgri"
            name="gender"
            value="m"
            checked="checked" />
        女
        <input type="radio"

```

```

        class="inputgri"
        name="gendar"
        value="f" />

    </td>
</tr>

<tr>
    <td valign="middle" align="right">
        验证码:
        
        <a href="javascript:;"
            onclick="document
                .getElementById('num')
                .src = 'image?'+(new Date()).getTime()">
            换一张</a>
    </td>
    <td valign="middle" align="left">
        <input type="text"
            class="inputgri"
            name="number" />
    </td>
</tr>
</table>
<p>
    <input type="submit"
        class="button"
        value="确认" />
</p>
</form>

</div>
</div>
<div id="footer">
    <div id="footer_bg">
        ABC@126.com
    </div>
</div>
</div>
</body>

```

```
</html>
```

## 9) 新建 web.ActionServlet

### 写注册逻辑

```
package web;

import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

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

import util.DBUtil;
import dao.UserDAO;
import entity.User;

public class ActionServlet extends HttpServlet {

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

        request.setCharacterEncoding("utf-8");
        String uri = request.getRequestURI();
        String path = uri.substring(uri.lastIndexOf("/"),
            uri.lastIndexOf("."));
        if(path.equals("/regist")){
            String username =
                request.getParameter("username");
            UserDAO dao = new UserDAO();
            try {
```

```

        User user = dao.findByUsername(username);
        if(user == null){
            //用户名不存在，插入操作
            user = new User();
            user.setName(request.getParameter("name"));
            user.setPwd(request.getParameter("pwd"));
            user.setGendar(request.getParameter("gendar"));
            user.setUsername(username);
            dao.save(user);
            response.sendRedirect("main.jsp");
        }else{
            //用户名存在，则提示用户
            request.setAttribute("regist_error",
                "用户名已经存在");
            request
                .getRequestDispatcher("regist.jsp")
                .forward(request, response);
        }
    } catch (Exception e) {
        e.printStackTrace();
        throw new ServletException(e);
    }
}
}
}
}

```

## 10) web.xml

```

web.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="2.4"
3     xmlns="http://java.sun.com/xml/ns/j2ee"
4     xmlns:xsi="http://www.w3.org/2001/XMLSchema-in
5     xsi:schemaLocation="http://java.sun.com/xml/ns
6     http://java.sun.com/xml/ns/j2ee/web-app_2_4.xs
7 <servlet>
8     <servlet-name>ActionServlet</servlet-name>
9     <servlet-class>web.ActionServlet</servlet-clas
10 </servlet>
11
12 <servlet-mapping>
13     <servlet-name>ActionServlet</servlet-name>
14     <url-pattern>*.do</url-pattern>
15 </servlet-mapping>
16 </web-app>
17

```

### 11) 新建 main.jsp

用于测试，注册成功后直接重定向到 main.jsp

```

main.jsp
1 <%@page pageEncoding="utf-8"
2 contentType="text/html; charset=utf-8" %>
3 <h1>主功能页面</h1>
4

```

### 12) 提问

在重定向代码后写输出语句会不会执行？

会执行。JVM 将顺序执行代码，除非遇到 return 或 System.exit();

### 13) 测试

访问 [http://localhost:8080/web05\\_login/regist.jsp](http://localhost:8080/web05_login/regist.jsp)

**main**2009/11/20

注册

用户名:aaa

真实姓名:zhen aaa

密码:●●●|

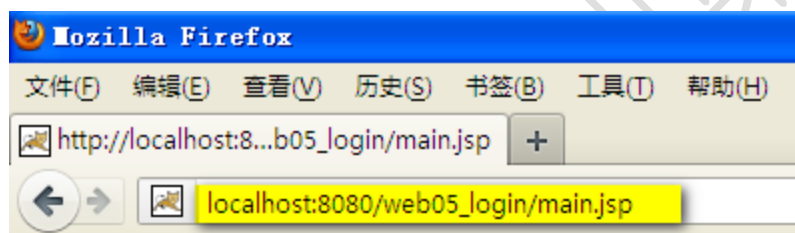
性别:男女

验证码: [换一张](#)

确认

ABC@126.com

a. 成功则跳转到主页面



## 主功能页面

b. 重名，则出现红色提示“用户名已存在”

main

2009/11/20

注册

用户名:

用户已经存在

真实姓名:


密码:

性别:

男 ☒

女 ☐

验证码:

 [换一张](#)

确认

ABC@126.com

#### 14) 拷贝 login.html 到项目下，并修改为 login.jsp

```
<%@page pageEncoding="utf-8"
contentType="text/html; charset=utf-8" %>
<html>
  <head>
    <title>login</title>
    <meta http-equiv="Content-Type"
      content="text/html; charset=UTF-8">
    <link rel="stylesheet" type="text/css"
      href="css/style.css" />
  </head>

  <body>
    <div id="wrap">
      <div id="top_content">
        <div id="header">
          <div id="righthead">
            <p>
              2009/11/20
            <br />
          </p>
        </div>
        <div id="topheader">
```



```

        <h1 id="title">
            <a href="#">main</a>
        </h1>
    </div>
    <div id="navigation">
    </div>
</div>
<div id="content">
    <p id="whereami">
    </p>
    <h1>
        登录
    </h1>
    <form action="login.do"
        method="post">
        <table cellpadding="0" cellspacing="0" border="0"
            class="form_table">
            <tr>
                <td valign="middle" align="right">
                    用户名:

                </td>
                <td valign="middle" align="left">
                    <input type="text" class="inputgri"
                        name="username" />
                    <%
                        String msg =
                            (String)request
                                .getAttribute("login_error");
                    %>
                    <span style="color:red;">
                        <%= (msg == null ? "" : msg) %>
                    </span>
                </td>
            </tr>
            <tr>
                <td valign="middle" align="right">
                    密码:

```

```

        </td>
        <td valign="middle" align="left">
            <input type="password" class="inputgri"
                name="pwd" />
        </td>
    </tr>
</table>
<p>
    <input type="submit" class="button"
        value="确认" />
</p>
</form>

    </div>
</div>
<div id="footer">
    <div id="footer_bg">
        ABC@126.com
    </div>
</div>
</div>
</body>
</html>

```

## 15) 修改 ActionServlet

### 增加注册的代码

```

package web;

import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

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

```

```
import util.DBUtil;
import dao.UserDAO;
import entity.User;

public class ActionServlet extends HttpServlet {

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

        request.setCharacterEncoding("utf-8");
        String uri = request.getRequestURI();
        String path = uri.substring(uri.lastIndexOf("/"),
            uri.lastIndexOf("."));
        if(path.equals("/regist")){
            String username =
                request.getParameter("username");
            UserDAO dao = new UserDAO();
            try {
                User user = dao.findByUsername(username);
                if(user == null){
                    //用户名不存在，插入操作
                    user = new User();
                    user.setName(request.getParameter("name"));
                    user.setPwd(request.getParameter("pwd"));
                    user.setGendar(request.getParameter("gendar"));
                    user.setUsername(username);
                    dao.save(user);
                    response.sendRedirect("main.jsp");
                }else{
                    //用户名存在，则提示用户
                    request.setAttribute("regist_error",
                        "用户名已经存在");
                    request
                        .getRequestDispatcher("regist.jsp")
                        .forward(request, response);
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```

```

    }
    } catch (Exception e) {
        e.printStackTrace();
        throw new ServletException(e);
    }
} else if (path.equals("/login")) {
    String username = request.getParameter("username");
    String pwd = request.getParameter("pwd");
    UserDAO dao = new UserDAO();
    try {
        User user = dao.findByUsername(username);
        if (user != null && user.getPwd().equals(pwd)) {
            // 登录成功
            response.sendRedirect("main.jsp");
        } else {
            // 登录失败
            request.setAttribute("login_error",
                "用户名或者密码错误");
            request.getRequestDispatcher("login.jsp")
                .forward(request, response);
        }
    } catch (Exception e) {
        e.printStackTrace();
        throw new ServletException(e);
    }
}
}
}
}

```

## 16) 重新部署

## 17) 测试

访问 [http://localhost:8080/web05\\_login/login.jsp](http://localhost:8080/web05_login/login.jsp)

main

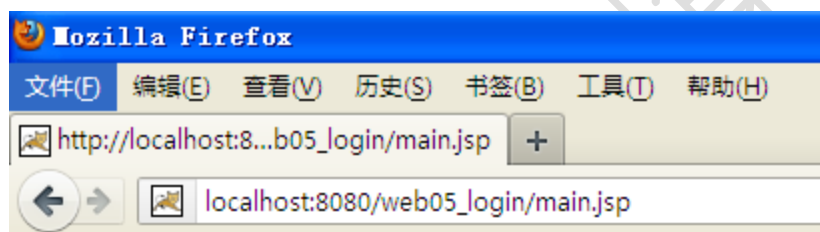
登录

用户名:

密码:

ABC@126.com

a. 成功, 则跳转到 main.jsp



## 主功能页面

b. 失败, 则显示“用户名或密码错误”

main

登录

用户名:

密码:

用户名或密码错误

ABC@126.com

**18) 演示****SQL 注射 ( SQL 注入 )**

登录使用 Statement 会有较大的安全问题：SQL 注入

**a. 修改 ActionServlet**

```
package web;

import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

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

import util.DBUtil;
import dao.UserDAO;
import entity.User;

public class ActionServlet extends HttpServlet {

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

        request.setCharacterEncoding("utf-8");
        String uri = request.getRequestURI();
        String path = uri.substring(uri.lastIndexOf("/"),
            uri.lastIndexOf("."));
        if(path.equals("/regist")){
            String username =
                request.getParameter("username");
            UserDAO dao = new UserDAO();
            try {
                User user = dao.findByUsername(username);
```

```

        if(user == null){
            //用户名不存在，插入操作
            user = new User();
            user.setName(request.getParameter("name"));
            user.setPwd(request.getParameter("pwd"));
            user.setGendar(request.getParameter("gendar"));
            user.setUsername(username);
            dao.save(user);
            response.sendRedirect("main.jsp");
        }else{
            //用户名存在，则提示用户
            request.setAttribute("regist_error",
                "用户名已经存在");
            request
                .getRequestDispatcher("regist.jsp")
                .forward(request, response);
        }
    } catch (Exception e) {
        e.printStackTrace();
        throw new ServletException(e);
    }
} else if(path.equals("/login")){
    String username = request.getParameter("username");
    String pwd = request.getParameter("pwd");
    try {
        Connection conn = DBUtil.getConnection();

        //1 使用 Statement 会有 SQL 注入的安全问题
        Statement stat = conn.createStatement();
        String sql =
            "select * from t_user where username='"
            + username + "' and pwd='" + pwd + "'";
        System.out.println("sql:" + sql);
        ResultSet rst = stat.executeQuery(sql);

        //2 使用 PreparedStatement 能防止 SQL 注入&&提高效率
        // PreparedStatement prep =
        // conn.prepareStatement(

```

```
//          "select * from t_user " +
//          "where username=? and pwd=?");
//      prep.setString(1, username);
//      prep.setString(2, pwd);
//      ResultSet rst = prep.executeQuery();

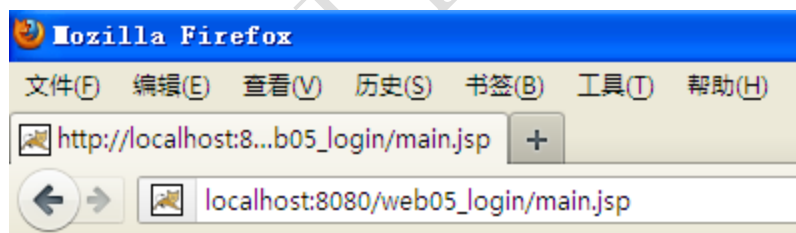
//      if(rst.next()){
//          response.sendRedirect("main.jsp");
//      }else{
//          System.out.println("登录失败");
//      }
//  } catch (Exception e) {
//      e.printStackTrace();
//  }
//  }
//  }
```

**b. 重新部署**

**c. 访问 [http://localhost:8080/web05\\_login/login.jsp](http://localhost:8080/web05_login/login.jsp)**

输入任意用户名，输入密码 **1' or '1' = '1'**

登录成功



## 主功能页面

**d. 打印的 SQL 如下**

```
select * from t_user
where username='xiaoaxiaoaxiao' and pwd='1' or '1' = '1'
```

'1' = '1'永远成立，在没有用户名和密码的情况下登录成功

所以使用拼接 SQL 是很危险的

**e. 解决办法**



- ✓ 在客户端写 JS 脚本屏蔽这样的字符串：**'1' or '1' = '1'**
- ✓ 在服务器端使用 PreparedStatement

### 19) 演示

#### 登录中的 Bug

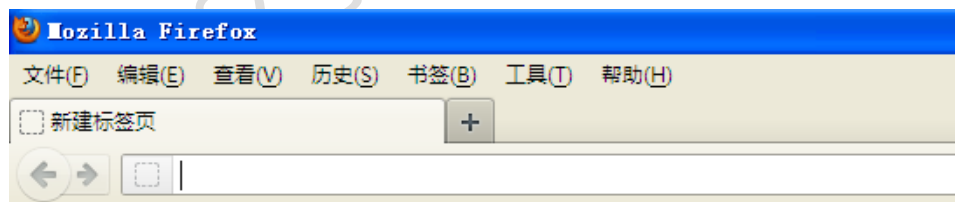
登录成功一次，只要记住地址，下次可以绕过登录

##### a. 用户 aaa 登录成功

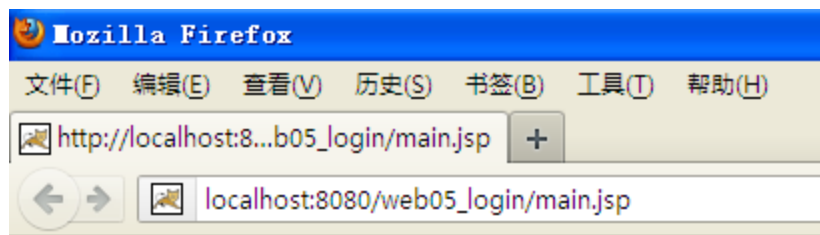


##### b. 复制地址栏里的地址 ( [http://localhost:8080/web05\\_login/main.jsp](http://localhost:8080/web05_login/main.jsp) )

##### c. 打开新页面



##### d. 输入 ( [http://localhost:8080/web05\\_login/main.jsp](http://localhost:8080/web05_login/main.jsp) ) 即可绕过登录



## 主功能页面

因为如上步中出现的问题，我们引入“状态管理”的知识。之后讲解。