

### 1. 搭建环境

#### a. 搭建虚拟主机

- i. 搭建一台www.easymall.com的虚拟主机，并且将这台虚拟主机配置成一个缺省的虚拟主机。

- 1) 实现过程：

- a) 修改[tomcat]/conf目录中的server.xml文件添加内容如下：

```
<Host name="www.easymall.com" appBase="D:/www.easymall.com">
</Host>
```

- b) 在Engine标签身上修改缺省的虚拟主机为www.easymall.com

```
<Engine name="Catalina" defaultHost="www.easymall.com">
```

- c) 配置hosts文件：

C:\Windows\System32\drivers\etc\hosts

配置内容如下：

```
127.0.0.1 www.easymall.com
```

#### b. 发布一个web应用到虚拟主机中

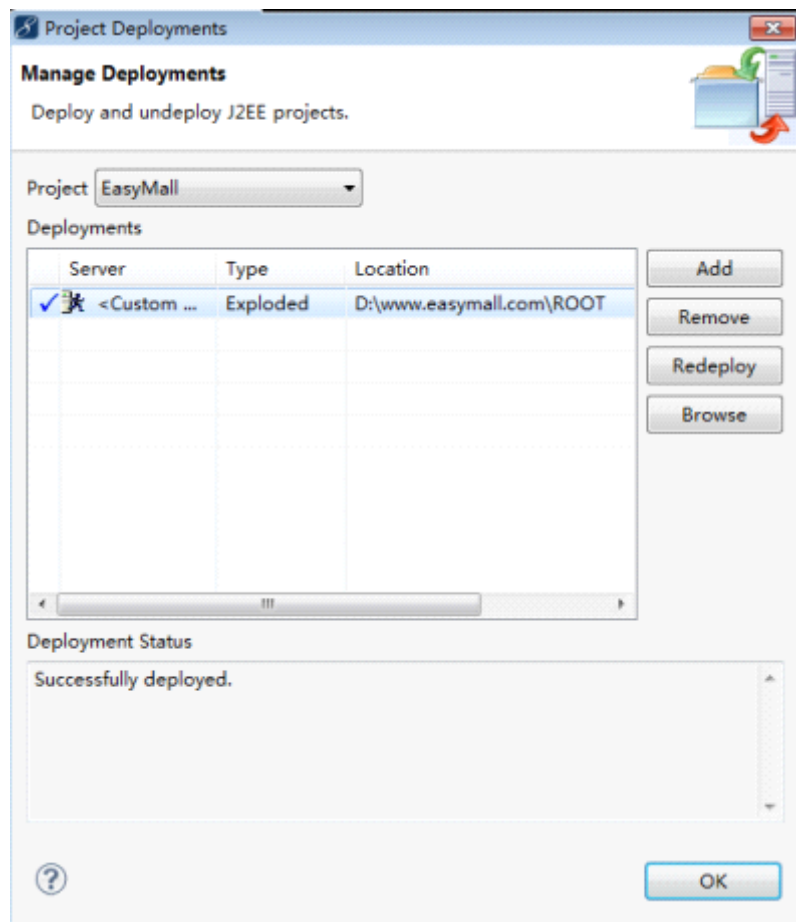
- i. 创建一个名称为EasyMall的web应用，并且将web应用作为缺省web应用部署到www.easymall.com网站中。

- 1) 实现过程：

- a) 创建一个EasyMall的web应用。

- b) 发布EasyMall过程：

- i) 点击发布按钮
- ii) 选择EasyMall项目
- iii) 点击右侧的ADD按钮
- iv) 在弹出的窗口中选择下拉框中最后一项<Custom Location>完成自定义发布。
- v) 选择之后，弹框下方的地址会变为可以修改的状态，将地址修改为www.easymall.com虚拟主机的路径。
- vi) 并将EasyMall修改为ROOT变为缺省web应用。



## 2. 添加页面到EasyMall中

- 在课前资料中寻找index、\_foot、\_head目录复制其中全部内容，粘贴到EasyMall的webRoot下。
- 如果出现提示问是否overwrite, 点击yes to all。
- 选择index.html页面，复制其中全部内容。
- 选择index.jsp页面，删除除了第一行之外的全部内容，并将复制的内容粘贴。
- 修改页面第一行ISO8859-1为UTF-8。
- 同样修改\_foot.jsp和\_head.jsp页面，重复c~e内容。
- 删除后缀为.html的页面。

## 3. 组合页面

- 首页的构成分为三部分，头部，身体和尾部
- 在身体index.jsp页面中引入头部和尾部。

代码实现：在body标签中添加头部和尾部。

```
<BODY>
<!-- 添加首部和尾部 -->
  <!-- 添加首部 -->
  <%-- <%request.getRequestDispatcher("/_head.jsp").include(request, response); %> --%>
  <%@include file="_head.jsp" %>

  <!-- 引入尾部 -->
  <%-- <%request.getRequestDispatcher("/_foot.jsp").include(request, response); %> --%>
  <%@include file="_foot.jsp" %>
</BODY>
```

<%@include file="文件路径"%> ---静态引入。

## 4. 引入regist.jsp页面

- 将课前资料中的regist目录所有内容复制粘贴到WebRoot目录下。

- 修改regist.html页面为regist.jsp 方式和上方修改方式一样。
- 删除regist.html页面。
- 修改\_head.jsp页面中内容：

```
<div id="content">  
    <a href="#"><b>登 录</b></a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
    <a href="%=request.getContextPath() %"/regist.jsp"><b>注 册</b></a>  
</div>
```

### 5. 修改regist.jsp页面代码:

```
<form action="<%=request.getContextPath() %>/RegistServlet" method="POST">
```

## 6. 创建一个注册功能的RegistServlet

在其中添加代码如下：

```
package com.easymall.servlet;

import java.io.IOException;

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

import com.easymall.utils.WebUtils;
/**
 * 注册功能
 * @author Administrator
 *
 */
public class RegistServlet extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        //乱码处理
        //请求乱码和响应乱码处理
        request.setCharacterEncoding("utf-8");
        response.setContentType("text/html;charset=utf-8");

        //获取用户发送的参数信息
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        String password2 = request.getParameter("password2");
        String nickname = request.getParameter("nickname");
        String email = request.getParameter("email");
        String valistr = request.getParameter("valistr");
        //非空校验
        if(WebUtils.isNull(username)){
            //注册页面提示：用户名称不能为空
            request.setAttribute("msg", "用户名称不能为空");
            //请求转发之后当前servlet与regist.jsp页面看到的是同一块域，可以共享数据。
            request.getRequestDispatcher("/regist.jsp").forward(request, response);
            return;
        }
        if(WebUtils.isNull(password)){
            //注册页面提示：用户名称不能为空
            request.setAttribute("msg", "密码不能为空");
            //请求转发之后当前servlet与regist.jsp页面看到的是同一块域，可以共享数据。
            request.getRequestDispatcher("/regist.jsp").forward(request, response);
        }
    }
}
```

```

        return;
    }
    if(WebUtils.isNull(password2)){
        //注册页面提示：用户名称不能为空
        request.setAttribute("msg","确认密码不能为空");
        //请求转发之后当前servlet与regist.jsp页面看到的是同一块域，可以共享数据。
        request.getRequestDispatcher("/regist.jsp").forward(request, response);
        return;
    }
    if(WebUtils.isNull(nickname)){
        //注册页面提示：用户名称不能为空
        request.setAttribute("msg","昵称不能为空");
        //请求转发之后当前servlet与regist.jsp页面看到的是同一块域，可以共享数据。
        request.getRequestDispatcher("/regist.jsp").forward(request, response);
        return;
    }
    if(WebUtils.isNull(email)){
        //注册页面提示：用户名称不能为空
        request.setAttribute("msg","邮箱不能为空");
        //请求转发之后当前servlet与regist.jsp页面看到的是同一块域，可以共享数据。
        request.getRequestDispatcher("/regist.jsp").forward(request, response);
        return;
    }
    if(WebUtils.isNull(valistr)){
        //注册页面提示：用户名称不能为空
        request.setAttribute("msg","验证码不能为空");
        //请求转发之后当前servlet与regist.jsp页面看到的是同一块域，可以共享数据。
        request.getRequestDispatcher("/regist.jsp").forward(request, response);
        return;
    }
    //邮箱格式校验
    //lishuai@tedu.com
    String reg = "^\\w+@\\w+(\\.\\w+)+$";
    if(!email.matches(reg)){
        request.setAttribute("msg","邮箱格式错误");
        request.getRequestDispatcher("/regist.jsp").forward(request, response);
        return;
    }
    //密码一致性校验
    if(!password.equals(password2)){
        request.setAttribute("msg","两次密码不一致");
        request.getRequestDispatcher("/regist.jsp").forward(request, response);
        return;
    }
    //验证码校验
    //TODO:session

    //完成注册----查看右侧分栏【连接数据库判断用户名是否存在】
    //跳转回首页
    response.getWriter().write(
        "<h1 align='center'><font color='red'>" +
        "恭喜注册成功，3秒之后跳转回首页</font></h1>");
    response.setHeader("refresh", "3;url=http://www.easymall.com");
}

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

```

```

        doGet(request, response);
    }
}

```

## 7. 创建WebUtils工具类：

```

package com.easymall.utils;
/**
 * 判断是否为空的工具类
 * @author Administrator
 *
 */
public class WebUtils {
    private WebUtils(){

    }
    public static boolean isNull(String name){
        return "".equals(name.trim()) || name == null;
    }
}

```

## 8. 在regist.jsp页面中添加如下代码：

### a. 获取后台传递来的错误消息提示信息：

table中添加一个新的tr标签：

```

<table>
    <tr>
        <td class="tds" colspan=2 style="color:red;text-align:center">
            <%=request.getAttribute("msg")==null?"": request.getAttribute("msg")%>
        </td>

    </tr>

```

### b. 数据回显操作

在<input name="username" />添加value属性：

```

<input type="text" name="username" value="<%
= request.getParameter("username")==null?"":request.getParameter("username")%>"/>

<input type="text" name="nickname" value="<%
= request.getParameter("nickname")==null?"":request.getParameter("nickname")%>"/>

<input type="text" name="email" value="<%
= request.getParameter("email")==null?"":request.getParameter("email")%>"/>

```

## 9. 创建数据库

执行如下代码，创建数据库及表

```

create database easymall;
use easymall;
create table user(
    id int primary key auto_increment,
    username varchar(100),
    password varchar(100),
    nickname varchar(100),
    email varchar(100)
);

insert into user values(null, 'admin', '123', '炒鸡管理员', 'admin@tedu.cn');
insert into user values(null, '张飞', '123', '管理员', 'admin@tedu.cn');

```

## 10. 连接数据库

- 在当前web应用中添加数据库连接所需jar包，和c3p0的jar包。将两个包放入WEB-INF中的lib目录下。（会自动Build Path）。
- 导入之前使用的c3p0.properties配置文件，在其中修改配置如下：

```
c3p0.driverClass=com.mysql.jdbc.Driver
c3p0.jdbcUrl=jdbc:mysql://localhost:3306/easymall
c3p0.user=root
c3p0.password=root
```

## 11. 添加一个JDBCUtils工具类---获取连接、关闭资源 代码如下：

```
/**
 * 获取连接和关闭资源
 * @author Administrator
 */
public class JDBCUtils {
    private JDBCUtils(){
    }
    private static ComboPooledDataSource source = new ComboPooledDataSource();
    //获取连接
    public static Connection getConnection() throws SQLException{
        return source.getConnection();
    }
    //关闭资源
    public static void close(Connection conn,Statement stat,ResultSet rs){
        if(rs != null){
            try {
                rs.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }finally{
                rs = null;
            }
        }
        if(stat != null){
            try {
                stat.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }finally{
                stat =null;
            }
        }
        if(conn != null){
            try {
                conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }finally{
                conn = null;
            }
        }
    }
}
```

## 12. 页面中添加前台校验

在用户访问一个可提交的页面的时候，可能会频繁点击提交按钮，造成大量无效访问，让服务器的访问产生无意义的压力，应该通过一个前台校验避免这种情况。

- 在regist.jsp的head标签中添加如下代码：

```

<!-- 引入jQuery函数库 -->
<script type="text/javascript" src="js/jquery-1.4.2.js">
</script>
<script type="text/javascript">
    /* 前台校验操作 */
    var formObj = {
        checkForm : function(){
            var canSub = true;

            canSub = this.checkNotNull("username","用户名称不能为空") && canSub;
            canSub = this.checkNotNull("password","密码不能为空") && canSub ;
            canSub = this.checkNotNull("password2","确认密码不能为空")&& canSub;
            canSub = this.checkNotNull("email","邮箱不能为空")&& canSub;
            canSub = this.checkNotNull("nickname","昵称不能为空")&& canSub;
            canSub = this.checkNotNull("valistr","验证码不能为空")&& canSub;
            canSub = this.checkEmail()&& canSub;
            canSub = this.checkPassword()&& canSub;

            return canSub;
        },
        checkPassword : function(){
            //密码一致性校验
            var password = $("input[name='password']").val();
            var password2 = $("input[name='password2']").val();
            if(password!=""&&password2!=""&&password != password2){
                this.setMsg("password2", "两次密码不一致");
                return false;
            }
            return true;
        },
        checkEmail : function(){
            //邮箱格式校验
            var reg = /\w+@\w+(\.\w+)+/;
            var email = $("input[name='email']").val();

            if(email!="" && !reg.test(email)){
                this.setMsg("email", "邮箱格式不正确");
                return false;
            }
            return true;
        },
        //非空校验
        checkNull : function(name,msg){
            var tag = $("input[name='"+name+"']").val();
            //清空操作
            /* $("input[name='"+name+"']").nextAll("span").text(""); */
            this.setMsg(name, "");
            if($.trim(tag) == ""){
                /* $("input[name='"+name+"']").nextAll("span").text(msg); */
                this.setMsg(name, msg);
                return false;
            }
            return true;
        },
        setMsg : function(name,msg){
            //设置信息
            $("input[name='"+name+"']").nextAll("span").text(msg).css("color","red");
        }
    }

</script>

```

b. 在form标签身上添加onsubmit属性:

```

<form action="<%=request.getContextPath()%>/RegistServlet" method="POST"
onsubmit="return formObj.checkForm()">

```

**注意：**前台的js校验只是为了降低无效访问，并不能真正阻拦用户提交错误数据到

数据库中。后台Servlet代码才会真正的验证数据是否有效，阻拦错误数据进入数据库中。

### 13. 添加jQuery离开焦点校验

#### a. 在regist.jsp中添加如下jQuery代码：

```
//文档就绪事件
$(function(){
    //鼠标离开焦点事件
    $("input[name='username']").blur(function(){
        formObj.checkNotNull("username", "用户名不能为空");
    });
    $("input[name='password']").blur(function(){
        formObj.checkNotNull("password", "密码不能为空");
    });
    $("input[name='password2']").blur(function(){
        formObj.checkNotNull("password2", "确认密码不能为空");
        formObj.checkPassword();
    });
    $("input[name='email']").blur(function(){
        formObj.checkNotNull("email", "邮箱不能为空");
        formObj.checkEmail();
    });
    $("input[name='nickname']").blur(function(){
        formObj.checkNotNull("nickname", "昵称不能为空");
    });
    $("input[name='valistr']").blur(function(){
        formObj.checkNotNull("valistr", "验证码不能为空");
    });
});
```



# 连接数据库判断用户名是否存在

2019年5月18日 星期六

14:53

## 1. 代码如下：

```
//完成注册user
//用户名存在校验
Connection conn = null;
PreparedStatement ps = null;
ResultSet rs = null;

try {
    conn = JDBCUtils.getConnection();
    ps = conn.prepareStatement("select * from user where username = ?");
    ps.setString(1, username);
    rs = ps.executeQuery();
    //如果能够查询到结果则证明用户已经存在不允许注册,
    //并且在前台页面做出提示
    if(rs.next()){
        request.setAttribute("msg", "用户名已经存在");
        request.getRequestDispatcher("/regist.jsp").forward(request, response);
        return;
    }else{
        //用户名不存就可以将用户信息插入到数据库当中
        ps = conn.prepareStatement("insert into user values(null,?,?,?,?,?)");
        ps.setString(1, username);
        ps.setString(2, password);
        ps.setString(3, nickname);
        ps.setString(4, email);
        ps.executeUpdate();
    }
} catch (SQLException e) {
    e.printStackTrace();
}finally{
    JDBCUtils.close(conn, ps, rs);
}
```