---

title: "06 案例：用户列表信息展示"
tags: ""

案例：用户列表信息展示

1. 需求：用户增删查改操作

2. 设计：

- **1. 技术选型：Servlet+JSP+MYSQL+JDBCTempleat+Duird+BeanUtils+tomcat**
- **2. 数据库设计：**

```sql
create database userShowTest; -- 创建数据库
use userShowTest;             -- 使用数据库
create table user(            --创建表
    id int primary key auto_increment,
    name varchar(20) not null,
    gender varchar(5),
    age int,
    address varchar(32),
    qq varchar(20),
    email varchar(50)
);
```

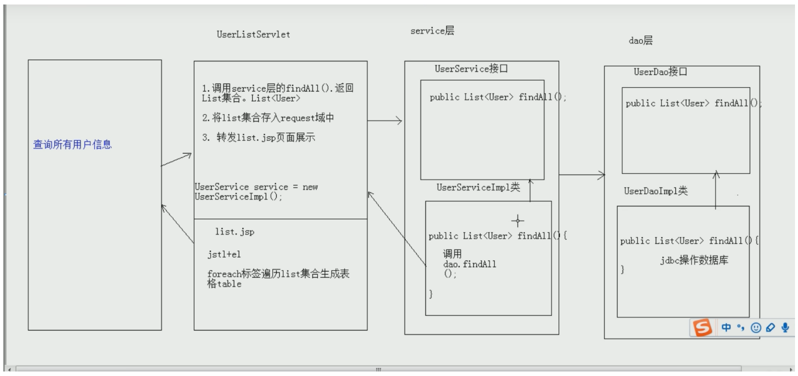
3. 开发

- **1. 环境搭建**

  1. 创建项目环境

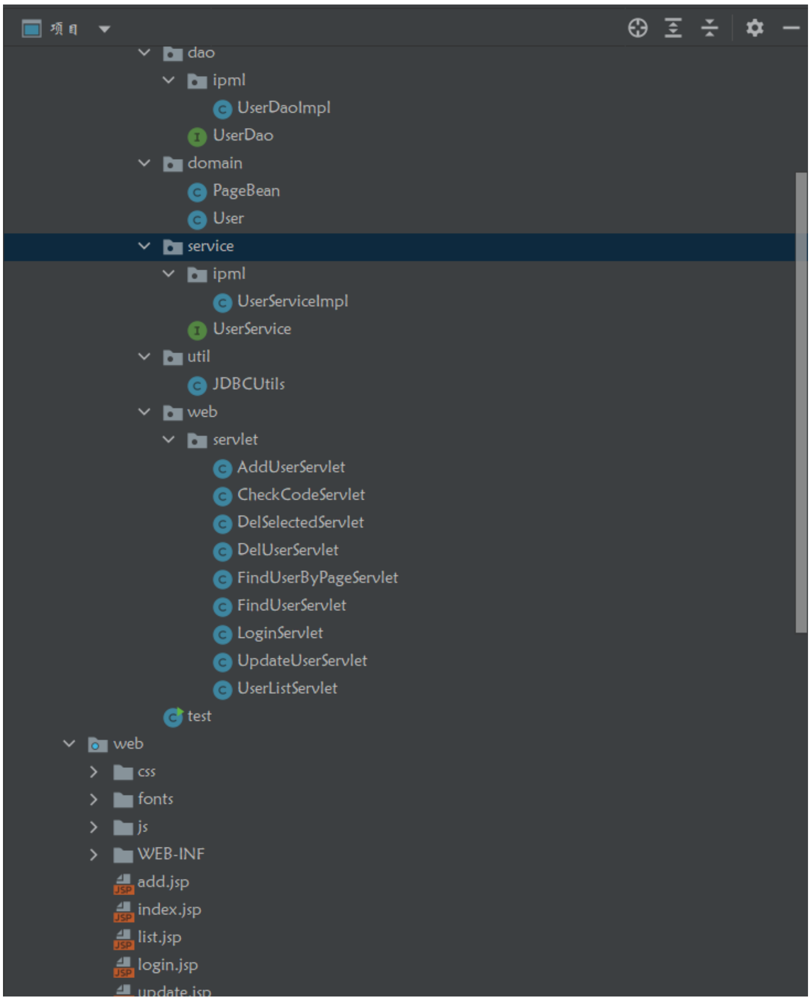
  2. 创建项目，导入需要jar包

- **2. 编码**


UserListServlet
service层
dao层
查询所有用户信息
1.调用service层的findAll().返回List集合。List<User>
2.将list集合存入request域中
3. 转发list.jsp页面展示
UserService service = new UserServiceImpl();
list.jsp
jstl+el
foreach标签遍历list集合生成表格table
UserService接口
public List<User> findAll();
UserServiceImpl类
public List<User> findAll(){
调用
dao.findAll
();
}
UserDao接口
public List<User> findAll();
UserDaoImpl类
public List<User> findAll(){
jdbc操作数据库
}

4. 功能实现

00-项目整体



```
    dao
      ipml
        UserDaoImpl
      UserDao
    domain
      PageBean
      User
    service
      ipml
        UserServiceImpl
      UserService
    util
      JDBCUtils
    web
      servlet
        AddUserServlet
        CheckCodeServlet
        DelSelectedServlet
        DelUserServlet
        FindUserByPageServlet
        FindUserServlet
        LoginServlet
        UpdateUserServlet
        UserListServlet
    test
web
  css
  fonts
  js
  WEB-INF
  add.jsp
  index.jsp
  list.jsp
  login.jsp
  update.jsp
```

01-准备工作 👇

- teat.java-测试数据库连接

```
package cn.case1;
import java.sql.DriverManager;
import java.sql.SQLException;

import java.sql.Connection;
```

```java


public class test {
    public static void main(String[] args) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
//          String url="jdbc:mysql://localhost:3306/day17";
        String url = "jdbc:mysql://localhost:3306/day17?useUnicode=true&characterEncoding=utf8";
        Connection connection= null;
        try {
            connection = DriverManager.getConnection(url,"root","123456");
        } catch (SQLException e) {
            e.printStackTrace();
        }
        try {
            connection.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

resources/druid.properties-数据库配置文件

```properties
driverClassName=com.mysql.cj.jdbc.Driver
url=jdbc:mysql:///day17?useSSL=true&characterEncoding=utf-8&serverTimezone=GMT
username=root
password=123456
initialSize=5
maxActive=20
maxWait=3000
```

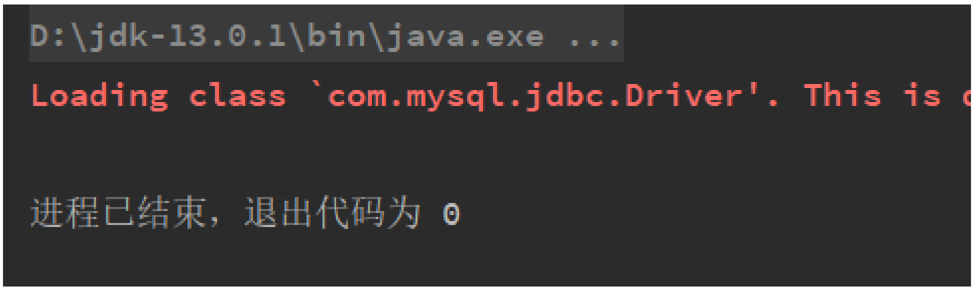
☑ .properties文件需要在项目文件夹下新建一个resources文件夹放入

☑ 新版mysql包使用driverClassName=com.mysql.cj.jdbc.Driver

☑ url地址后面写上数据库名+编码，否则会报错：url=jdbc:mysql:///day17?useSSL=true&characterEncoding=utf-8&serverTimezone=GMT

☑ 用户名密码需要进行确认

- 运行test.java 若显示0则连接成功，显示-1则连接出现问题，对日志文件提示进行排错



```
D:\jdk-13.0.1\bin\java.exe ...
Loading class `com.mysql.jdbc.Driver'. This is

进程已结束，退出代码为 0
```

[]

02-代码部分

- dao>>>impl>>>UserDaoImpl

```java
package cn.case1.dao.ipml;

import cn.case1.dao.UserDao;
import cn.case1.domain.User;
import cn.case1.util.JDBCUtils;
import org.springframework.jdbc.core.BeanPropertyRowMapper;
import org.springframework.jdbc.core.JdbcTemplate;

import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import java.util.Set;

public class UserDaoImpl implements UserDao {
    private JdbcTemplate template=new JdbcTemplate(JDBCUtils.getDataSource
());
    @Override
    public List<User> findAll() {
        //使用JDBC操作数据库
        //1.定义sql
        String sql="select * from user";
        List<User> users = template.query(sql, new BeanPropertyRowMapper<>
(User.class));
        return users;
    }

    @Override
    public User findUsernameAndPassword(String username,String password){
        try {
            String sql="select * from user where username = ? and password
 = ? ";
            User user=template.queryForObject(sql,new BeanPropertyRowMapper
<User>(User.class),
username,password);
            return user;
        }catch (Exception e){
            e.printStackTrace();
            return null;
        }
    }

    @Override
    public void add(User user) {
        //1.定义sql
        String sql="insert into user values(null,?,?,?,?,?,?,null,null)";
        //2.执行sql
        template.update(sql,user.getName(),user.getGender(),user.getAge(),u
ser.getAddress(),
user.getQq(),user.getEmail());
    }

    @Override
    public void delete(int id) {
        //1.定义sql
        String sql="delete from user where id = ?";
        //2.执行sql
        template.update(sql,id);
    }

    @Override
    public User findById(int id) {
        //1.定义sql
        String sql="select * from user where id= ?";
        return template.queryForObject(sql,new BeanPropertyRowMapper<User>
(User.class),id);
    }

    @Override
    public void update(User user) {
       //1.定义sql
       String sql=
"update user set name = ?,gender = ? ,age = ? ,address = ? ,qq = ?,email =
 ? where id = ?";
       template.update(sql,user.getName(),user.getGender(),user.getAge(),us
er.getAddress(),
user.getQq(),user.getEmail(),user.getId());
    }

    @Override
    public List<User> findByPage(int start, int rows, Map<String, String[]>
condition) {
        String sql="select * from user where 1 = 1 ";
        StringBuffer sb=new StringBuffer(sql);
        //2.遍历map
        Set<String> keySet = condition.keySet();
        //定义参数集合
        List<Object> params=new ArrayList<Object>();
        for (String key:keySet) {
            //排除分页条件参数
            if("currentPage".equals(key)||"rows".equals(key)){
                continue;
            }
            //获取value
            String value = condition.get(key)[0];
            //判断value是否有值
            if(value !=null && !"".equals(value)){
                //有值
                sb.append(" and "+key+" like ?");
                params.add("%"+value+"%");//?条件的值
            }
        }
        //添加分页查询
        sb.append(" limit ?,? ");
        //添加分页参数
        params.add(start);
        params.add(rows);

        sql=sb.toString();
        System.out.println(sql);
        System.out.println(params);
        return template.query(sql,new BeanPropertyRowMapper<User>(User.class),params.toArray());
    }

    @Override
    public int findTotalCount(Map<String, String[]> condition) {
        //1.定义模板初始化sql
        String sql="select count(*) from user where 1 = 1 ";
        StringBuffer sb=new StringBuffer(sql);
        //2.遍历map
        Set<String> keySet = condition.keySet();
        //定义参数集合
        List<Object> params=new ArrayList<Object>();
        for (String key:keySet) {
            //排除分页条件参数
            if("currentPage".equals(key)||"rows".equals(key)){
                continue;
            }
            //获取value
            String value = condition.get(key)[0];
            //判断value是否有值
            if(value !=null && !"".equals(value)){
                //有值
                sb.append(" and "+key+" like ?");
                params.add("%"+value+"%");//?条件的值
            }
        }
        System.out.println(sb.toString());
        System.out.println(params);

        return template.queryForObject(sb.toString(),Integer.class,params.toArray());
    }
}
```

- ## dao>>>UserDao

```
package cn.case1.dao;

import cn.case1.domain.User;

import java.util.List;
import java.util.Map;

/*
*用户创作的DAO
* */
public interface UserDao {

    public List<User> findAll();

    User findUsernameAndPassword(String username, String password);

    void add(User user);

    void delete(int id);

    User findById(int id);

    void update(User user);
//查询总记录数
    List<User> findByPage(int start, int rows, Map<String, String[]> condition);
//分页查询

```

```
    int findTotalCount(Map<String, String[]> condition);
}
```

- ## domain>>>PageBean/

```java
package cn.case1.domain;

import java.util.List;

//分页对象
public class PageBean<T> {
    private int totalCount;//总记录数
    private int totalPage;//总页码数
    private List<T> list;//每页的数据
    private int currentPage;//当前页码
    private int rows;//每页显示的记录数

    @Override
    public String toString() {
        return "PageBean{" +
                "totalCount=" + totalCount +
                ", totalPage=" + totalPage +
                ", list=" + list +
                ", currentPage=" + currentPage +
                ", rows=" + rows +
                '}';
    }

    public int getTotalCount() {
        return totalCount;
    }

    public void setTotalCount(int totalCount) {
        this.totalCount = totalCount;
    }

    public int getTotalPage() {
        return totalPage;
    }

    public void setTotalPage(int totalPage) {
        this.totalPage = totalPage;
    }

    public List<T> getList() {
        return list;
    }

    public void setList(List<T> list) {
        this.list = list;
    }

    public int getCurrentPage() {
        return currentPage;
    }

    public void setCurrentPage(int currentPage) {
        this.currentPage = currentPage;
    }

    public int getRows() {
        return rows;
    }

    public void setRows(int rows) {
        this.rows = rows;
    }
}
```

- ## domain>>>User

```
package cn.case1.domain;

public class User {
    private int id;
    private String name;
    private String gender;
    private int age;
    private String address;
    private String qq;
    private String email;
    private String username;
    private String password;

    public String getUsername() {
        return username;
    }

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

    public String getPassword() {
        return password;
```

```java
    }

    public void setPassword(String password) {
        this.password = password;
    }

    @Override
    public String toString() {
        return "User{" +
                "id=" + id +
                ", name='" + name + '\'' +
                ", gender='" + gender + '\'' +
                ", age=" + age +
                ", address='" + address + '\'' +
                ", qq='" + qq + '\'' +
                ", email='" + email + '\'' +
                ", username='" + username + '\'' +
                ", password='" + password + '\'' +
                '}';
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getGender() {
        return gender;
    }

    public void setGender(String gender) {
        this.gender = gender;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public String getQq() {
        return qq;
    }

    public void setQq(String qq) {
        this.qq = qq;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }
}
```

- ### 🗃service>>>🗀Impl>>>📠UserServiceImpl

```java
package cn.case1.service.ipml;

import cn.case1.dao.UserDao;
import cn.case1.dao.ipml.UserDaoImpl;
import cn.case1.domain.PageBean;
import cn.case1.domain.User;
import cn.case1.service.UserService;

import java.util.List;
import java.util.Map;

public class UserServiceImpl implements UserService {
    private UserDao dao=new UserDaoImpl();
    @Override
    public List<User> findAll() {
        //调用Dao完成查询
        return dao.findAll();
    }

    @Override
    public User login(User user){
        return dao. findUsernameAndPassword(user.getUsername(),user.getPass
word());
    }

    @Override
    public void addUser(User user) {
        dao.add(user);
    }

    @Override
    public void deleteUser(String id) {
        dao.delete(Integer.parseInt(id));
    }

    @Override
    public User findUserById(String id) {
        return dao.findById(Integer.parseInt(id));
    }

    @Override
    public void updateUser(User user) {
        dao.update(user);
    }

    @Override
    public void delSelectedUser(String[] ids) {
        if(ids != null && ids.length > 0){
            //1.遍历数组
            for (String id : ids) {
                //2.调用dao删除
                dao.delete(Integer.parseInt(id));
            }
        }
    }

    @Override
    public PageBean<User> findUserByPage(String _currentPage, String _rows,
Map<String, String[]> condition) {
        int currentPage=Integer.parseInt(_currentPage);
        int rows=Integer.parseInt(_rows);
        if (currentPage<=0){
            currentPage=1;
        }
        //1.创建空的PageBean对象
        PageBean<User> pb=new PageBean<User>();
        //2.设置参数
        pb.setCurrentPage(currentPage);
        pb.setRows(rows);
        //3.调用dao查询总记录数
        int totalCount=dao.findTotalCount(condition);
        pb.setTotalCount(totalCount);
        //4.调用dao查询List集合
        //计算一下开始的记录索引
        int start=(currentPage - 1)*rows;
        List<User> list=dao.findByPage(start,rows,condition);
        pb.setList(list);
        //5.计算总页码
        int totalPage=(totalCount%rows )== 0 ? totalCount/rows : (totalCoun
t/rows )+ 1;
        pb.setTotalPage(totalPage);
        return pb;
    }

}
```

- **service>>>UserService**

```
package cn.case1.service;

import cn.case1.domain.PageBean;
import cn.case1.domain.User;

import java.util.List;
import java.util.Map;

/*
* 用户管理的业务接口
* */
public interface UserService {
    /*
    * 查询所有用户信息
    * @return
    * */
    public List<User> findAll();

    /**
     * 登录方法
     * @param user
     * @return
     */
    User login(User user);
    /**
     * 保存对象
     * @param user
     *
     */
    void addUser(User user);
    /**
     * 根据id删除User
     * @param id
     *
     */
    void deleteUser(String id);
    /**
     * 根据id查询User
     * @param id
     *
     * @return
     */
    User findUserById(String id);
    /**
     * 修改用户信息
     * @param user
     * @return
     */
    void updateUser(User user);
    /**
     * 批量删除用户
     * @param ids
     */
    void delSelectedUser(String[] ids);



    /**
     * 分页条件查询
     * @param _currentPage
     * @param _rows
     * @param condition
     * @return
     */
    PageBean<User> findUserByPage(String _currentPage, String _rows, Map<String, String[]> condition);
}
```

- **util>>>JDBCUtils**

```java
package cn.case1.util;

import com.alibaba.druid.pool.DruidDataSourceFactory;

import javax.sql.DataSource;
import java.io.IOException;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.Properties;

/**
 * JDBC工具类 使用Durid连接池
 */
public class JDBCUtils {

    private static DataSource ds ;

    static {

        try {
            //1.加载配置文件
            Properties pro = new Properties();
            //使用ClassLoader加载配置文件，获取字节输入流
            InputStream is = JDBCUtils.class.getClassLoader().getResourceAs
Stream("druid.properties");
            pro.load(is);

            //2.初始化连接池对象
            ds = DruidDataSourceFactory.createDataSource(pro);

        } catch (IOException e) {
            e.printStackTrace();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    /**
     * 获取连接池对象
     */
    public static DataSource getDataSource(){
        return ds;
    }


    /**
```

```
     * 获取连接Connection对象
     */
    public static Connection getConnection() throws SQLException {
        return  ds.getConnection();
    }
}
```

- **web>>>servlet>>>AddUserServlet**

```
package cn.case1.web.servlet;

import cn.case1.domain.User;

import cn.case1.service.ipml.UserServiceImpl;
import org.apache.commons.beanutils.BeanUtils;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.lang.reflect.InvocationTargetException;
import java.util.Map;

@WebServlet("/addUserServlet")
public class AddUserServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        //1.设置编码
        req.setCharacterEncoding("utf-8");
        //2.获取参数
        Map<String, String[]> map = req.getParameterMap();
        //3.封装对象
        User user = new User();
        try {
            BeanUtils.populate(user,map);
        } catch (IllegalAccessException e) {
            e.printStackTrace();
        } catch (InvocationTargetException e) {
            e.printStackTrace();
        }
        //4.调用Service保存
        UserServiceImpl userService = new UserServiceImpl();
        userService.addUser(user);
        //5.跳转到userListServlet
        resp.sendRedirect(req.getContextPath()+"/userListServlet");
    }

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        this.doPost(req, resp);
    }
}
```

- ## web>>>servlet>>>CheckCodeServlet/

```
package cn.case1.web.servlet;

import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.image.BufferedImage;
import java.io.IOException;
import java.util.Random;
import javax.imageio.ImageIO;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * 验证码
 */
@WebServlet("/checkCodeServlet")
public class CheckCodeServlet extends HttpServlet {
        public void doGet(HttpServletRequest request, HttpServletResponse r
esponse)throws ServletException, IOException {

                //服务器通知浏览器不要缓存
                response.setHeader("pragma","no-cache");
                response.setHeader("cache-control","no-cache");
                response.setHeader("expires","0");

                //在内存中创建一个长80，宽30的图片，默认黑色背景
                //参数一：长
                //参数二：宽
                //参数三：颜色
                int width = 80;
                int height = 30;
                BufferedImage image = new BufferedImage(width,height,Buffer
edImage.TYPE_INT_RGB);

                //获取画笔
                Graphics g = image.getGraphics();
                //设置画笔颜色为灰色
                g.setColor(Color.GRAY);
                //填充图片
                g.fillRect(0,0, width,height);

                //产生4个随机验证码，12Ey
                String checkCode = getCheckCode();
                //将验证码放入HttpSession中
                request.getSession().setAttribute("CHECKCODE_SERVER",checkC
ode);

                //设置画笔颜色为黄色
                g.setColor(Color.YELLOW);
                //设置字体的小大
                g.setFont(new Font("黑体",Font.BOLD,24));
                //向图片上写入验证码
                g.drawString(checkCode,15,25);

                //将内存中的图片输出到浏览器
                //参数一：图片对象
                //参数二：图片的格式，如PNG,JPG,GIF
                //参数三：图片输出到哪里去
                ImageIO.write(image,"PNG",response.getOutputStream());
        }
        /**
         * 产生4位随机字符串
         */
        private String getCheckCode() {
                String base = "0123456789ABCDEFGabcdefg";
                int size = base.length();
                Random r = new Random();
                StringBuffer sb = new StringBuffer();
                for(int i=1;i<=4;i++){
                        //产生0到size-1的随机值
                        int index = r.nextInt(size);
                        //在base字符串中获取下标为index的字符
                        char c = base.charAt(index);
                        //将c放入到StringBuffer中去
                        sb.append(c);
                }
                return sb.toString();
        }
        public void doPost(HttpServletRequest request, HttpServletResponse
 response) throws ServletException, IOException {
```

```
                this.doGet(request,response);
        }
}
```

- 🖴web>>>🗀servlet>>>🖫DelSelectedServlet

```
package cn.case1.web.servlet;

import cn.case1.service.UserService;
import cn.case1.service.ipml.UserServiceImpl;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
@WebServlet("/delSelectedServlet")
public class DelSelectedServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        //1.获取所有id
        String[] ids = req.getParameterValues("uid");
        //2.调用service删除
        UserService service = new UserServiceImpl();
        service.delSelectedUser(ids);

        //3.跳转查询所有Servlet
        resp.sendRedirect(req.getContextPath()+"/findUserByPageServlet");
    }

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        this.doPost(req, resp);
    }
}
```

- 🖴web>>>🗀servlet>>>🖫DelUserServlet

```
package cn.case1.web.servlet;

```

```java
import cn.case1.service.UserService;
import cn.case1.service.ipml.UserServiceImpl;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

@WebServlet("/delUserServlet")
public class DelUserServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        //1.获取id
        String id = req.getParameter("id");
        //2.调用service删除
        UserService service = new UserServiceImpl();
        service.deleteUser(id);

        //3.跳转到查询所有Servlet
        resp.sendRedirect(req.getContextPath()+"/findUserByPageServlet");
    }

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        this.doPost(req, resp);
    }
}
```

- ## web>>>servlet>>>FindUserByPageServle

```java
package cn.case1.web.servlet;

import cn.case1.domain.PageBean;
import cn.case1.domain.User;
import cn.case1.service.ipml.UserServiceImpl;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.Map;
```

```java

@WebServlet("/findUserByPageServlet")
public class FindUserByPageServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        req.setCharacterEncoding("utf-8");

        //1.获取参数
        String currentPage = req.getParameter("currentPage");//当前页码
        String rows = req.getParameter("rows");//显示条数
        if(currentPage == null||"".equals(currentPage)){
            currentPage="1";
        }
        if(rows == null||"".equals(rows)){
            rows="5";
        }
        //获取条件查询参数
        Map<String, String[]> condition = req.getParameterMap();

        //2.调用service查询
        UserServiceImpl userService = new UserServiceImpl();
        PageBean<User> pb= userService.findUserByPage(currentPage,rows,condition);
        //3.将PageBean存入request
        req.setAttribute("pb",pb);
        req.setAttribute("condition",condition);//将查询条件也存入
        //4.转发到list.jsp
        req.getRequestDispatcher("/list.jsp").forward(req,resp);
    }


    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        this.doPost(req, resp);
    }
}
```

- **web>>>servlet>>>FindUserServletServlet**

```java
package cn.case1.web.servlet;

import cn.case1.domain.User;
import cn.case1.service.ipml.UserServiceImpl;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

@WebServlet("/findUserServlet")
public class FindUserServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        //1.获取id
        String id=req.getParameter("id");
        //2.调用Service查询
        UserServiceImpl userService = new UserServiceImpl();
        User user=userService.findUserById(id);
        //3.将user存入request
        req.setAttribute("user",user);
        //4.转发到updata.jsp
        req.getRequestDispatcher("/update.jsp").forward(req,resp);
    }

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        this.doPost(req, resp);
    }
}
```

- **web>>>servlet>>>LoginServlet**

```java
package cn.case1.web.servlet;

import cn.case1.domain.User;
import cn.case1.service.UserService;
import cn.case1.service.ipml.UserServiceImpl;
import com.mchange.v2.codegen.bean.BeangenUtils;
import org.apache.commons.beanutils.BeanUtils;

import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.*;
import java.io.IOException;
import java.lang.reflect.InvocationTargetException;
import java.util.Map;

@WebServlet("/loginServlet")
```

```java
public class LoginServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        //1.设置编码
        request.setCharacterEncoding("utf-8");

        //2.获取数据
        //2.1获取用户填写验证码
        String verifycode = request.getParameter("verifycode");
        Map<String, String[]> map = request.getParameterMap();

        //3.验证码校验
        HttpSession session = request.getSession();
        String checkcode_server = (String) session.getAttribute("CHECKCODE_SERVER");
        session.removeAttribute("CHECKCODE_SERVER");//确保验证码一次性
        if (!checkcode_server.equalsIgnoreCase(verifycode)){
            //验证码不正确
            //提示信息
            request.setAttribute("login_msg","验证码错误！");
            //跳转登陆界面
            request.getRequestDispatcher("/login.jsp").forward(request,response);
            return;
        }

        //4.封装User对象
        User user=new User();
        try {
            BeanUtils.populate(user,map);
        } catch (IllegalAccessException e) {
            e.printStackTrace();
        } catch (InvocationTargetException e) {
            e.printStackTrace();
        }

        //5.调用Service查询
        UserService userService=new UserServiceImpl();
        User loginUser=userService.login(user);
        //6.判断是否登陆成功
        if (loginUser != null){
            //登陆成功
            //将用户存入session
            session.setAttribute("user",loginUser);
            //跳转页面
            response.sendRedirect(request.getContextPath()+"/index.jsp");
        }else {
            //登陆失败
            //提示信息
            request.setAttribute("login_msg","用户名或密码错误！");
            //跳转登陆界面
            request.getRequestDispatcher("/login.jsp").forward(request,resp
onse);
        }
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse r
esponse) throws ServletException, IOException {
        this.doGet(request, response);
    }
}
```

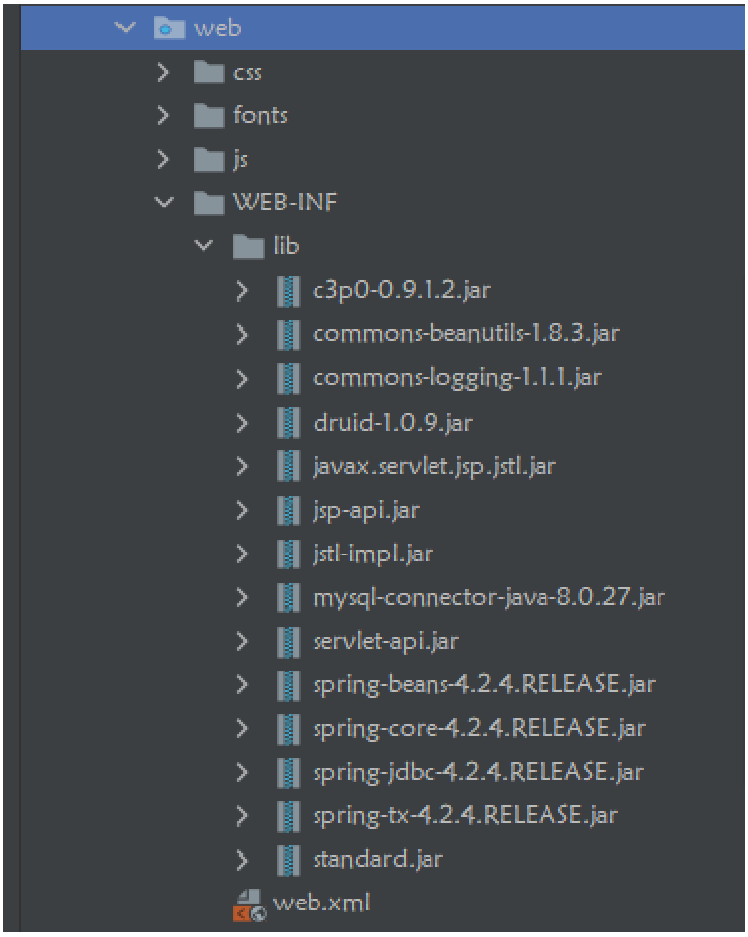
- **web>>>servlet>>>UpdateUserServlet**

```java
package cn.case1.web.servlet;

import cn.case1.domain.User;
import cn.case1.service.ipml.UserServiceImpl;
import org.apache.commons.beanutils.BeanUtils;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.lang.reflect.InvocationTargetException;
import java.util.Map;

@WebServlet("/updateUserServlet")
public class UpdateUserServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        //1.设置编码
        req.setCharacterEncoding("utf-8");
        //2.获取map
        Map<String, String[]> map = req.getParameterMap();
        //3.封装对象
        User user = new User();
        try {
            BeanUtils.populate(user,map);
        } catch (IllegalAccessException e) {
            e.printStackTrace();
        } catch (InvocationTargetException e) {
            e.printStackTrace();
        }
        //4.调用Service修改
        UserServiceImpl userService = new UserServiceImpl();
        userService.updateUser(user);
        //5.转到查询所有Servlet
        resp.sendRedirect(req.getContextPath()+"/userListServlet");
    }

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        this.doPost(req, resp);
    }
}
```

- **web>>>servlet>>>UserListServlet**

```java
package cn.case1.web.servlet;

import cn.case1.domain.User;
import cn.case1.service.UserService;
import cn.case1.service.ipml.UserServiceImpl;

import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.*;
import java.io.IOException;
import java.util.List;

@WebServlet("/userListServlet")
public class UserListServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        //1.调用UserService完成查询
        UserService service=new UserServiceImpl();
        List<User> users = service.findAll();
        //2.将list存入request域
        request.setAttribute("users",users);
        //3.转发到list.jsp
        request.getRequestDispatcher("/list.jsp").forward(request,response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        this.doGet(request, response);
    }
}
```

- **🌐web>>>🗂WEB-INF>>>🗁lib**



```
web
  css
  fonts
  js
  WEB-INF
    lib
      c3p0-0.9.1.2.jar
      commons-beanutils-1.8.3.jar
      commons-logging-1.1.1.jar
      druid-1.0.9.jar
      javax.servlet.jsp.jstl.jar
      jsp-api.jar
      jstl-impl.jar
      mysql-connector-java-8.0.27.jar
      servlet-api.jar
      spring-beans-4.2.4.RELEASE.jar
      spring-core-4.2.4.RELEASE.jar
      spring-jdbc-4.2.4.RELEASE.jar
      spring-tx-4.2.4.RELEASE.jar
      standard.jar
    web.xml
```

- **🌐web>>>📄add.jsp**

```html
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!-- HTML5文档-->
<!DOCTYPE html>
<!-- 网页使用的语言 -->
<html lang="zh-CN">
<head>
    <!-- 指定字符集 -->
    <meta charset="utf-8">
    <!-- 使用Edge最新的浏览器的渲染方式 -->
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <!-- viewport视口：网页可以根据设置的宽度自动进行适配，在浏览器的内部虚拟一个容器，容器的宽度与设备的宽度相同。
    width: 默认宽度与设备的宽度相同
    initial-scale: 初始的缩放比，为1:1 -->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- 上述3个meta标签*必须*放在最前面，任何其他内容都*必须*跟随其后！ -->
    <title>添加用户</title>

    <!-- 1. 导入CSS的全局样式 -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <!-- 2. jQuery导入，建议使用1.9以上的版本 -->
    <script src="js/jquery-2.1.0.min.js"></script>
    <!-- 3. 导入bootstrap的js文件 -->
    <script src="js/bootstrap.min.js"></script>
</head>
<body>
<div class="container">
    <center><h3>添加联系人页面</h3></center>
    <form action="${pageContext.request.contextPath}/addUserServlet" method
="post">
        <div class="form-group">
            <label for="name">姓名：</label>
            <input type="text" class="form-control" id="name" name="name" p
laceholder="请输入姓名">
        </div>

        <div class="form-group">
            <label>性别：</label>
            <input type="radio" name="gender" value="男" checked="checked"/>
男
            <input type="radio" name="gender" value="女"/>女
        </div>

        <div class="form-group">
            <label for="age">年龄：</label>
            <input type="text" class="form-control" id="age" name="age" pla
ceholder="请输入年龄">
        </div>

        <div class="form-group">
            <label for="address">籍贯：</label>
            <select name="address" class="form-control" id="jiguan">
                <option value="广东">广东</option>
                <option value="广西">广西</option>
                <option value="湖南">湖南</option>
            </select>
        </div>

        <div class="form-group">
            <label for="qq">QQ：</label>
            <input type="text" class="form-control" name="qq" placeholder=
"请输入QQ号码"/>
        </div>

        <div class="form-group">
            <label for="email">Email：</label>
            <input type="text" class="form-control" name="email" placeholder="请输入邮箱地址"/>
        </div>

        <div class="form-group" style="text-align: center">
            <input class="btn btn-primary" type="submit" value="提交" />
            <input class="btn btn-default" type="reset" value="重置" />
            <input class="btn btn-default" type="button" value="返回" />
        </div>
    </form>
</div>
</body>
</html>
```

- **🌐web>>>📄index.jsp**

```
<%@ page import="java.sql.DriverManager" %>
<%@ page import="java.sql.ResultSet" %>
<%@ page import="java.sql.Connection" %>
<%@ page import="java.sql.Statement" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html lang="zh-CN">
<head>
  <meta charset="utf-8"/>
  <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
  <meta name="viewport" content="width=device-width, initial-scale=1"/>
  <title>首页</title>

  <!-- 1. 导入CSS的全局样式 -->
  <link href="css/bootstrap.min.css" rel="stylesheet">
  <!-- 2. jQuery导入，建议使用1.9以上的版本 -->
  <script src="js/jquery-2.1.0.min.js"></script>
  <!-- 3. 导入bootstrap的js文件 -->
  <script src="js/bootstrap.min.js"></script>
  <script type="text/javascript">
  </script>
</head>
<body>
<div>${user.name}，欢迎您！</div>
<div align="center">
  <%
    try {
```

```
   Class.forName("com.mysql.cj.jdbc.Driver");  //驱动程序名
   String url = "jdbc:mysql://localhost:3306/day17?useSSL=true&characterEncoding=utf-8&serverTimezone=GMT"; //数据库名
   String username = "root";  //数据库用户名
   String password = "123456";  //数据库用户密码
   Connection conn = DriverManager.getConnection(url, username, password);  //连接状态
   if(conn != null){
    out.print("数据库连接成功！");
    out.print("<br />");
    Statement stmt = null;
    ResultSet rs = null;
    String sql = "select *from user";  //查询语句
    stmt = conn.createStatement();
    rs = stmt.executeQuery(sql);
    out.print("查询结果：");
    out.print("<br/>");
    out.println("name");
    out.print("<br/>");
    while (rs.next()) {
     out.println(rs.getString("name")); //将查询结果输出
     out.print("<br/>");
    }
   }
   else{
    out.print("连接失败！");
    out.print("<br/>");
   }
  }catch (Exception e) {
   out.print("数据库连接异常！");
   out.print("<br/>");
  }
 %>
 <a href="${pageContext.request.contextPath}/findUserByPageServlet" style="text-decoration:none;font-size:33px">查询所有用户信息
 </a>
</div>
</body>
</html>
```

- ### web>>>list.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<!DOCTYPE html>
<!-- 网页使用的语言 -->
```

```
<html lang="zh-CN">
<head>
    <!-- 指定字符集 -->
    <meta charset="utf-8">
    <!-- 使用Edge最新的浏览器的渲染方式 -->
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <!-- viewport视口：网页可以根据设置的宽度自动进行适配，在浏览器的内部虚拟一个容
器，容器的宽度与设备的宽度相同。
    width: 默认宽度与设备的宽度相同
    initial-scale: 初始的缩放比，为1:1 -->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- 上述3个meta标签*必须*放在最前面，任何其他内容都*必须*跟随其后！  -->
    <title>用户信息管理系统</title>

    <!-- 1. 导入CSS的全局样式 -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <!-- 2. jQuery导入，建议使用1.9以上的版本 -->
    <script src="js/jquery-2.1.0.min.js"></script>
    <!-- 3. 导入bootstrap的js文件 -->
    <script src="js/bootstrap.min.js"></script>
    <style type="text/css">
        td, th {
            text-align: center;
        }
    </style>
    <script>
        function deleteUser(id){
            //用户安全提示
            if (confirm("您确定要删除吗？")){
                //访问路径
                location.href="${pageContext.request.contextPath}/delUserSe
rvlet?id="+id;
            }
        }
        window.onload = function(){
            //给删除选中按钮添加单击事件
            document.getElementById("delSelected").onclick = function(){
                if(confirm("您确定要删除选中条目吗？")){
                    var flag = false;
                    //判断是否有选中条目
                    var cbs = document.getElementsByName("uid");
                    for (var i = 0; i < cbs.length; i++) {
                        if(cbs[i].checked){
                            //有一个条目选中了
                            flag = true;
                            break;
                        }
                    }
                    if(flag){//有条目被选中
                        //表单提交
                        document.getElementById("form").submit();
                    }
                }
            }
            //1.获取第一个cb
            document.getElementById("firstCb").onclick = function(){
                //2.获取下边列表中所有的cb
                var cbs = document.getElementsByName("uid");
                //3.遍历
                for (var i = 0; i < cbs.length; i++) {
                    //4.设置这些cbs[i]的checked状态 = firstCb.checked
                    cbs[i].checked = this.checked;
                }
            }
        }
    </script>
</head>
<body>
<div class="container">
    <h3 style="text-align: center">用户信息列表</h3>
    <div style="float: left">
        <form class="form-inline" action="${pageContext.request.contextPat
h}/findUserByPageServlet" method="post">
            <div class="form-group">
                <label for="exampleInputName2">姓名</label>
                <input type="text" name="name" value="${condition.name[0]}"
class="form-control" id="exampleInputName2">
            </div>
            <div class="form-group">
                <label for="exampleInputName3">籍贯</label>
                <input type="text" name="address" value="${condition.adress
[0]}"  class="form-control" id="exampleInputName3" >
            </div>
            <div class="form-group">
                <label for="exampleInputEmail2">邮箱</label>
                <input type="email" name="emmail" value="${condition.email
[0]}"  class="form-control" id="exampleInputEmail2" >
            </div>
            <button type="submit" class="btn btn-default">查询</button>
        </form>
    </div>
    <div style="float: right;margin: 5px">
        <a class="btn btn-primary" href="${pageContext.request.contextPat
h}/add.jsp">添加联系人</a>
        <a class="btn btn-primary" href="javascript:void(0);" id="delSelect
ed">删除选中</a>
```

```
    </div>
    <form action="${pageContext.request.contextPath}/delSelectedServlet" method="post" id="form">
    <table border="1" class="table table-bordered table-hover">

        <tr class="success">
            <th><input type="checkbox" id="firstCb"></th>
            <th>编号</th>
            <th>姓名</th>
            <th>性别</th>
            <th>年龄</th>
            <th>籍贯</th>
            <th>QQ</th>
            <th>邮箱</th>
            <th>操作</th>
        </tr>
        <c:forEach items="${pb.list}" var="user" varStatus="s">
            <tr>
                <th><input type="checkbox" name="uid" value="${user.id}"></th>
                <td>${s.count}</td>
                <td>${user.name}</td>
                <td>${user.gender}</td>
                <td>${user.age}</td>
                <td>${user.address}</td>
                <td>${user.qq}</td>
                <td>${user.email}</td>
                <td><a class="btn btn-default btn-sm" href="${pageContext.request.contextPath}/findUserServlet?id=${user.id}">修改</a>&nbsp;
                    <a class="btn btn-default btn-sm" href="javascript:deleteUser(${user.id});">删除</a></td>
            </tr>
        </c:forEach>

    </table>
    </form>
    <div>
        <nav aria-label="Page navigation">
            <ul class="pagination">
                <c:if test="${pb.currentPage == 1}">
                    <li class="disabled">
                </c:if>
                <c:if test="${pb.currentPage != 1}">
                    <li>
                </c:if>
                <li>
                    <a href="${pageContext.request.contextPath}/findUserByPageServlet?currentPage=${pb.currentPage-1}
```

```
          &rows=5&name=${condition.name[0]}&address=${condition.a
ddress[0]}&email=${condition.email[0]} " aria-label="Previous">
            <span aria-hidden="true">&laquo;</span>
          </a>
        </li>
        <c:forEach begin="1" end="${pb.totalPage}" var="i">
          <c:if test="${pb.currentPage==i}">
            <li class="active"><a href="${pageContext.request.c
ontextPath}/findUserByPageServlet?currentPage=${i}
            &rows=5&name=${condition.name[0]}&address=${conditi
on.address[0]}&email=${condition.email[0]}">${i}</a></li>
          </c:if>
          <c:if test="${pb.currentPage!=i}">
            <li><a href="${pageContext.request.contextPath}/fin
dUserByPageServlet?currentPage=${i}
            &rows=5&name=${condition.name[0]}&address=${conditi
on.address[0]}&email=${condition.email[0]}">${i}</a></li>
          </c:if>
        </c:forEach>
        <li>
          <a href="${pageContext.request.contextPath}/findUserByP
ageServlet?currentPage=${pb.currentPage+1}
          &rows=5&name=${condition.name[0]}&address=${condition.a
ddress[0]}&email=${condition.email[0]}" aria-label="Next">
            <span aria-hidden="true">&raquo;</span>
          </a>
        </li>
        <span style="font-size:25px;margin-left: 5px ">
          共${pb.totalCount}条记录，共${pb.totalPage}页
        </span>
      </ul>
    </nav>
  </div>
</div>
</body>
</html>
```

- ### 🌐web>>>📄login.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html lang="zh-CN">
 <head>
  <meta charset="utf-8"/>
  <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
  <meta name="viewport" content="width=device-width, initial-scale=1"/>
  <title>管理员登录</title>
```

```

    <!-- 1. 导入CSS的全局样式 -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <!-- 2. jQuery导入，建议使用1.9以上的版本 -->
    <script src="js/jquery-2.1.0.min.js"></script>
    <!-- 3. 导入bootstrap的js文件 -->
    <script src="js/bootstrap.min.js"></script>
    <script type="text/javascript">
                //切换验证码
                function refreshCode(){
                        //1.获取验证码图片属性
                        var vcode=document.getElementById("vcode");
                        //2.设置src属性，加上时间戳
                        vcode.src="${pageContext.request.contextPath}/checkCodeServlet?time="+new Date().getTime();
                }
    </script>
  </head>
  <body>
        <div class="container" style="width: 400px;">
                <h3 style="text-align: center;">管理员登录</h3>
                <!-- 出错显示的信息框 -->
                <div class="alert alert-warning alert-dismissible" role="alert">
                        <button type="button" class="close" data-dismiss="alert" >
                                <span>&times;</span></button>
                        <strong>${login_msg}</strong>
                </div>
        <form action="${pageContext.request.contextPath}/loginServlet" method="post">
              <div class="form-group">
                <label for="user">用户名：</label>
                <input type="text" name="username" class="form-control" id="user" placeholder="请输入用户名"/>
              </div>

              <div class="form-group">
                <label for="password">密码：</label>
                <input type="password" name="password" class="form-control" id="password" placeholder="请输入密码"/>
              </div>

              <div class="form-inline">
                <label for="vcode">验证码：</label>
                <input type="text" name="verifycode" class="form-control" id="verifycode" placeholder="请输入验证码" style="width: 120px;"/>
                <a href="javascript:refreshCode()"><img src="${pageContext.
```

```
request.contextPath}/checkCodeServlet"

title="看不清点击刷新" id="vcode"/></a>
              </div>
              <hr/>
              <div class="form-group" style="text-align: center;">
                <input class="btn btn btn-primary" type="submit" value="登
录">
               </div>
                </form>
        </div>
  </body>
</html>
```

- 🌐web>>>📄update.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<!-- 网页使用的语言 -->
<html lang="zh-CN">
    <head>
        <!-- 指定字符集 -->
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1"
>
        <title>修改用户</title>

        <link href="css/bootstrap.min.css" rel="stylesheet">
        <script src="js/jquery-2.1.0.min.js"></script>
        <script src="js/bootstrap.min.js"></script>

    </head>
    <body>
        <div class="container" style="width: 400px;">
        <h3 style="text-align: center;">修改联系人</h3>
            <form action="${pageContext.request.contextPath}/updateUserServ
let" method="post">
            <input type="hidden" name="id" value="${user.id}">
          <div class="form-group">
            <label for="name">姓名：</label>
            <input type="text" class="form-control" id="name" name="name" v
alue="${user.name}" readonly="readonly" placeholder="请输入姓名" />
          </div>

          <div class="form-group">
```

```
            <label>性别：</label>
              <c:if test="${user.gender == '男'}">
                  <input type="radio" name="gender" value="男"  checked/>男
                  <input type="radio" name="gender" value="女"  />女
              </c:if>
              <c:if test="${user.gender == '女'}">
                  <input type="radio" name="gender" value="男"  />男
                  <input type="radio" name="gender" value="女"  checked/>女
              </c:if>
          </div>

          <div class="form-group">
            <label for="age">年龄：</label>
            <input type="text" class="form-control" id="age" value="${user.
age}" name="age" placeholder="请输入年龄" />
          </div>

          <div class="form-group">
            <label for="address">籍贯：</label>
             <select name="address" class="form-control" id="address">
                 <c:if test="${user.address == '广东'}">
                     <option value="广东" selected>广东</option>
                     <option value="广西">广西</option>
                     <option value="湖南">湖南</option>
                 </c:if>
                 <c:if test="${user.address == '广西'}">
                     <option value="广东" >广东</option>
                     <option value="广西" selected>广西</option>
                     <option value="湖南">湖南</option>
                 </c:if>
                 <c:if test="${user.address == '湖南'}">
                     <option value="广东" >广东</option>
                     <option value="广西">广西</option>
                     <option value="湖南" selected>湖南</option>
                 </c:if>
            </select>
          </div>

          <div class="form-group">
            <label for="qq">QQ：</label>
            <input type="text" value="${user.qq}" class="form-control" name
="qq" placeholder="请输入QQ号码"/>
          </div>

          <div class="form-group">
            <label for="email">Email：</label>
            <input type="text" value="${user.email}" class="form-control" n
ame="email" placeholder="请输入邮箱地址"/>
```

```html
          </div>

             <div class="form-group" style="text-align: center">
                <input class="btn btn-primary" type="submit" value="提交" />
                <input class="btn btn-default" type="reset" value="重置" />
                <input class="btn btn-default" type="button" value="返回"/>
             </div>
        </form>
        </div>
    </body>
</html>
```