
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：

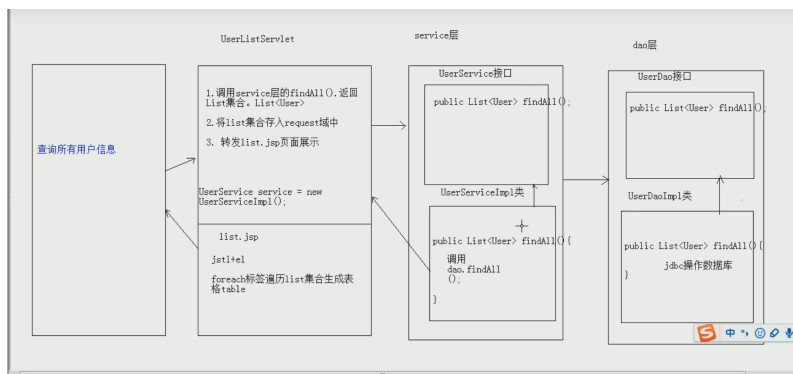
```
1  create database userShowTest; -- 创建数据库
2  use userShowTest;           -- 使用数据库
3  create table user(          --创建表
4      id int primary key auto_increment,
5      name varchar(20) not null,
6      gender varchar(5),
7      age int,
8      address varchar(32),
9      qq varchar(20),
10     email varchar(50)
11 );
```

3. 开发

- 1. 环境搭建

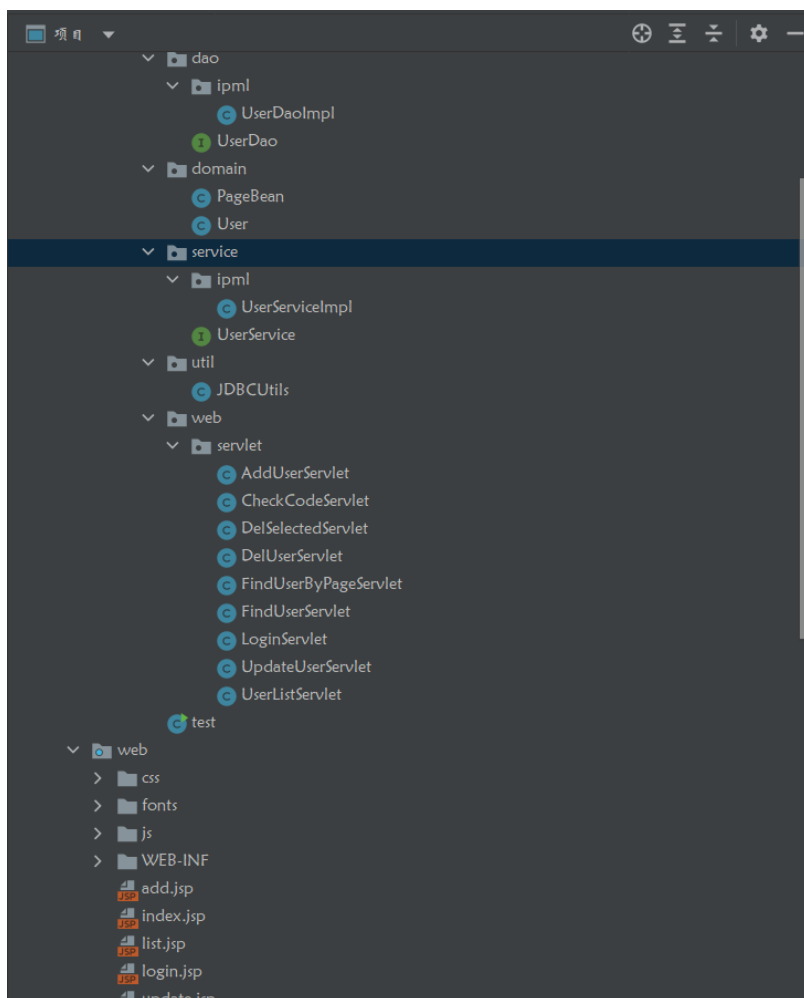
1. 创建项目环境
2. 创建项目，导入需要jar包

- 2. 编码



4. 功能实现

00-项目整体



01-准备工作

- teat.java-测试数据库连接

```

1 package cn.case1;
2 import java.sql.DriverManager;
3 import java.sql.SQLException;
4
5 import java.sql.Connection;
  
```

```

6
7
8 public class test {
9     public static void main(String[] args) {
10         try {
11             Class.forName("com.mysql.cj.jdbc.Driver");
12         } catch (ClassNotFoundException e) {
13             e.printStackTrace();
14         }
15         // String url="jdbc:mysql://localhost:3306/day17";
16         String url = "jdbc:mysql://localhost:3306/day17?useUnicode=true&cha
17 racterEncoding=utf8";
18         Connection connection= null;
19         try {
20             connection = DriverManager.getConnection(url,"root","123456");
21         } catch (SQLException e) {
22             e.printStackTrace();
23         }
24         try {
25             connection.close();
26         } catch (SQLException e) {
27             e.printStackTrace();
28         }
29     }
30 }

```

- **resources/druid.properties-数据库配置文件**

```

1 driverClassName=com.mysql.cj.jdbc.Driver
2 url=jdbc:mysql:///day17?useSSL=true&characterEncoding=utf-8&serverTimezone=
3 GMT
4 username=root
5 password=123456
6 initialSize=5
7 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 c
进程已结束，退出代码为 0
```

[]

02-代码部分

-  **dao** > > >  **impl** > > >  **UserDaoImpl**

```
1 package cn.case1.dao.ipml;
2
3 import cn.case1.dao.UserDao;
4 import cn.case1.domain.User;
5 import cn.case1.util.JDBCUtils;
6 import org.springframework.jdbc.core.BeanPropertyRowMapper;
7 import org.springframework.jdbc.core.JdbcTemplate;
8
9 import java.util.ArrayList;
10 import java.util.List;
11 import java.util.Map;
12 import java.util.Set;
13
14 public class UserDaoImpl implements UserDao {
15     private JdbcTemplate template=new JdbcTemplate(JDBCUtils.getDataSource
16 ());
17     @Override
18     public List<User> findAll() {
19         //使用JDBC操作数据库
20         //1.定义sql
21         String sql="select * from user";
22         List<User> users = template.query(sql, new BeanPropertyRowMapper<>
23 (User.class));
24         return users;
25     }
26
27     @Override
28     public User findUsernameAndPassword(String username,String password){
29         try {
30             String sql="select * from user where username = ? and password
31 = ? ";
32             User user=template.queryForObject(sql,new BeanPropertyRowMapper
```

```

33 <User>(User.class),
34 username,password);
35         return user;
36     }catch (Exception e){
37         e.printStackTrace();
38         return null;
39     }
40 }
41
42 @Override
43 public void add(User user) {
44     //1.定义sql
45     String sql="insert into user values(null,?,?,?,?,?,null,null)";
46     //2.执行sql
47     template.update(sql,user.getName(),user.getGender(),user.getAge(),u
48 ser.getAddress(),
49 user.getQq(),user.getEmail());
50 }
51
52 @Override
53 public void delete(int id) {
54     //1.定义sql
55     String sql="delete from user where id = ?";
56     //2.执行sql
57     template.update(sql,id);
58 }
59
60 @Override
61 public User findById(int id) {
62     //1.定义sql
63     String sql="select * from user where id= ?";
64     return template.queryForObject(sql,new BeanPropertyRowMapper<User>
65 (User.class),id);
66 }
67
68 @Override
69 public void update(User user) {
70     //1.定义sql
71     String sql=
72 "update user set name = ?,gender = ? ,age = ? ,address = ? ,qq = ?,email =
73 ? where id = ?";
74     template.update(sql,user.getName(),user.getGender(),user.getAge(),us
75 er.getAddress(),
76 user.getQq(),user.getEmail(),user.getId());
77 }
78
79 @Override
80 public List<User> findByPage(int start, int rows, Map<String, String[]>

```

```

81 condition) {
82     String sql="select * from user where 1 = 1 ";
83     StringBuffer sb=new StringBuffer(sql);
84     //2.遍历map
85     Set<String> keySet = condition.keySet();
86     //定义参数集合
87     List<Object> params=new ArrayList<Object>();
88     for (String key:keySet) {
89         //排除分页条件参数
90         if("currentPage".equals(key)||"rows".equals(key)){
91             continue;
92         }
93         //获取value
94         String value = condition.get(key)[0];
95         //判断value是否有值
96         if(value !=null && !"".equals(value)){
97             //有值
98             sb.append(" and "+key+" like ?");
99             params.add("%"+value+"%");//?条件的值
100         }
101     }
102     //添加分页查询
103     sb.append(" limit ?,? ");
104     //添加分页参数
105     params.add(start);
106     params.add(rows);
107
108     sql=sb.toString();
109     System.out.println(sql);
110     System.out.println(params);
111     return template.query(sql,new BeanPropertyRowMapper<User>(User.class),params.toArray());
112 }
113
114
115 @Override
116 public int findTotalCount(Map<String, String[]> condition) {
117     //1.定义模板初始化sql
118     String sql="select count(*) from user where 1 = 1 ";
119     StringBuffer sb=new StringBuffer(sql);
120     //2.遍历map
121     Set<String> keySet = condition.keySet();
122     //定义参数集合
123     List<Object> params=new ArrayList<Object>();
124     for (String key:keySet) {
125         //排除分页条件参数
126         if("currentPage".equals(key)||"rows".equals(key)){
127             continue;
128         }

```

```

129         //获取value
130         String value = condition.get(key)[0];
131         //判断value是否有值
132         if(value !=null && !"".equals(value)){
133             //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

```

1 package cn.case1.dao;
2
3 import cn.case1.domain.User;
4
5 import java.util.List;
6 import java.util.Map;
7
8 /*
9  *用户创作的DAO
10  * */
11 public interface UserDao {
12
13     public List<User> findAll();
14
15     User findUsernameAndPassword(String username, String password);
16
17     void add(User user);
18
19     void delete(int id);
20
21     User findById(int id);
22
23     void update(User user);
24 //查询总记录数
25     List<User> findByPage(int start, int rows, Map<String, String[]> condition);
26 //分页查询
27
28

```

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

• domain > > > PageBean/

```
1 package cn.case1.domain;
2
3 import java.util.List;
4
5 //分页对象
6 public class PageBean<T> {
7     private int totalCount;//总记录数
8     private int totalPage;//总页码数
9     private List<T> list;//每页的数据
10    private int currentPage;//当前页码
11    private int rows;//每页显示的记录数
12
13    @Override
14    public String toString() {
15        return "PageBean{" +
16            "totalCount=" + totalCount +
17            ", totalPage=" + totalPage +
18            ", list=" + list +
19            ", currentPage=" + currentPage +
20            ", rows=" + rows +
21            '}';
22    }
23
24    public int getTotalCount() {
25        return totalCount;
26    }
27
28    public void setTotalCount(int totalCount) {
29        this.totalCount = totalCount;
30    }
31
32    public int getTotalPage() {
33        return totalPage;
34    }
35
36    public void setTotalPage(int totalPage) {
37        this.totalPage = totalPage;
38    }
39
40    public List<T> getList() {
41        return list;
42    }
43 }
```



```

43
44     public void setList(List<T> list) {
45         this.list = list;
46     }
47
48     public int getCurrentPage() {
49         return currentPage;
50     }
51
52     public void setCurrentPage(int currentPage) {
53         this.currentPage = currentPage;
54     }
55
56     public int getRows() {
57         return rows;
58     }
59
60     public void setRows(int rows) {
61         this.rows = rows;
62     }
63 }

```

• domain > > > User

```

1 package cn.case1.domain;
2
3 public class User {
4     private int id;
5     private String name;
6     private String gender;
7     private int age;
8     private String address;
9     private String qq;
10    private String email;
11    private String username;
12    private String password;
13
14    public String getUsername() {
15        return username;
16    }
17
18    public void setUsername(String username) {
19        this.username = username;
20    }
21
22    public String getPassword() {
23        return password;

```

```
24     }
25
26     public void setPassword(String password) {
27         this.password = password;
28     }
29
30     @Override
31     public String toString() {
32         return "User{" +
33             "id=" + id +
34             ", name='" + name + '\'' +
35             ", gender='" + gender + '\'' +
36             ", age=" + age +
37             ", address='" + address + '\'' +
38             ", qq='" + qq + '\'' +
39             ", email='" + email + '\'' +
40             ", username='" + username + '\'' +
41             ", password='" + password + '\'' +
42             '}';
43     }
44
45     public int getId() {
46         return id;
47     }
48
49     public void setId(int id) {
50         this.id = id;
51     }
52
53     public String getName() {
54         return name;
55     }
56
57     public void setName(String name) {
58         this.name = name;
59     }
60
61     public String getGender() {
62         return gender;
63     }
64
65     public void setGender(String gender) {
66         this.gender = gender;
67     }
68
69     public int getAge() {
70         return age;
71     }
```

```

72
73     public void setAge(int age) {
74         this.age = age;
75     }
76
77     public String getAddress() {
78         return address;
79     }
80
81     public void setAddress(String address) {
82         this.address = address;
83     }
84
85     public String getQq() {
86         return qq;
87     }
88
89     public void setQq(String qq) {
90         this.qq = qq;
91     }
92
93     public String getEmail() {
94         return email;
95     }
96
97     public void setEmail(String email) {
98         this.email = email;
99     }
100 }

```

• service>>> Impl>>> UserServiceImpl

```

1  package cn.case1.service.ipml;
2
3  import cn.case1.dao.UserDao;
4  import cn.case1.dao.ipml.UserDaoImpl;
5  import cn.case1.domain.PageBean;
6  import cn.case1.domain.User;
7  import cn.case1.service.UserService;
8
9  import java.util.List;
10 import java.util.Map;
11
12 public class UserServiceImpl implements UserService {
13     private UserDao dao=new UserDaoImpl();
14     @Override
15     public List<User> findAll() {

```

```
16         //调用Dao完成查询
17         return dao.findAll();
18     }
19
20     @Override
21     public User login(User user){
22         return dao. findUsernameAndPassword(user.getUsername(),user.getPass
23 word());
24     }
25
26     @Override
27     public void addUser(User user) {
28         dao.add(user);
29     }
30
31     @Override
32     public void deleteUser(String id) {
33         dao.delete(Integer.parseInt(id));
34     }
35
36     @Override
37     public User findUserById(String id) {
38         return dao.findById(Integer.parseInt(id));
39     }
40
41     @Override
42     public void updateUser(User user) {
43         dao.update(user);
44     }
45
46     @Override
47     public void delSelectedUser(String[] ids) {
48         if(ids != null && ids.length > 0){
49             //1.遍历数组
50             for (String id : ids) {
51                 //2.调用dao删除
52                 dao.delete(Integer.parseInt(id));
53             }
54         }
55     }
56
57     @Override
58     public PageBean<User> findUserByPage(String _currentPage, String _rows,
59 Map<String, String[]> condition) {
60         int currentPage=Integer.parseInt(_currentPage);
61         int rows=Integer.parseInt(_rows);
62         if (currentPage<=0){
63             currentPage=1;
```

```

64     }
65     //1.创建空的PageBean对象
66     PageBean<User> pb=new PageBean<User>();
67     //2.设置参数
68     pb.setCurrentPage(currentPage);
69     pb.setRows(rows);
70     //3.调用dao查询总记录数
71     int totalCount=dao.findTotalCount(condition);
72     pb.setTotalCount(totalCount);
73     //4.调用dao查询List集合
74     //计算一下开始的记录索引
75     int start=(currentPage - 1)*rows;
76     List<User> list=dao.findByPage(start,rows,condition);
77     pb.setList(list);
78     //5.计算总页码
79     int totalPages=(totalCount%rows )== 0 ? totalCount/rows : (totalCoun
80 t/rows )+ 1;
81     pb.setTotalPage(totalPage);
82     return pb;
    }
}

```

• service>>> UserService

```

1  package cn.case1.service;
2
3  import cn.case1.domain.PageBean;
4  import cn.case1.domain.User;
5
6  import java.util.List;
7  import java.util.Map;
8
9  /*
10 * 用户管理的业务接口
11 * */
12 public interface UserService {
13     /*
14     * 查询所有用户信息
15     * @return
16     * */
17     public List<User> findAll();
18
19     /**
20     * 登录方法
21     * @param user
22     * @return

```

```

23     */
24     User login(User user);
25     /**
26      * 保存对象
27      * @param user
28      *
29      */
30     void addUser(User user);
31     /**
32      * 根据id删除User
33      * @param id
34      *
35      */
36     void deleteUser(String id);
37     /**
38      * 根据id查询User
39      * @param id
40      *
41      * @return
42      */
43     User findUserById(String id);
44     /**
45      * 修改用户信息
46      * @param user
47      * @return
48      */
49     void updateUser(User user);
50     /**
51      * 批量删除用户
52      * @param ids
53      */
54     void delSelectedUser(String[] ids);
55
56
57
58     /**
59      * 分页条件查询
60      * @param _currentPage
61      * @param _rows
62      * @param condition
63      * @return
64      */
65     PageBean<User> findUserByPage(String _currentPage, String _rows, Map<String, String[]> condition);
66 }

```

```
1 package cn.case1.util;
2
3 import com.alibaba.druid.pool.DruidDataSourceFactory;
4
5 import javax.sql.DataSource;
6 import java.io.IOException;
7 import java.io.InputStream;
8 import java.sql.Connection;
9 import java.sql.SQLException;
10 import java.util.Properties;
11
12 /**
13  * JDBC工具类 使用Druid连接池
14  */
15 public class JDBCUtils {
16
17     private static DataSource ds ;
18
19     static {
20
21         try {
22             //1.加载配置文件
23             Properties pro = new Properties();
24             //使用ClassLoader加载配置文件，获取字节输入流
25             InputStream is = JDBCUtils.class.getClassLoader().getResourceAsStream("druid.properties");
26             pro.load(is);
27
28             //2.初始化连接池对象
29             ds = DruidDataSourceFactory.createDataSource(pro);
30
31         } catch (IOException e) {
32             e.printStackTrace();
33         } catch (Exception e) {
34             e.printStackTrace();
35         }
36     }
37
38
39     /**
40     * 获取连接池对象
41     */
42     public static DataSource getDataSource(){
43         return ds;
44     }
45
46
47     /**
```

```

48     * 获取连接Connection对象
49     */
50     public static Connection getConnection() throws SQLException {
51         return ds.getConnection();
52     }
53 }

```

• web>>> servlet>>> AddUserServlet

```

1  package cn.case1.web.servlet;
2
3  import cn.case1.domain.User;
4
5  import cn.case1.service.ipml.UserServiceImpl;
6  import org.apache.commons.beanutils.BeanUtils;
7
8  import javax.servlet.ServletException;
9  import javax.servlet.annotation.WebServlet;
10 import javax.servlet.http.HttpServlet;
11 import javax.servlet.http.HttpServletRequest;
12 import javax.servlet.http.HttpServletResponse;
13 import java.io.IOException;
14 import java.lang.reflect.InvocationTargetException;
15 import java.util.Map;
16
17 @WebServlet("/addUserServlet")
18 public class AddUserServlet extends HttpServlet {
19     @Override
20     protected void doPost(HttpServletRequest req, HttpServletResponse resp)
21     throws ServletException, IOException {
22         //1.设置编码
23         req.setCharacterEncoding("utf-8");
24         //2.获取参数
25         Map<String, String[]> map = req.getParameterMap();
26         //3.封装对象
27         User user = new User();
28         try {
29             BeanUtils.populate(user, map);
30         } catch (IllegalAccessException e) {
31             e.printStackTrace();
32         } catch (InvocationTargetException e) {
33             e.printStackTrace();
34         }
35         //4.调用Service保存
36         UserServiceImpl userService = new UserServiceImpl();
37         userService.addUser(user);
38         //5.跳转到userListServlet

```



```

39         resp.sendRedirect(req.getContextPath()+"/userListServlet");
40     }
41
42     @Override
43     protected void doGet(HttpServletRequest req, HttpServletResponse resp)
44     throws ServletException, IOException {
45         this.doPost(req, resp);
46     }
47 }

```

• web>>> servlet>>> CheckCodeServlet/

```

1  package cn.case1.web.servlet;
2
3  import java.awt.Color;
4  import java.awt.Font;
5  import java.awt.Graphics;
6  import java.awt.image.BufferedImage;
7  import java.io.IOException;
8  import java.util.Random;
9  import javax.imageio.ImageIO;
10 import javax.servlet.ServletException;
11 import javax.servlet.annotation.WebServlet;
12 import javax.servlet.http.HttpServlet;
13 import javax.servlet.http.HttpServletRequest;
14 import javax.servlet.http.HttpServletResponse;
15
16 /**
17  * 验证码
18  */
19 @WebServlet("/checkCodeServlet")
20 public class CheckCodeServlet extends HttpServlet {
21     public void doGet(HttpServletRequest request, HttpServletResponse r
22     eponse)throws ServletException, IOException {
23
24         //服务器通知浏览器不要缓存
25         response.setHeader("pragma","no-cache");
26         response.setHeader("cache-control","no-cache");
27         response.setHeader("expires","0");
28
29         //在内存中创建一个长80，宽30的图片，默认黑色背景
30         //参数一：长
31         //参数二：宽
32         //参数三：颜色
33         int width = 80;
34         int height = 30;
35         BufferedImage image = new BufferedImage(width,height,Buffer

```

```

36 edImage.TYPE_INT_RGB);
37
38         //获取画笔
39         Graphics g = image.getGraphics();
40         //设置画笔颜色为灰色
41         g.setColor(Color.GRAY);
42         //填充图片
43         g.fillRect(0,0, width,height);
44
45         //产生4个随机验证码, 12Ey
46         String checkCode = getCheckCode();
47         //将验证码放入HttpSession中
48         request.getSession().setAttribute("CHECKCODE_SERVER",checkC
49 ode);
50
51         //设置画笔颜色为黄色
52         g.setColor(Color.YELLOW);
53         //设置字体的小大
54         g.setFont(new Font("黑体",Font.BOLD,24));
55         //向图片上写入验证码
56         g.drawString(checkCode,15,25);
57
58         //将内存中的图片输出到浏览器
59         //参数一: 图片对象
60         //参数二: 图片的格式, 如PNG,JPG,GIF
61         //参数三: 图片输出到哪里去
62         ImageIO.write(image,"PNG",response.getOutputStream());
63     }
64     /**
65      * 产生4位随机字符串
66      */
67     private String getCheckCode() {
68         String base = "0123456789ABCDEFGHabcdefgh";
69         int size = base.length();
70         Random r = new Random();
71         StringBuffer sb = new StringBuffer();
72         for(int i=1;i<=4;i++){
73             //产生0到size-1的随机值
74             int index = r.nextInt(size);
75             //在base字符串中获取下标为index的字符
76             char c = base.charAt(index);
77             //将c放入到StringBuffer中去
78             sb.append(c);
79         }
80         return sb.toString();
81     }
82     public void doPost(HttpServletRequest request, HttpServletResponse
83 response) throws ServletException, IOException {

```

```

84         this.doGet(request, response);
85     }
    }

```

• web>>> servlet>>> DelSelectedServlet

```

1  package cn.case1.web.servlet;
2
3  import cn.case1.service.UserService;
4  import cn.case1.service.ipml.UserServiceImpl;
5
6  import javax.servlet.ServletException;
7  import javax.servlet.annotation.WebServlet;
8  import javax.servlet.http.HttpServlet;
9  import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11 import java.io.IOException;
12 @WebServlet("/delSelectedServlet")
13 public class DelSelectedServlet extends HttpServlet {
14     @Override
15     protected void doPost(HttpServletRequest req, HttpServletResponse resp)
16     throws ServletException, IOException {
17         //1.获取所有id
18         String[] ids = req.getParameterValues("uid");
19         //2.调用service删除
20         UserService service = new UserServiceImpl();
21         service.delSelectedUser(ids);
22
23         //3.跳转查询所有Servlet
24         resp.sendRedirect(req.getContextPath()+"/findUserByPageServlet");
25     }
26
27     @Override
28     protected void doGet(HttpServletRequest req, HttpServletResponse resp)
29     throws ServletException, IOException {
30         this.doPost(req, resp);
31     }
32 }

```

• web>>> servlet>>> DelUserServlet

```

1  package cn.case1.web.servlet;
2

```

```

3 import cn.case1.service.UserService;
4 import cn.case1.service.ipml.UserServiceImpl;
5
6 import javax.servlet.ServletException;
7 import javax.servlet.annotation.WebServlet;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11 import java.io.IOException;
12
13 @WebServlet("/delUserServlet")
14 public class DelUserServlet extends HttpServlet {
15     @Override
16     protected void doPost(HttpServletRequest req, HttpServletResponse resp)
17     throws ServletException, IOException {
18         //1.获取id
19         String id = req.getParameter("id");
20         //2.调用service删除
21         UserService service = new UserServiceImpl();
22         service.deleteUser(id);
23
24         //3.跳转到查询所有Servlet
25         resp.sendRedirect(req.getContextPath()+"/findUserByPageServlet");
26     }
27
28     @Override
29     protected void doGet(HttpServletRequest req, HttpServletResponse resp)
30     throws ServletException, IOException {
31         this.doPost(req, resp);
32     }
33 }

```

• web>>> servlet>>> FindUserByPageServlet

```

1 package cn.case1.web.servlet;
2
3 import cn.case1.domain.PageBean;
4 import cn.case1.domain.User;
5 import cn.case1.service.ipml.UserServiceImpl;
6
7 import javax.servlet.ServletException;
8 import javax.servlet.annotation.WebServlet;
9 import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12 import java.io.IOException;
13 import java.util.Map;

```

```

14
15 @WebServlet("/findUserByPageServlet")
16 public class FindUserByPageServlet extends HttpServlet {
17     @Override
18     protected void doPost(HttpServletRequest req, HttpServletResponse resp)
19     throws ServletException, IOException {
20         req.setCharacterEncoding("utf-8");
21
22         //1.获取参数
23         String currentPage = req.getParameter("currentPage");//当前页码
24         String rows = req.getParameter("rows");//显示条数
25         if(currentPage == null||"".equals(currentPage)){
26             currentPage="1";
27         }
28         if(rows == null||"".equals(rows)){
29             rows="5";
30         }
31         //获取条件查询参数
32         Map<String, String[]> condition = req.getParameterMap();
33
34         //2.调用service查询
35         UserServiceImpl userService = new UserServiceImpl();
36         PageBean<User> pb= userService.findUserByPage(currentPage,rows,cond
37 ition);
38         //3.将PageBean存入request
39         req.setAttribute("pb",pb);
40         req.setAttribute("condition",condition);//将查询条件也存入
41         //4.转发到list.jsp
42         req.getRequestDispatcher("/list.jsp").forward(req,resp);
43     }
44
45
46     @Override
47     protected void doGet(HttpServletRequest req, HttpServletResponse resp)
48     throws ServletException, IOException {
49         this.doPost(req, resp);
50     }
51 }

```

• web>>> servlet>>> FindUserServletServlet

```

1 package cn.case1.web.servlet;
2
3 import cn.case1.domain.User;
4 import cn.case1.service.ipml.UserServiceImpl;
5
6 import javax.servlet.ServletException;

```

```

7 import javax.servlet.annotation.WebServlet;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11 import java.io.IOException;
12
13 @WebServlet("/findUserServlet")
14 public class FindUserServlet extends HttpServlet {
15     @Override
16     protected void doPost(HttpServletRequest req, HttpServletResponse resp)
17     throws ServletException, IOException {
18         //1.获取id
19         String id=req.getParameter("id");
20         //2.调用Service查询
21         UserServiceImpl userService = new UserServiceImpl();
22         User user=userService.findUserById(id);
23         //3.将user存入request
24         req.setAttribute("user",user);
25         //4.转发到update.jsp
26         req.getRequestDispatcher("/update.jsp").forward(req,resp);
27     }
28
29     @Override
30     protected void doGet(HttpServletRequest req, HttpServletResponse resp)
31     throws ServletException, IOException {
32         this.doPost(req, resp);
33     }
34 }

```

• web>>> servlet>>> LoginServlet

```

1 package cn.case1.web.servlet;
2
3 import cn.case1.domain.User;
4 import cn.case1.service.UserService;
5 import cn.case1.service.ipml.UserServiceImpl;
6 import com.mchange.v2.codegen.bean.BeangenUtils;
7 import org.apache.commons.beanutils.BeanUtils;
8
9 import javax.servlet.*;
10 import javax.servlet.http.*;
11 import javax.servlet.annotation.*;
12 import java.io.IOException;
13 import java.lang.reflect.InvocationTargetException;
14 import java.util.Map;
15
16 @WebServlet("/loginServlet")

```

```

17 public class LoginServlet extends HttpServlet {
18     @Override
19     protected void doGet(HttpServletRequest request, HttpServletResponse re
20 sponse) throws ServletException, IOException {
21         //1.设置编码
22         request.setCharacterEncoding("utf-8");
23
24         //2.获取数据
25         //2.1获取用户填写验证码
26         String verifycode = request.getParameter("verifycode");
27         Map<String, String[]> map = request.getParameterMap();
28
29         //3.验证码校验
30         HttpSession session = request.getSession();
31         String checkcode_server = (String) session.getAttribute("CHECKCODE_
32 SERVER");
33         session.removeAttribute("CHECKCODE_SERVER");//确保验证码一次性
34         if (!checkcode_server.equalsIgnoreCase(verifycode)){
35             //验证码不正确
36             //提示信息
37             request.setAttribute("login_msg", "验证码错误! ");
38             //跳转登陆界面
39             request.getRequestDispatcher("/login.jsp").forward(request, resp
40 onse);
41             return;
42         }
43
44         //4.封装User对象
45         User user=new User();
46         try {
47             BeanUtils.populate(user,map);
48         } catch (IllegalAccessException e) {
49             e.printStackTrace();
50         } catch (InvocationTargetException e) {
51             e.printStackTrace();
52         }
53
54         //5.调用Service查询
55         UserService userService=new UserServiceImpl();
56         User loginUser=userService.login(user);
57         //6.判断是否登陆成功
58         if (loginUser != null){
59             //登陆成功
60             //将用户存入session
61             session.setAttribute("user",loginUser);
62             //跳转页面
63             response.sendRedirect(request.getContextPath()+"/index.jsp");
64         }else {

```

```

65         //登陆失败
66         //提示信息
67         request.setAttribute("login_msg", "用户名或密码错误!");
68         //跳转登陆界面
69         request.getRequestDispatcher("/login.jsp").forward(request, resp
70 onse);
71     }
72 }
73
74 @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse r
    esponse) throws ServletException, IOException {
        this.doGet(request, response);
    }
}

```

• web>>> servlet>>> UpdateUserServlet

```

1 package cn.case1.web.servlet;
2
3 import cn.case1.domain.User;
4 import cn.case1.service.ipml.UserServiceIml;
5 import org.apache.commons.beanutils.BeanUtils;
6
7 import javax.servlet.ServletException;
8 import javax.servlet.annotation.WebServlet;
9 import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12 import java.io.IOException;
13 import java.lang.reflect.InvocationTargetException;
14 import java.util.Map;
15
16 @WebServlet("/updateUserServlet")
17 public class UpdateUserServlet extends HttpServlet {
18     @Override
19     protected void doPost(HttpServletRequest req, HttpServletResponse resp)
20 throws ServletException, IOException {
21         //1.设置编码
22         req.setCharacterEncoding("utf-8");
23         //2.获取map
24         Map<String, String[]> map = req.getParameterMap();
25         //3.封装对象
26         User user = new User();
27         try {
28             BeanUtils.populate(user, map);
29         } catch (IllegalAccessException e) {

```



```

30         e.printStackTrace();
31     } catch (InvocationTargetException e) {
32         e.printStackTrace();
33     }
34     //4.调用Service修改
35     UserServiceImpl userService = new UserServiceImpl();
36     userService.updateUser(user);
37     //5.转到查询所有Servlet
38     resp.sendRedirect(req.getContextPath()+"/userListServlet");
39 }
40
41 @Override
42 protected void doGet(HttpServletRequest req, HttpServletResponse resp)
43 throws ServletException, IOException {
44     this.doPost(req, resp);
45 }
46 }

```

• web>>> servlet>>> UserListServlet

```

1 package cn.case1.web.servlet;
2
3 import cn.case1.domain.User;
4 import cn.case1.service.UserService;
5 import cn.case1.service.impl.UserServiceImpl;
6
7 import javax.servlet.*;
8 import javax.servlet.http.*;
9 import javax.servlet.annotation.*;
10 import java.io.IOException;
11 import java.util.List;
12
13 @WebServlet("/userListServlet")
14 public class UserListServlet extends HttpServlet {
15     @Override
16     protected void doGet(HttpServletRequest request, HttpServletResponse re
17 sponse) throws ServletException, IOException {
18         //1.调用UserService完成查询
19         UserService service=new UserServiceImpl();
20         List<User> users = service.findAll();
21         //2.将list存入request域
22         request.setAttribute("users",users);
23         //3.转发到list.jsp
24         request.getRequestDispatcher("/list.jsp").forward(request, respons
25 e);
26     }
27 }

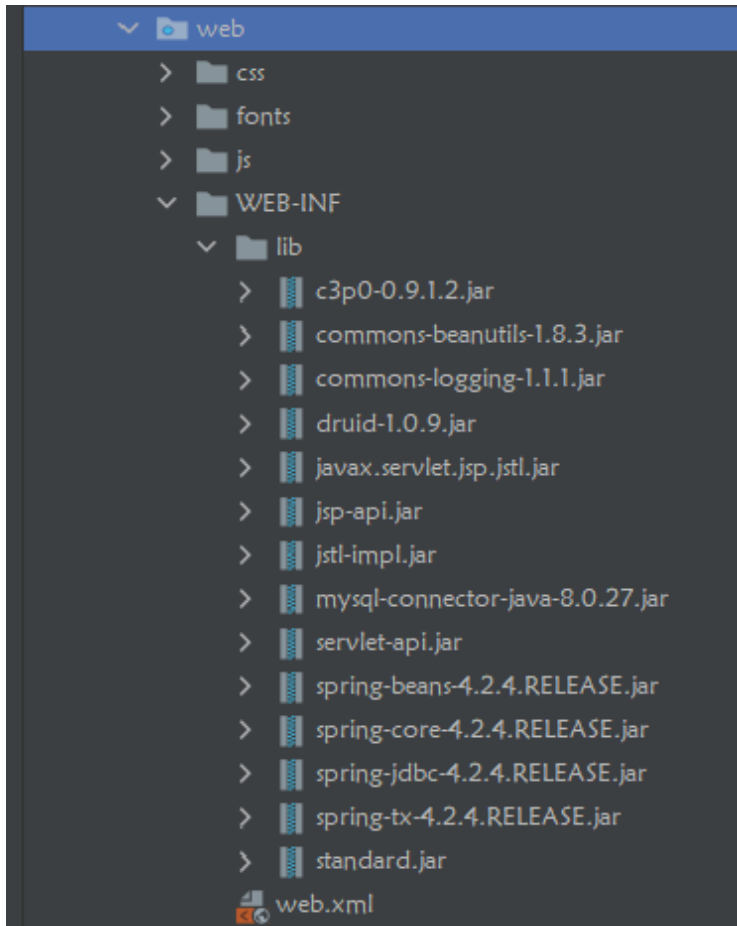
```

```

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

```

• 🌐 web > > > 📁 WEB-INF > > > 📁 lib



• 🌐 web > > > 📄 add.jsp

```

1 <%@ page contentType="text/html; charset=UTF-8" language="java" %>
2 <!-- HTML5文档 -->
3 <!DOCTYPE html>
4 <!-- 网页使用的语言 -->
5 <html lang="zh-CN">
6 <head>
7     <!-- 指定字符集 -->
8     <meta charset="utf-8">
9     <!-- 使用Edge最新的浏览器的渲染方式 -->
10    <meta http-equiv="X-UA-Compatible" content="IE=edge">
11    <!-- 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
12 器, 容器的宽度与设备的宽度相同。
13    width: 默认宽度与设备的宽度相同

```

```

14     initial-scale: 初始的缩放比, 为1:1 -->
15     <meta name="viewport" content="width=device-width, initial-scale=1">
16     <!-- 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! -->
17     <title>添加用户</title>
18
19     <!-- 1. 导入CSS的全局样式 -->
20     <link href="css/bootstrap.min.css" rel="stylesheet">
21     <!-- 2. jQuery导入, 建议使用1.9以上的版本 -->
22     <script src="js/jquery-2.1.0.min.js"></script>
23     <!-- 3. 导入bootstrap的js文件 -->
24     <script src="js/bootstrap.min.js"></script>
25 </head>
26 <body>
27 <div class="container">
28     <center><h3>添加联系人页面</h3></center>
29     <form action="{pageContext.request.contextPath}/addUserServlet" method
30 = "post">
31         <div class="form-group">
32             <label for="name">姓名: </label>
33             <input type="text" class="form-control" id="name" name="name" p
34 laceholder="请输入姓名">
35         </div>
36
37         <div class="form-group">
38             <label>性别: </label>
39             <input type="radio" name="gender" value="男" checked="checked"/>
40 男
41             <input type="radio" name="gender" value="女"/>女
42         </div>
43
44         <div class="form-group">
45             <label for="age">年龄: </label>
46             <input type="text" class="form-control" id="age" name="age" pla
47 ceholder="请输入年龄">
48         </div>
49
50         <div class="form-group">
51             <label for="address">籍贯: </label>
52             <select name="address" class="form-control" id="jiguan">
53                 <option value="广东">广东</option>
54                 <option value="广西">广西</option>
55                 <option value="湖南">湖南</option>
56             </select>
57         </div>
58
59         <div class="form-group">
60             <label for="qq">QQ: </label>
61             <input type="text" class="form-control" name="qq" placeholder=

```

```

62 "请输入QQ号码"/>
63     </div>
64
65     <div class="form-group">
66         <label for="email">Email: </label>
67         <input type="text" class="form-control" name="email" placeholder="请输入邮箱地址"/>
68     </div>
69
70
71     <div class="form-group" style="text-align: center">
72         <input class="btn btn-primary" type="submit" value="提交" />
73         <input class="btn btn-default" type="reset" value="重置" />
74         <input class="btn btn-default" type="button" value="返回" />
75     </div>
76 </form>
77 </div>
78 </body>
79 </html>

```

• web>>> index.jsp

```

1 <%@ page import="java.sql.DriverManager" %>
2 <%@ page import="java.sql.ResultSet" %>
3 <%@ page import="java.sql.Connection" %>
4 <%@ page import="java.sql.Statement" %>
5 <%@ page contentType="text/html; charset=UTF-8" language="java" %>
6 <html lang="zh-CN">
7 <head>
8     <meta charset="utf-8"/>
9     <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
10    <meta name="viewport" content="width=device-width, initial-scale=1"/>
11    <title>首页</title>
12
13    <!-- 1. 导入CSS的全局样式 -->
14    <link href="css/bootstrap.min.css" rel="stylesheet">
15    <!-- 2. jQuery导入, 建议使用1.9以上的版本 -->
16    <script src="js/jquery-2.1.0.min.js"></script>
17    <!-- 3. 导入bootstrap的js文件 -->
18    <script src="js/bootstrap.min.js"></script>
19    <script type="text/javascript">
20    </script>
21 </head>
22 <body>
23 <div>${user.name}, 欢迎您! </div>
24 <div align="center">
25     <%
26     try {

```

```

27     Class.forName("com.mysql.cj.jdbc.Driver"); //驱动程序名
28     String url = "jdbc:mysql://localhost:3306/day17?useSSL=true&character
29 Encoding=utf-8&serverTimezone=GMT"; //数据库名
30     String username = "root"; //数据库用户名
31     String password = "123456"; //数据库用户密码
32     Connection conn = DriverManager.getConnection(url, username, passwor
33 d); //连接状态
34     if(conn != null){
35         out.print("数据库连接成功! ");
36         out.print("<br />");
37         Statement stmt = null;
38         ResultSet rs = null;
39         String sql = "select *from user"; //查询语句
40         stmt = conn.createStatement();
41         rs = stmt.executeQuery(sql);
42         out.print("查询结果: ");
43         out.print("<br/>");
44         out.println("name");
45         out.print("<br/>");
46         while (rs.next()) {
47             out.println(rs.getString("name")); //将查询结果输出
48             out.print("<br/>");
49         }
50     }
51     else{
52         out.print("连接失败! ");
53         out.print("<br/>");
54     }
55 }catch (Exception e) {
56     out.print("数据库连接异常! ");
57     out.print("<br/>");
58 }
59 %>
60 <a href="${pageContext.request.contextPath}/findUserByPageServlet" style=
61 "text-decoration:none;font-size:33px">查询所有用户信息
62 </a>
</div>
</body>
</html>

```

• web>>> list.jsp

```

1 <%@ page contentType="text/html;charset=UTF-8" language="java" %>
2 <%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
3
4 <!DOCTYPE html>
5 <!-- 网页使用的语言 -->

```

```

6 <html lang="zh-CN">
7 <head>
8     <!-- 指定字符集 -->
9     <meta charset="utf-8">
10    <!-- 使用Edge最新的浏览器的渲染方式 -->
11    <meta http-equiv="X-UA-Compatible" content="IE=edge">
12    <!-- 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
13 器，容器的宽度与设备的宽度相同。
14    width：默认宽度与设备的宽度相同
15    initial-scale：初始的缩放比，为1:1 -->
16    <meta name="viewport" content="width=device-width, initial-scale=1">
17    <!-- 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！ -->
18    <title>用户信息管理系统</title>
19
20    <!-- 1. 导入CSS的全局样式 -->
21    <link href="css/bootstrap.min.css" rel="stylesheet">
22    <!-- 2. jQuery导入，建议使用1.9以上的版本 -->
23    <script src="js/jquery-2.1.0.min.js"></script>
24    <!-- 3. 导入bootstrap的js文件 -->
25    <script src="js/bootstrap.min.js"></script>
26    <style type="text/css">
27        td, th {
28            text-align: center;
29        }
30    </style>
31    <script>
32        function deleteUser(id){
33            //用户安全提示
34            if (confirm("您确定要删除吗? ")){
35                //访问路径
36                location.href="${pageContext.request.contextPath}/delUserSe
37rvlet?id="+id;
38            }
39        }
40        window.onload = function(){
41            //给删除选中按钮添加单击事件
42            document.getElementById("delSelected").onclick = function(){
43                if(confirm("您确定要删除选中条目吗? ")){
44                    var flag = false;
45                    //判断是否有选中条目
46                    var cbs = document.getElementsByName("uid");
47                    for (var i = 0; i < cbs.length; i++) {
48                        if(cbs[i].checked){
49                            //有一个条目选中了
50                            flag = true;
51                            break;
52                        }
53                    }

```

```

54         if(flag){//有条目被选中
55             //表单提交
56             document.getElementById("form").submit();
57         }
58     }
59 }
60 //1.获取第一个cb
61 document.getElementById("firstCb").onclick = function(){
62     //2.获取下边列表中所有的cb
63     var cbs = document.getElementsByName("uid");
64     //3.遍历
65     for (var i = 0; i < cbs.length; i++) {
66         //4.设置这些cbs[i]的checked状态 = firstCb.checked
67         cbs[i].checked = this.checked;
68     }
69 }
70 }
71 </script>
72 </head>
73 <body>
74 <div class="container">
75     <h3 style="text-align: center">用户信息列表</h3>
76     <div style="float: left">
77         <form class="form-inline" action="${pageContext.request.contextPat
78 h}/findUserByPageServlet" method="post">
79             <div class="form-group">
80                 <label for="exampleInputName2">姓名</label>
81                 <input type="text" name="name" value="${condition.name[0]}"
82 class="form-control" id="exampleInputName2">
83             </div>
84             <div class="form-group">
85                 <label for="exampleInputName3">籍贯</label>
86                 <input type="text" name="address" value="${condition.adress
87 [0]}" class="form-control" id="exampleInputName3" >
88             </div>
89             <div class="form-group">
90                 <label for="exampleInputEmail2">邮箱</label>
91                 <input type="email" name="emmail" value="${condition.email
92 [0]}" class="form-control" id="exampleInputEmail2" >
93             </div>
94             <button type="submit" class="btn btn-default">查询</button>
95         </form>
96     </div>
97     <div style="float: right;margin: 5px">
98         <a class="btn btn-primary" href="${pageContext.request.contextPat
99 h}/add.jsp">添加联系人</a>
100         <a class="btn btn-primary" href="javascript:void(0);" id="delSelect
101 ed">删除选中</a>

```

```

102     </div>
103     <form action="${pageContext.request.contextPath}/delSelectedServlet" me
104 thod="post" id="form">
105         <table border="1" class="table table-bordered table-hover">
106
107             <tr class="success">
108                 <th><input type="checkbox" id="firstCb"></th>
109                 <th>编号</th>
110                 <th>姓名</th>
111                 <th>性别</th>
112                 <th>年龄</th>
113                 <th>籍贯</th>
114                 <th>QQ</th>
115                 <th>邮箱</th>
116                 <th>操作</th>
117             </tr>
118             <c:forEach items="${pb.list}" var="user" varStatus="s">
119                 <tr>
120                     <th><input type="checkbox" name="uid" value="${user.id}"></
121 th>
122                     <td>${s.count}</td>
123                     <td>${user.name}</td>
124                     <td>${user.gender}</td>
125                     <td>${user.age}</td>
126                     <td>${user.address}</td>
127                     <td>${user.qq}</td>
128                     <td>${user.email}</td>
129                     <td><a class="btn btn-default btn-sm" href="${pageContext.r
130 equest.contextPath}/findUserServlet?id=${user.id}">修改</a>&nbsp;
131                     <a class="btn btn-default btn-sm" href="javascript:dele
132 teUser(${user.id});">删除</a></td>
133                 </tr>
134             </c:forEach>
135
136         </table>
137     </form>
138     <div>
139         <nav aria-label="Page navigation">
140             <ul class="pagination">
141                 <c:if test="${pb.currentPage == 1}">
142                     <li class="disabled">
143                 </c:if>
144                 <c:if test="${pb.currentPage != 1}">
145                     <li>
146                 </c:if>
147                     <li>
148                         <a href="${pageContext.request.contextPath}/findUserByP
149 ageServlet?currentPage=${pb.currentPage-1}

```



```

150         &rows=5&name=${condition.name[0]}&address=${condition.a
151 ddress[0]}&email=${condition.email[0]} " aria-label="Previous">
152         <span aria-hidden="true">&lquo;</span>
153     </a>
154 </li>
155     <c:forEach begin="1" end="${pb.totalPage}" var="i">
156         <c:if test="${pb.currentPage==i}">
157             <li class="active"><a href="${pageContext.request.c
158 ontextPath}/findUserByPageServlet?currentPage=${i}
159         &rows=5&name=${condition.name[0]}&address=${conditi
160 on.address[0]}&email=${condition.email[0]}">${i}</a></li>
161         </c:if>
162         <c:if test="${pb.currentPage!=i}">
163             <li><a href="${pageContext.request.contextPath}/fin
164 dUserByPageServlet?currentPage=${i}
165         &rows=5&name=${condition.name[0]}&address=${conditi
166 on.address[0]}&email=${condition.email[0]}">${i}</a></li>
167         </c:if>
168     </c:forEach>
169     <li>
170         <a href="${pageContext.request.contextPath}/findUserByP
171 ageServlet?currentPage=${pb.currentPage+1}
172         &rows=5&name=${condition.name[0]}&address=${condition.a
173 ddress[0]}&email=${condition.email[0]}" aria-label="Next">
174         <span aria-hidden="true">&raquo;</span>
175     </a>
176 </li>
177     <span style="font-size:25px;margin-left: 5px ">
178         共${pb.totalCount}条记录, 共${pb.totalPage}页
179     </span>
180 </ul>
181 </nav>
182 </div>
183 </div>
184 </body>
185 </html>

```

• web>>> login.jsp

```

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

```

```

9
10 <!-- 1. 导入CSS的全局样式 -->
11 <link href="css/bootstrap.min.css" rel="stylesheet">
12 <!-- 2. jQuery导入, 建议使用1.9以上的版本 -->
13 <script src="js/jquery-2.1.0.min.js"></script>
14 <!-- 3. 导入bootstrap的js文件 -->
15 <script src="js/bootstrap.min.js"></script>
16 <script type="text/javascript">
17     //切换验证码
18     function refreshCode(){
19         //1.获取验证码图片属性
20         var vcode=document.getElementById("vcode");
21         //2.设置src属性, 加上时间戳
22         vcode.src="${pageContext.request.contextPath}/check
23 CodeServlet?time="+new Date().getTime();
24     }
25 </script>
26 </head>
27 <body>
28     <div class="container" style="width: 400px;">
29         <h3 style="text-align: center;">管理员登录</h3>
30         <!-- 出错显示的信息框 -->
31         <div class="alert alert-warning alert-dismissible" role="al
32 ert">
33             <button type="button" class="close" data-dismiss="a
34 lert" >
35                 <span>&times;</span></button>
36                 <strong>${login_msg}</strong>
37             </div>
38         <form action="${pageContext.request.contextPath}/loginServlet" meth
39 od="post">
40             <div class="form-group">
41                 <label for="user">用户名: </label>
42                 <input type="text" name="username" class="form-control" id=
43 "user" placeholder="请输入用户名"/>
44             </div>
45
46             <div class="form-group">
47                 <label for="password">密码: </label>
48                 <input type="password" name="password" class="form-control"
49 id="password" placeholder="请输入密码"/>
50             </div>
51
52             <div class="form-inline">
53                 <label for="vcode">验证码: </label>
54                 <input type="text" name="verifycode" class="form-control" i
55 d="verifycode" placeholder="请输入验证码" style="width: 120px;"/>
56                 <a href="javascript:refreshCode()"></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

```

1 <%@ page contentType="text/html; charset=UTF-8" language="java" %>
2 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
3 <!DOCTYPE html>
4 <!-- 网页使用的语言 -->
5 <html lang="zh-CN">
6     <head>
7         <!-- 指定字符集 -->
8         <meta charset="utf-8">
9         <meta http-equiv="X-UA-Compatible" content="IE=edge">
10        <meta name="viewport" content="width=device-width, initial-scale=1"
11    >
12        <title>修改用户</title>
13
14        <link href="css/bootstrap.min.css" rel="stylesheet">
15        <script src="js/jquery-2.1.0.min.js"></script>
16        <script src="js/bootstrap.min.js"></script>
17
18    </head>
19    <body>
20        <div class="container" style="width: 400px;">
21            <h3 style="text-align: center;">修改联系人</h3>
22            <form action="${pageContext.request.contextPath}/updateUserServ
23let" method="post">
24                <input type="hidden" name="id" value="${user.id}">
25                <div class="form-group">
26                    <label for="name">姓名: </label>
27                    <input type="text" class="form-control" id="name" name="name" v
28alue="${user.name}" readonly="readonly" placeholder="请输入姓名" />
29                </div>
30
31                <div class="form-group">

```

```
32     <label>性别: </label>
33     <c:if test="${user.gender == '男'}">
34         <input type="radio" name="gender" value="男" checked/>男
35         <input type="radio" name="gender" value="女" />女
36     </c:if>
37     <c:if test="${user.gender == '女'}">
38         <input type="radio" name="gender" value="男" />男
39         <input type="radio" name="gender" value="女" checked/>女
40     </c:if>
41 </div>
42
43 <div class="form-group">
44     <label for="age">年龄: </label>
45     <input type="text" class="form-control" id="age" value="${user.
46 age}" name="age" placeholder="请输入年龄" />
47 </div>
48
49 <div class="form-group">
50     <label for="address">籍贯: </label>
51     <select name="address" class="form-control" id="address">
52         <c:if test="${user.address == '广东'}">
53             <option value="广东" selected>广东</option>
54             <option value="广西">广西</option>
55             <option value="湖南">湖南</option>
56         </c:if>
57         <c:if test="${user.address == '广西'}">
58             <option value="广东" >广东</option>
59             <option value="广西" selected>广西</option>
60             <option value="湖南">湖南</option>
61         </c:if>
62         <c:if test="${user.address == '湖南'}">
63             <option value="广东" >广东</option>
64             <option value="广西">广西</option>
65             <option value="湖南" selected>湖南</option>
66         </c:if>
67     </select>
68 </div>
69
70 <div class="form-group">
71     <label for="qq">QQ: </label>
72     <input type="text" value="${user.qq}" class="form-control" name
73 ="qq" placeholder="请输入QQ号码"/>
74 </div>
75
76 <div class="form-group">
77     <label for="email">Email: </label>
78     <input type="text" value="${user.email}" class="form-control" n
79 ame="email" placeholder="请输入邮箱地址"/>
```

```
80         </div>
81
82         <div class="form-group" style="text-align: center">
83             <input class="btn btn-primary" type="submit" value="提交" />
84             <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>
```