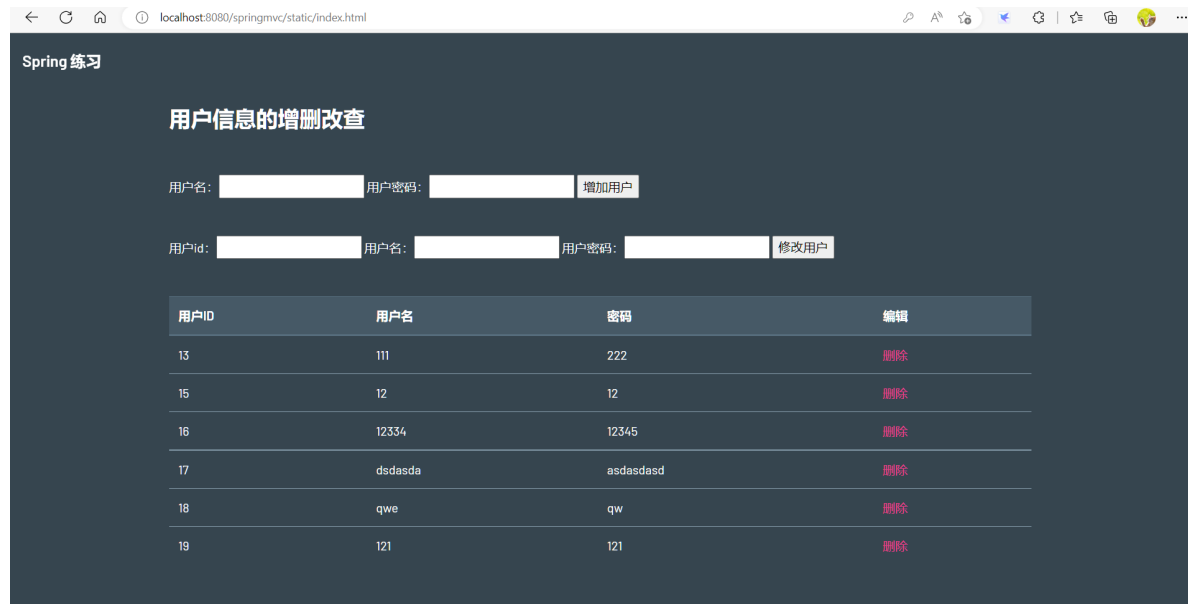


# SSM 框架实战---用户信息增删改查

## 效果展示



## pojo层

User 实体类

```
1 package com.bit.pojo;
2
3 import lombok.AllArgsConstructor;
4 import lombok.Data;
5 import lombok.NoArgsConstructor;
6
7 @Data
8 @AllArgsConstructor
9 @NoArgsConstructor
10 public class User {
11     private int id;
12     private String username;
13     private String password;
14 }
15
```

# mapper层

---

UserMapper接口以及对应的xml文件

UserMapper接口

```
1 package com.bit.mapper;
2
3 import com.bit.pojo.User;
4 import org.apache.ibatis.annotations.Param;
5
6 import java.util.HashMap;
7 import java.util.List;
8
9 public interface UserMapper {
10
11     int insert(User user);
12
13     int delete(@Param("id") int id);
14
15     int update(HashMap<String, Object> map);
16
17     List<User> selectAll();
18
19     User selectById(int id);
20
21 }
```

UserMapper.xml

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE mapper
3     PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
4     "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
5
6 <mapper namespace="com.bit.mapper.UserMapper">
7     <insert id="insert" parameterType="user">
8         insert into users values(#{id},#{username},#{password})
9     </insert>
10
11     <delete id="delete">
12         delete from users where id=#{id}
13     </delete>
14
15     <update id="update">
16         update users set username=#{username},password=#{password} where
17         id=#{id}
18     </update>
```

```

18
19     <select id="selectAll" resultType="user" >
20         select * from users;
21     </select>
22
23     <select id="selectById" resultType="user" >
24         select * from users where id=#{id};
25     </select>
26
27 </mapper>

```

## service层

---

UserService接口

```

1  package com.bit.service;
2
3  import com.bit.pojo.User;
4
5  import java.util.List;
6
7  public interface UserService {
8
9      int addUser(User user);
10
11     int deleteUser(int id);
12
13     int updateUser(int id, User user);
14
15     List<User> selectAllUser();
16
17     User selectOne(int id);
18
19 }
20

```

UserServiceImpl 实现类

```

1  package com.bit.service;
2
3  import com.bit.mapper.UserMapper;
4  import com.bit.pojo.User;
5  import org.springframework.beans.factory.annotation.Autowired;
6  import org.springframework.stereotype.Service;
7

```

```

8  import java.util.HashMap;
9  import java.util.List;
10
11  @Service
12  public class UserServiceImpl implements UserService{
13
14      @Autowired
15      private UserMapper userMapper;
16
17      public int addUser(User user) {
18          return userMapper.insert(user);
19      }
20
21      public int deleteUser(int id) {
22          return userMapper.delete(id);
23      }
24
25      public int updateUser(int id, User user) {
26
27          HashMap<String, Object> map = new HashMap<String, Object>();
28
29          map.put("id", id);
30          map.put("username", user.getUsername());
31          map.put("password", user.getPassword());
32
33          return userMapper.update(map);
34      }
35
36      public List<User> selectAllUser() {
37          return userMapper.selectAll();
38      }
39
40      public User selectOne(int id) {
41          return userMapper.selectById(id);
42      }
43
44      public static void main(String[] args) {
45
46      }
47  }
48

```

## controller层

---

UserController 类

```

1  package com.bit.controller;
2
3  import com.bit.pojo.User;
4  import com.bit.service.UserService;
5  import org.springframework.beans.factory.annotation.Autowired;

```

```
6 import org.springframework.stereotype.Controller;
7 import org.springframework.ui.Model;
8 import org.springframework.web.bind.annotation.RequestMapping;
9 import org.springframework.web.bind.annotation.ResponseBody;
10
11 import javax.ws.WebParam;
12
13 @Controller
14 @RequestMapping("/user")
15 public class UserController {
16
17     @Autowired
18     private UserService userService;
19
20     @RequestMapping("/select")
21     @ResponseBody
22     public Object select(){
23         return userService.selectAllUser();
24     }
25
26     @RequestMapping("/insert")
27     public Object insert(String username,String password){
28         System.out.println("进入到后端的insert接口中!");
29         User user = new User();
30         user.setUsername(username);
31         user.setPassword(password);
32
33         System.out.println(user);
34         userService.addUser(user);
35
36         return "redirect:/static/index.html";
37     }
38
39     @RequestMapping("/update")
40     public Object update(Integer id, String username,String password){
41         System.out.println("进入到后端的update接口中!");
42         User user = new User();
43
44         System.out.println(id);
45         user.setUsername(username);
46         user.setPassword(password);
47
48
49         System.out.println(user);
50         userService.updateUser(id,user);
51
52         return "redirect:/static/index.html";
53     }
54
55     @RequestMapping("/delete")
56     public Object delete(Integer id){
57         System.out.println("进入到后端的delete接口中!");
58
59         System.out.println(id);
60
61         int ret = userService.deleteUser(id);
62         if(ret==1){
63             System.out.println("删除成功!");
```

```

64         }else{
65             System.out.println("删除失败!");
66         }
67
68         return "redirect:/static/index.html";
69     }
70 }
71

```

## 前端主页

使用form表单实现增加、修改，使用Ajax、Dom API 实现查询和修改

```

1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <meta charset="utf-8">
5          <title>练习Spring MVC</title>
6          <meta name="description" content="Charcoal is a free Bootstrap 4
          UI kit build by @attacomsian at Wired Dots." />
7          <meta name="viewport" content="width=device-width, initial-
          scale=1.0">
8          <!--Bootstrap 4-->
9          <link rel="stylesheet" href="css/bootstrap.min.css">
10     </head>
11     <body>
12
13         <nav class="navbar navbar-expand-md navbar-dark fixed-top sticky-
          navigation">
14             <a class="navbar-brand font-weight-bold" href="#">Spring 练习
15             </a>
16             <button class="navbar-toggler" type="button" data-
          toggle="collapse" data-target="#topMenu" aria-
17             controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
          navigation">
18
19                 <span class="navbar-toggler-icon"></span>
20             </button>
21
22             <div class="collapse navbar-collapse" id="topMenu">
23                 </div>
24         </nav>
25
26         <!--components-->
27         <section class="my-5 pt-5">
28             <div class="container">
29
30                 <!-- Tables -->
31                 <div class="row mb-5" id="tables">
32                     <div class="col-sm-12">

```

```

32         <br class="mt-3 mb-5">
33         <h3>用户信息的增删改查</h3>
34
35         <br>
36         <br>
37
38
39         <form
40 action="http://localhost:8080/springmvc/user/insert" >
41         用户名: <input type="text" name="username">
42         用户密码: <input type="password"
43 name="password">
44         <input type="submit" value="增加用户">
45         </form>
46
47         <br>
48         <br>
49         <form
50 action="http://localhost:8080/springmvc/user/update" >
51         用户id: <input type="number" name="id">
52         用户名: <input type="text" name="username">
53         用户密码: <input type="password"
54 name="password">
55         <input type="submit" value="修改用户">
56         </form>
57
58         <br>
59         <br>
60
61         <table class="table">
62         <thead class="thead-dark">
63         <tr>
64         <th>用户ID</th>
65         <th>用户名</th>
66         <th>密码</th>
67         <th>编辑</th>
68         </tr>
69         </thead>
70         <tbody id="usertable">
71         <!-- <tr-->
72         <!-- <td>1</td-->
73         <!-- <td>两数之和</td-->
74         <!-- <td>简单</td-->
75         <!-- <td> <a href="" >删除</a>
76         </td-->
77         <!-- </tr-->
78
79         </tbody>
80         </table>
81     </div>
82 </div>
83 </div>
84 </section>

```

```

85
86
87
88     <script src="https://code.jquery.com/jquery-3.1.1.min.js">
</script>
89     <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.3/umd/popper.mi
n.js"></script>
90     <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
beta.2/js/bootstrap.min.js"></script>
91     <script src="js/app.js"></script>
92     <script>
93         // 在页面加载的时候, 尝试从服务器获取题目列表, 通过 ajax 的方式进行获取
94
95         function getUserMessages() {
96             $.ajax({
97                 url: 'http://localhost:8080/springmvc/user/select',
98                 method: "GET",
99                 success: function (data) {
100                     createUserMessages(data);
101                 }
102             })
103
104         }
105
106         // 使用DOM API 将得到的数据data 写入到页面里面
107         function createUserMessages(data) {
108
109             console.log("进入回调函数内部!")
110             let userTable = document.querySelector("#usertable");
111
112             for (let user of data){
113                 let tr = document.createElement("tr");
114
115                 let tdId = document.createElement("td");
116                 tdId.innerHTML = user.id;
117
118                 let tdName = document.createElement("td");
119                 tdName.innerHTML = user.username;
120
121                 let tdPassword = document.createElement("td");
122                 tdPassword.innerHTML = user.password;
123
124                 let tdA = document.createElement("td");
125                 let a = document.createElement("a");
126                 a.href = "http://localhost:8080/springmvc/user/delete?
id="+user.id;
127                 a.innerHTML="删除";
128                 tdA.appendChild(a);
129
130
131                 tr.appendChild(tdId);
132                 tr.appendChild(tdName);
133                 tr.appendChild(tdPassword);
134                 tr.appendChild(tdA);
135
136                 userTable.appendChild(tr);
137             }

```



```
138
139
140     }
141
142     getUserMessages();
143
144     </script>
145 </body>
146 </html>
147
```