

# 旅游网

---

## 一、注册功能

---

### 1-去注册页面

---

#### a-页面

在header.jsp里添加这个超链接

```
1 | <a href="toRegPage.do">注册</a>
```

#### b-控制层

在UserController中

```
1 | @RequestMapping("/toRegPage.do")
2 | public String toRegPage(){
3 |     return "register";
4 | }
```

### 2-验证输入的合法性

---

#### a-验证规则

**用户名:**不为空且由数字、字母、下划线组成的6-10个字符，不符合要求给提示信息

**密码:**不为空且由数字、字母、下划线组成的6-15个字符，不符合要求给提示信息

**姓名:**非空，不符合要求给提示信息

**出生日期:**非空，不符合要求给提示信息

**邮箱:**不为空且要合法，不符合要求给提示信息

**手机号:**不为空且由11位符合规则的数字组成，不符合要求给提示信息

#### b-验证时机

b1、提交表单时 b2、离开焦点时

#### c-验证代码

```
1 | <script type="text/javascript">
2 |     $(function(){ //页面初始化
3 |         $("#btn1").click(function(){//点击时 验证
```

```

4     $("#registerForm").submit(function(){
5         return checkUsername() && checkEmail() && checkPhone() && checkPwd() &&
checkName() && checkBirth();
6     });
7
8     $("#username").blur(checkUsername); //离开焦点时 验证
9     $("#email").blur(checkEmail);
10    $("#telephone").blur(checkPhone);
11    $("#password").blur(checkPwd);
12    $("#name").blur(checkName);
13    $("#birthday").blur(checkBirth);
14 });
15 //验证用户名
16 function checkUsername(){
17     var username=$("#username").val();
18     var reg=/^\w{6,10}$/; //正则表达式
19     if(!reg.test(username)){
20         $("#username").css("border","1px solid red"); //加红色 实线边框
21         $("#msg").html("用户名由数字字母下划线组成的6-10个字符!");
22         return false;
23     }
24     $("#username").css("border",""); //清空红色 实线边框
25     $("#msg").html("");
26     return true;
27 }
28
29 //验证邮箱
30 function checkEmail(){
31     var email=$("#email").val();
32     var reg=/^[a-z0-9]{3,12}@[a-z0-9]{2,3}\.com$/; //邮箱正则表达式 sunwei@163.com
1111@qq.com
33     if(!reg.test(email)){
34         $("#email").css("border","1px solid red"); //加红色 实线边框
35         $("#msg").html("邮箱不合法!");
36         return false;
37     }
38     $("#email").css("border",""); //清空红色 实线边框
39     $("#msg").html("");
40     return true;
41 }
42
43 //验证手机号
44 function checkPhone(){
45     var telephone=$("#telephone").val();
46     var reg=/^1[356789]\d{9}$/; //手机号 正则表达式
47     if(!reg.test(telephone)){
48         $("#telephone").css("border","1px solid red"); //加红色 实线边框
49         $("#msg").html("请填写正确的手机号!");
50         return false;
51     }
52     $("#telephone").css("border",""); //清空红色 实线边框
53     $("#msg").html("");
54     return true;

```

```

55 }
56 //验证密码
57 function checkPwd(){
58     var password=$("#password").val();
59     var reg=/^\w{6,15}$/; //密码 正则表达式
60     if(!reg.test(password)){
61         $("#password").css("border","1px solid red"); //加红色 实线边框
62         $("#msg").html("密码由6-15位数字、字母、下划线组成!");
63         return false;
64     }
65     $("#password").css("border",""); //清空红色 实线边框
66     $("#msg").html("");
67     return true;
68 }
69
70 //验证姓名
71 function checkName(){
72     var name=$("#name").val();
73     if(name.length==0){
74         $("#name").css("border","1px solid red"); //加红色 实线边框
75         $("#msg").html("姓名为必填项!");
76         return false;
77     }
78     $("#name").css("border",""); //清空红色 实线边框
79     $("#msg").html("");
80     return true;
81 }
82
83 //验证出生日期
84 function checkBirth(){
85     var birthday=$("#birthday").val();
86     if(birthday.length==0){
87         $("#birthday").css("border","1px solid red"); //加红色 实线边框
88         $("#msg").html("出生日期为必填项!");
89         return false;
90     }
91     $("#birthday").css("border",""); //清空 红色 实线边框
92     $("#msg").html("");
93     return true;
94 }
95 </script>

```

### 3-用户名唯一性

用户名通过验证要发送异步ajax请求-从后台验证用户名是否重复

#### a-数据层

```

1 public interface UserMapper {
2     /**
3      * 根据用户名查询该用户
4      */
5     @Select("select * from tab_user where username=#{username}")
6     User selectByUsername(String username);
7 }

```

## b-业务层接口

```

1 boolean checkUsername(String username);

```

## c-业务层实现

在UserServiceImpl中重写以下方法

```

1 @Override
2 public boolean checkUsername(String username) {
3     User user=uMapper.selectByUsername(username);
4     if(user!=null){
5         return false;
6     }else{
7         return true;
8     }
9 }

```

## d、控制层

```

1 /**
2  * 验证用户名是否重复
3  */
4 @RequestMapping("/checkUsername.do")
5 @ResponseBody
6 public String checkUserName(String username){
7     boolean flag=uService.checkUsername(username);
8     if(!flag){
9         return "false"; //用户名重复 返回false
10    }else{
11        return "true";
12    }
13 }

```

## e、表示层

在注册页面的jsp的验证用户名中 添加发送ajax请求部分

```

1 //验证用户名
2 function checkUsername(){
3     var uname=$("#username").val();

```

```

4      var reg=/^\w{6,10}$/; //正则表达式
5      if(!reg.test(uname)){
6          $("#username").css("border","1px solid red"); //加红色 实线边框
7          $("#msg").html("用户名由数字字母下划线组成的6-10个字符!");
8          return false;
9      }
10     $("#username").css("border",""); //清空红色 实线边框
11     $("#msg").html("");
12     //发送ajax请求
13     $.post("checkUname.do",{username:uname},function(res){
14         if(res=="false"){
15             $("#username").css("border","1px solid red"); //加红色 实线边框
16             $("#msg").html("用户名重复不可用!");
17             return false;
18         }else{
19             $("#username").css("border",""); //清空红色 实线边框
20             $("#msg").html("");
21             return true;
22         }
23     })
24     return true;
25 }

```

## 4、保存数据

### a-页面情况

```

1  <form id="registerForm" action="doReg.do" method="post">

```

### b-数据层

```

1  /**
2   * 增加用户
3   */
4  int addUser(User user);

```

### c-映射文件

```

1  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
2  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
3  <mapper namespace="com.ssm.dao.UserMapper">
4  <!--增加用户|注册 -->
5  <insert id="addUser" useGeneratedKeys="true" keyProperty="uid">
6      insert into
7      tab_user(username,password,name,birthday,sex,telephone,email,status,code)
8      values(#{username},#{password},#{name},#{birthday},#{sex},#{telephone},#{email},#
9      {status},#{code});
10 </insert>
11 </mapper>

```

## d-业务层接口

```
1 | boolean registerUser(User user);
```

## e-业务层实现

在UserServiceImpl中的方法实现

```
1 | @Transactional
2 | public boolean registerUser(User user) {
3 |     user.setCode(UuidUtil.getUuid()); //生成随机的激活码
4 |     user.setStatus("N"); //设置用户激活状态为N-未激活
5 |     int num=uMapper.addUser(user);
6 |     if(num>0){
7 |         return true;
8 |     }
9 |     return false;
10 | }
```

## f-控制层

```
1 | /**
2 |  * 注册功能
3 |  */
4 | @RequestMapping("/doReg.do")
5 | public ModelAndView doReg(User user){
6 |     ModelAndView mv =new ModelAndView();
7 |     boolean flag=uService.registerUser(user);
8 |     if(flag){ //如果注册成功-去注册成功页面
9 |         mv.setViewName("register_ok");
10 |     }else{//注册失败-重新回到注册页面
11 |         mv.setViewName("register");
12 |         mv.addObject("msg","注册失败");
13 |     }
14 |     return mv;
15 | }
```

## j-注册失败页面提示

注册页面注册按钮下面

```
1 | <!--提示信息 -->
2 | <tr>
3 |     <td class="td_left">
4 |     </td>
5 |     <td class="td_right">
6 |         <span id="msg" style="color: red;">${msg==null?'':msg}</span>
7 |     </td>
8 | </tr>
```

## 二、邮件处理

### A-邮箱设置

a1.申请邮箱

a2.开启授权码

a3.在MailUtils中设置自己的邮箱账号和密码(授权码)

#### POP3/IMAP/SMTP/Exchange/CardDAV/CalDAV服务

开启服务：	POP3/SMTP服务 (如何使用 Foxmail 等软件收发邮件?)	已开启   关闭
	IMAP/SMTP服务 (什么是 IMAP，它又是如何设置?)	已关闭   开启
	Exchange服务 (什么是Exchange，它又是如何设置?)	已关闭   开启
	CardDAV/CalDAV服务 (什么是CardDAV/CalDAV，它又是如何设置?)	已关闭   开启
	(POP3/IMAP/SMTP/CardDAV/CalDAV服务均支持SSL连接。如何设置?)	

邮件工具类：MailUtils，调用sendMail方法完成邮件发送

### B-发送邮件

在UserServiceImpl中注册方法里添加邮件发送代码

```
1  @Transactional
2  public boolean registerUser(User user) {
3      user.setCode(UUIDUtil.getUuid()); //生成随机的激活码
4      user.setStatus("N"); //设置用户激活状态为N-未激活
5      int num=uMapper.addUser(user);
6      if(num>0){
7          //用户的信息已经保存到数据库中
8          String context="<a href='http://localhost:8080/50-
TravelProject/activeMail.do?code="+user.getCode()+"'>点击激活旅游网</a>";
9          MailUtils.sendMail(user.getEmail(),context,"软件园旅游网激活邮件");
10         return true;
11     }
12     return false;
13 }
```

### C-用户激活

#### c1-数据层

```

1  /**
2   * 根据激活码查询用户
3   */
4  @Select("select * from tab_user where code=#{code}")
5  User selectByCode(String code);
6
7  /**
8   * 更新用户信息
9   */
10 int updateUser(User user);

```

## c2-映射文件

在UserMapper.xml中添加更新用户的动态SQL

```

1  <!--更新用户信息 -->
2  <update id="updateUser" parameterType="user">
3      update tab_user
4      <set>
5          <if test="username!=null and username!=''">username=#{username},</if>
6          <if test="password!=null and password!=''">password=#{password},</if>
7          <if test="name!=null and name!=''">name=#{name},</if>
8          <if test="birthday!=null and birthday!=''">birthday=#{birthday},</if>
9          <if test="sex!=null and sex!=''">sex=#{sex},</if>
10         <if test="telephone!=null and telephone!=''">telephone=#{telephone},</if>
11         <if test="email!=null and email!=''">email=#{email},</if>
12         <if test="status!=null and status!=''">status=#{status},</if>
13         <if test="code!=null and code!=''">code=#{code}</if>
14     </set>
15     where uid=#{uid}
16 </update>

```

## c3-业务接口

```

1  /**
2   * 激活用户
3   */
4  boolean activeUser(String code);

```

## c4-业务实现



```

1  @Transactional
2  public boolean activeUser(String code) {
3      User user=uMapper.selectByCode(code);
4      if(user!=null){//有注册-去激活-返回true
5          user.setStatus("Y"); //设置 激活状态为Y
6          if(uMapper.updateUser(user)>0){ //更新激活状态为Y
7              return true;
8          }else{
9              return false;
10         }
11     }
12     return false;
13 }

```

## c5-控制层

```

1  /**
2   * 用户激活
3   * 激活成功-去登录url;激活成功-给其提示(激活失败, 请联系管理员)
4   * @throws Exception
5   */
6  @RequestMapping("/activeMail.do")
7  public void activeUser(String code, HttpServletResponse res) throws Exception{
8      boolean flag=uService.activeUser(code);
9      String msg=null; //提示信息
10     if(flag){
11         msg="<h3><a href='toLogin.do'>激活成功点击登录</a></h3>";
12     }else{
13         msg="<h3>激活失败请联系管理员!</h3>";
14     }
15     res.setContentType("text/html;charset=utf-8");
16     res.getWriter().write(msg);
17 }

```

## D-激活成功去登录

```

1  在UserController中
2  /**
3   * 去登录页面
4   */
5  @RequestMapping("/toLogin.do")
6  public String toLoginPage(){
7      return "login";
8  }

```

# 三、用户登录

## A-页面验证

## A1-验证规则

### 用户名:

用户名由6-15个数字、字母、下划线组成，不符合规则-边框变为红实线并给其提示 否则清空边框及信息提示

### 密码:

密码由6-15个数字、字母、下划线组成，不符合规则-边框变为红实线并给其提示 否则清空边框及信息提示

## A2-验证时机

1、用户点击提交时验证 2、离开焦点时验证

## A3-验证实现

在login.jsp中

```
1  <script type="text/javascript">
2      $(function(){
3          //1、用户点击时验证
4          $("#sub_login").click(function(){
5              return checkUname() && checkPwd();
6          });
7          //2、离开焦点时验证
8          $("#username").blur(checkUname);
9          $("#password").blur(checkPwd);
10     });
11
12     //验证用户名
13     function checkUname(){//用户名由6-15个数字、字母、下划线组成，不符合规则-边框变为红实线并给
其提示 否则清空边框及信息提示
14         //1、获取值
15         var uname=$("#username").val();
16         //2、正则表达式验证
17         var reg_uname=/^\w{6,15}$/;
18         //3、判断并提示
19         if(!reg_uname.test(uname)){
20             $("#username").css("border","1px solid red");
21             $("#errorMsg").html("用户名由6-15个数字、字母、下划线组成!");
22             return false;
23         }
24         $("#username").css("border",""); //清空边框和提示信息
25         $("#errorMsg").html("");
26         return true;
27     }
28
29     //验证密码
30     function checkPwd(){//密码由6-15个数字、字母、下划线组成，不符合规则-边框变为红实线并给其提
示 否则清空边框及信息提示
31         //1、获取密码值
32         var pwd=$("#password").val();
33         //2、正则表达式验证
34         var reg_pwd=/^\w{6,15}$/;
```

```

35         //3、判断并提示
36         if(!reg_pwd.test(pwd)){
37             $("#password").css("border","1px solid red");
38             $("#errorMsg").html("密码由6-15个数字、字母、下划线组成!");
39             return false;
40         }
41         $("#password").css("border",""); //清空边框和提示信息
42         $("#errorMsg").html("");
43         return true;
44     }
45     </script>

```

## B-登录实现

### b1-数据接口

```

1  /**
2   * 登录方法-在UserMapper接口中
3   */
4  @Select("select * from tab_user where username=#{username} and password=#{password}")
5  User login(@Param("username")String username,@Param("password")String password);

```

### b2-业务接口

```

1  /**
2   * 登录方法-在UserService接口中
3   */
4  User login(String username,String password);

```

### b3-业务实现

在UserServiceImpl中

```

1  @Override
2  public User login(String username, String password) {
3      return uMapper.login(username, password);
4  }

```

### b4-控制层

```

1  /**
2   * 用户登录
3   */
4  @RequestMapping("/doLogin.do")
5  public ModelAndView login(String username,String password,HttpServletRequest req){
6      ModelAndView mv=new ModelAndView();
7      //1、用户名和密码不正确-用户不存在-去登录页面并给其提示-用户名或密码不正确
8      User user=uService.login(username, password);
9      if(user==null){

```

```

10     mv.addObject("msg", "用户名或密码不正确!");
11     mv.setViewName("forward:toLogin.do");
12 }
13 //2、用户名和密码正确-用户存在-用户的状态不为Y-请去邮箱激活用户
14 if(user!=null && !"Y".equals(user.getStatus())){
15     mv.addObject("msg", "登录失败！请进行邮箱激活!");
16     mv.setViewName("forward:toLogin.do");
17 }
18 //3、用户名和密码正确-用户存在-用户的状态为Y
19 if(user!=null && "Y".equals(user.getStatus())){
20     //登录成功后将用户信息保存到session对象当中
21     req.getSession().setAttribute("user",user);
22     mv.setViewName("forward:indexPage.do");
23 }
24 return mv;
25 }

```

## C-登录信息错误提示

```

1 <!--登录错误提示信息-->
2 <div id="errorMsg" class="alert alert-danger" >${msg==null?'':msg}</div>

```

## D-首页用户信息显示

在header.jsp中

```

1 <span>欢迎, ${sessionScope.user.name==null?"游客":sessionScope.user.name}</span>

```

# 四、用户退出

## A-退出本质

退出本质是清空Session

## A-页面连接

在header.jsp页面中

```

1 <a href="toLogout.do">退出</a>

```

## B-控制层代码

```
1  /**
2   * 用户退出功能
3   */
4   @RequestMapping("/toLogout.do")
5   public ModelAndView logout(HttpServletRequest req){
6       req.getSession().invalidate(); //清空session
7       return new ModelAndView("forward:indexPage.do");
8   }
```

## 五、导航条分类

### A-数据接口

```
1  /**
2   * 分类Mapper
3   */
4   public interface CategoryMapper {
5
6       @Select("select cid,cname from tab_category")
7       List<Category> selectAllCategory();
8   }
```

### B-映射文件

创建CategoryMapper.xml

```
1  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
2   "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
3   <mapper namespace="com.ssm.dao.CategoryMapper">
4   </mapper>
```

### C-业务接口

```
1  public interface CategoryService {
2       List<Category> selectAllCategory();
3
4   }
```

### D-业务实现

```

1  @Service
2  public class CategoryServiceImpl implements CategoryService {
3      @Autowired
4      private CategoryMapper cMapper;
5      @Override
6      public List<Category> selectAllCategory() {
7          return cMapper.selectAllCategory();
8      }
9  }

```

## E-控制层

```

1  @Controller
2  public class CategoryController {
3
4      @Autowired
5      private CategoryService cgService;
6
7      @RequestMapping("/cgList.do")
8      @ResponseBody
9      public List<Category> getCategoryList(){
10         List<Category> cgList= cgService.selectAllCategory();
11         if(cgList!=null){
12             return cgList;
13         }
14         return null;
15     }
16 }

```

## F-页面改造

在header.jsp页面中

```

1  <!-- 首页导航 -->
2  <div class="navitem">
3      <ul class="nav" id="category">
4          <li class="nav-active"><a href="index.html">首页</a></li>
5          <li class='cg'></li>
6          <li><a href="favoriterank.html">收藏排行榜</a></li>
7      </ul>
8  </div>

```

## G-Ajax请求

在header.jsp中添加ajax请求

```

1  <script type="text/javascript">
2      $(function(){
3          $.post('cgList.do',function(res){
4              var html="";
5              for(var i=0;i<res.length;i++){
6                  html+="

```

## 六、分类数据模糊查

### A-页面改造

header.jsp页面中

```

1  <form action="routeList.do" method="post">
2      <input name="rname" id="rname" type="text" placeholder="请输入路线名称"
3      class="search_input"
4      value="${rname=='?':rname}">
5      <input type="submit" value="搜索" class="search-button"/>
6  </form>

```

分类超链接改造

```

1  for(var i=0;i<res.length;i++){
2      html+="

```

### B-数据接口

```

1  public interface RouteMapper {
2      List<Route> selectRouteByCidOrRname(@Param("cid")Integer cid,@Param("rname")String
3      rname);
4  }

```

### C-映射文件

```

1 <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
2 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
3 <mapper namespace="com.ssm.dao.RouteMapper">
4
5 <!--根据分类id或者线路名称进行模糊查询 -->
6 <select id="selectRouteByCidOrRname" resultType="route">
7     select rname,price,routeIntroduce,rimage from tab_route
8     <trim prefix="where" prefixOverrides="and || or">
9         <if test="cid!=null">cid=#{cid}</if>
10        <if test="rname!=null and rname!='">and rname like concat('%',{rname},'%')
11    </if>
12    </trim>
13 </select>
14 </mapper>

```

## D-业务接口

```

1 public interface RouteService {
2     List<Route> selectRouteByCidOrRname(Integer cid,String rname);
3 }

```

## E-业务实现

```

1 @Service
2 public class RouteServiceImpl implements RouteService {
3
4     @Autowired
5     private RouteMapper rMapper;
6
7     @Override
8     public List<Route> selectRouteByCidOrRname(Integer cid, String rname) {
9         return rMapper.selectRouteByCidOrRname(cid, rname);
10    }
11 }

```

## F-控制层

```

1 @Controller
2 public class RouteController {
3     @Autowired
4     private RouteService rService;
5     @RequestMapping("/routeList.do")
6     public ModelAndView getRouteList(
7         @RequestParam(name="pno",required=true,defaultValue="1")Integer pno,
8         @RequestParam(name="cid",required=true,defaultValue="5")Integer cid,
9         @RequestParam(name="rname",required=true,defaultValue="")String rname){
10        ModelAndView mv=new ModelAndView();
11        PageHelper.startPage(pno,10); //开始分页

```



```

12     List<Route> rlist= rService.selectRouteByCidOrRname(cid,rname);
13     PageInfo<Route> page=new PageInfo<Route>(rlist); //分好页的数据
14     if(rlist!=null){
15         mv.addObject("rlist",rlist);
16         mv.addObject("rname",rname);
17         mv.addObject("pno",pno);
18         mv.addObject("page",page);
19         mv.addObject("cid",cid);
20         mv.setViewName("route_list"); //逻辑视图名
21         return mv;
22     }
23     return null;
24 }
25 }

```

## G-页面展示

在route\_list.jsp中

```

1  <ul>
2      <c:forEach items="${rlist}" var="r">
3          <li>
4              <div class="img">
5                  
6              </div>
7              <div class="text1">
8                  <p>${r.rname}</p>
9                  <br />
10                 <p>${r.routeIntroduce}</p>
11             </div>
12             <div class="price">
13                 <p class="price_num">
14                     <span>&yen;</span> <span>${r.price}</span> <span>起</span>
15                 </p>
16                 <p>
17                     <a href="routeDetail?rid=${r.rid}">查看详情</a>
18                 </p>
19             </div>
20         </li>
21     </c:forEach>
22 </ul>

```

## H-百度分页效果分析

### 1、判断上页和下页的形成条件

上一页条件是:当前页码大于1，下一页条件是:当前页码大于1且当前页码小于总页数

### 2、上一页和下一页中间遍历部分

a、总页数小于等于10

b、总页数大于10时候

当前页码小于等于6和大于6情况

## G-分页效果展示

```
1 <div class="page_num_inf">
2   <i></i> 共 <span>${page.pages}</span>页<span>${page.total}</span>条
3 </div>
4 <div class="pageNum">
5   <ul>
6     <c:if test="${pno > 1}">
7       <li class="threeword"><a
8                                     href="routeList.do?
pno=${page.prePage}&cid=${cid}&rname=${rname}">&lt;t;上一页</a></li>
9     </c:if>
10    <c:choose>
11      <c:when test="${page.pages<=10}">
12        <c:forEach begin="1" end="${page.pages}" var="i">
13          <c:if test="${pno==i}">
14            <li class="curPage"><a
15                                          href="routeList.do?
pno=${i}&cid=${cid}&rname=${rname}">${i}</a></li>
16          </c:if>
17          <c:if test="${pno!=i}">
18            <li><a href="routeList.do?
pno=${i}&cid=${cid}&rname=${rname}">${i}</a></li>
19          </c:if>
20        </c:forEach>
21      </c:when>
22      <c:when test="${page.pages>10}">
23        <c:if test="${pno<=6}">
24          <c:forEach begin="1" end="10" var="i">
25            <c:if test="${pno==i}">
26              <li class="curPage"><a
27                                          href="routeList.do?
pno=${i}&cid=${cid}&rname=${rname}">${i}</a></li>
28            </c:if>
29            <c:if test="${pno!=i}">
30              <li><a href="routeList.do?
pno=${i}&cid=${cid}&rname=${rname}">${i}</a></li>
31            </c:if>
32          </c:forEach>
33        </c:if>
34        <c:if test="${pno>6}">
35          <c:forEach begin="${pno-5}" end="${(pno+4)>page.pages?page.pages:
(pno+4)}" var="i">
36            <c:if test="${pno==i}">
37              <li class="curPage"><a
38                                          href="routeList.do?
pno=${i}&cid=${cid}&rname=${rname}">${i}</a></li>
39            </c:if>
40            <c:if test="${pno!=i}">
```

```

41         <li><a href="routeList.do?
pno=${i}&cid=${cid}&rname=${rname}">${i}</a></li>
42     </c:if>
43 </c:forEach>
44 </c:if>
45 </c:when>
46 </c:choose>
47 <c:if test="${pno}>=1 && pno<page.pages}">
48     <li class="threeword"><a
49         href="routeList.do?
pno=${page.nextPage}&cid=${cid}&rname=${rname}">下一页</a></li>
50 </c:if>
51 </ul>
52 </div>

```

## 七、线路详情查询

### A-功能分析

详情页的数据主要来自3张表分别是：tab\_route(线路表)、tab\_route\_img(线路图片表)、tab\_seller(商家表)

线路表和商家表是一对一关系通过外键sid关联、线路表和线路图片表是一对多关系通过外键rid进行关联

以线路为主，在映射文件当中配置一对一和一对多关系

### B-数据接口

```

1 //根据线路id查询线路
2 Route selectByRid(Integer rid);

```

### C-映射文件

```

1 <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
2 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
3 <mapper namespace="com.ssm.dao.RouteMapper">
4
5 <resultMap type="route" id="rResultMap">
6     <id property="rid" column="rid"/>
7     <result property="rname" column="rname"/>
8     <result property="price" column="price"/>
9     <result property="routeIntroduce" column="routeIntroduce"/>
10    <result property="rflag" column="rflag"/>
11    <result property="rdate" column="rdate"/>
12    <result property="isThemeTour" column="isThemeTour"/>
13    <result property="count" column="count"/>
14    <result property="cid" column="cid"/>
15    <result property="rimage" column="rimage"/>
16    <result property="sid" column="sid"/>
17    <result property="sourceId" column="sourceId"/>

```

```

18      <!--一对一-->
19      <association property="seller" javaType="seller">
20          <id property="sid" column="sid"/>
21          <result property="sname" column="sname"/>
22          <result property="address" column="address"/>
23          <result property="consphone" column="consphone"/>
24      </association>
25      <!--一对多-->
26      <collection property="routeImgList" ofType="routeImg">
27          <id property="rgid" column="rgid"/>
28          <result property="rid" column="rid"/>
29          <result property="bigPic" column="bigPic"/>
30          <result property="smallPic" column="smallPic"/>
31      </collection>
32  </resultMap>
33
34  <!--根据线路id查询线路 -->
35  <select id="selectByRid" resultMap="rResultMap">
36      select r.*,i.*,s.* from tab_route r,tab_route_img i,tab_seller s
37      where r.rid=i.rid and r.sid=s.sid and r.rid=#{rid}
38  </select>
39
40  </mapper>

```

## D-业务接口

```

1  Route selectByRid(Integer rid);

```

## E-业务实现

```

1  @Override
2  public Route selectByRid(Integer rid) {
3      return rMapper.selectByRid(rid);
4  }

```

## F-控制层

```

1  /**
2   * 根据线路id查询线路详情
3   */
4  @RequestMapping("/routeDetail.do")
5  public ModelAndView routeDetail(
6      @RequestParam(name="rid",required=true,defaultValue="")Integer rid){
7      ModelAndView mv=new ModelAndView();
8      Route rt = rService.selectByRid(rid);
9      if(rt!=null){
10         List<RouteImg> imglist=rt.getRouteImgList();
11         Seller sl=rt.getSeller();
12         mv.addObject("rt",rt); //线路对象

```

```

13     mv.addObject("imglist",imglist);//图片集合
14     mv.addObject("s1",s1);//商家
15     mv.setViewName("route_detail");//去线路详情页面
16     return mv;
17 }
18 return null;
19 }

```

## G-页面展示

在route\_detail.jsp中

```

1  <div class="prosum_box">
2      <dl class="prosum_left">
3          <dt>
4              
5          </dt>
6          <dd>
7              <a class="up_img up_img_disable"></a>
8              <c:forEach items="${imglist}" var="i" varStatus="v">
9                  <c:if test="${v.count>4}">
10                     <a title="" class="little_img" data-bigpic="${i.bigPic}"
11 style="display:none;">
12                         
13                     </a>
14                 </c:if>
15                 <c:if test="${v.count<=4}">
16                     <a title="" class="little_img" data-bigpic="${i.bigPic}">
17                         
18                     </a>
19                 </c:if>
20             </c:forEach>
21             <a class="down_img down_img_disable" style="margin-bottom: 0;"></a>
22         </dd>
23     </dl>
24     <div class="prosum_right">
25         <p class="pros_title">${rt.rname}</p>
26         <p class="hot">${rt.routeIntroduce}</p>
27         <div class="pros_other">
28             <p>经营商家    : ${s1.sname}</p>
29             <p>咨询电话  : ${s1.consphone}</p>
30             <p>地址    : ${s1.address}</p>
31         </div>
32         <div class="pros_price">
33             <p class="price"><strong>¥${rt.price}</strong><span>起</span></p>
34             <p class="collect">
35                 <a class="btn"><i class="glyphicon glyphicon-heart-empty"></i>点击收藏
36             </p>
37             <span>已收藏${rt.count}次</span>
38         </div>

```

## 八、收藏功能

### A-思路分析

首先用户点击收藏时需要判断用户是否登录，如果没有登录给其提示并且跳转到登录页面

如果已经登录，需要查询该用户是否已经收藏过该线路，如果有收藏，则提示您已收藏过

如果已经登录，但是之前没有收藏过此线路，则将收藏的信息保存到数据库并更新线路的收藏次数

### B-底层方法

- 1、根据用户id和线路id查询是否有收藏
- 2、添加收藏的方法
- 3、更新线路收藏次数的方法

### C-数据接口1

在FavoriteMapper中

```
1  /**
2   * 收藏Mapper
3   */
4  public interface FavoriteMapper {
5      //根据线路id和用户id查询是否有收藏信息
6      @Select("select * from tab_favorite where rid=#{rid} and uid=#{uid}")
7      Favorite selectFtByRidAndUid(@Param("rid") Integer rid, @Param("uid") Integer uid);
8
9      @Insert("insert into tab_favorite(rid,date,uid) values(#{rid},now(),#{uid})")
10     int addCollect(Favorite ft);
11 }
```

### D-数据接口2

在RouteMapper接口中

```
1  //更新某个线路的收藏次数
2  @Update("update tab_route set count=#{count} where rid=#{rid}")
3  int updateCount(Route rt);
```

### E-业务接口

```
1 public interface FavoriteService {
2     //收藏功能
3     boolean doCollect(Favorite ft);
4 }
```

## F-业务实现

```
1 //收藏业务实现
2 @Service
3 public class FavoriteServiceImpl implements FavoriteService {
4
5     @Autowired
6     private FavoriteMapper ftMapper;
7
8     @Autowired
9     private RouteMapper rMapper;
10
11     @Transactional
12     public boolean doCollect(Favorite ft) {
13         Favorite ft1=ftMapper.selectFtByRidAndUid(ft.getRid(),ft.getUid());
14         if(ft1==null){
15             if(ftMapper.addCollect(ft)>0){
16                 Route route=rMapper.selectByRid(ft.getRid());
17                 route.setCount(route.getCount()+1);
18                 rMapper.updateCount(route);
19                 return true;
20             }
21             return false;
22         }
23         return false;
24     }
25 }
```

## G-控制层

```
1 @Controller
2 public class FavoriteController {
3     @Autowired
4     private FavoriteService fService;
5     /**
6      * 收藏
7      */
8     @RequestMapping("/doCollect.do")
9     @ResponseBody
10    public String doCollect(Favorite ft){
11        boolean flag=fService.doCollect(ft);
12        if(flag){
13            return "true";
14        }
15        return "false";
16    }
```

```
16     }
17
18 }
```

## H-页面请求

```
1  <script type="text/javascript">
2      $(function(){
3          $(".btn").click(function(){
4              var uname='${sessionScope.user.username}';
5              if(uname==null||uname==""){
6                  alert("请登录再收藏!");
7                  window.location.href='toLogin.do';
8                  return;
9              }
10             //已登录 -发送ajax判断该用户是否有收藏该线路
11             var rid='${rt.rid}';
12             var uid='${sessionScope.user.uid}';
13             $.post('doCollect.do',{ "rid":rid,"uid":uid},function(res){
14                 if(res=="false"){
15                     alert("您已收藏过该线路!");
16                     $(".btn").attr("disabled","disabled"); //禁用
17                 }
18                 alert("收藏成功!");
19                 $(".btn").attr("disabled","disabled"); //禁用
20             })
21         });
22     })
23 </script>
```