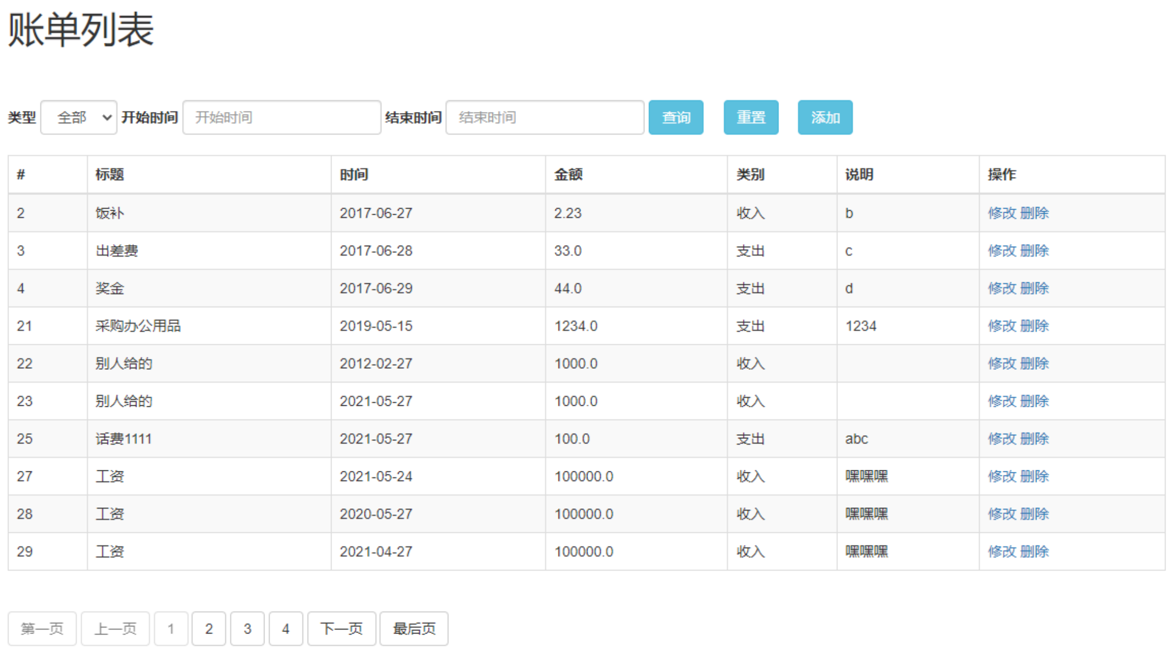
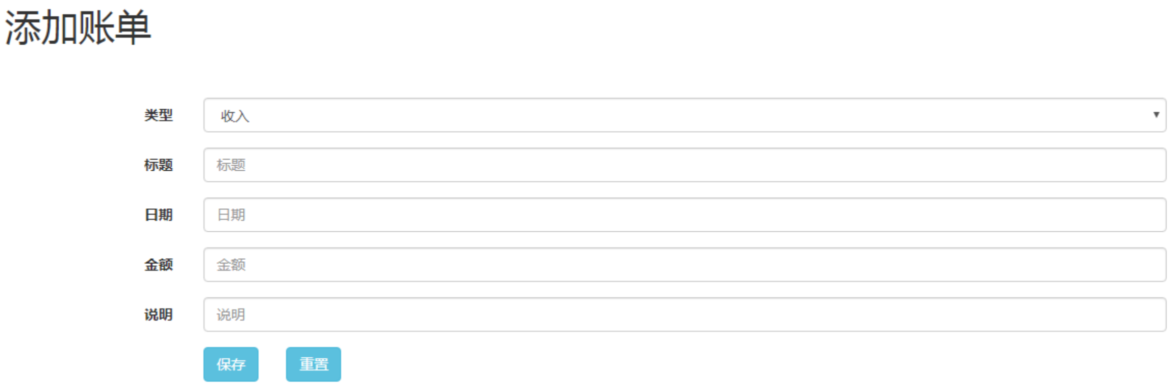
记账管理项目

> 需求：实现个人记账管理项目



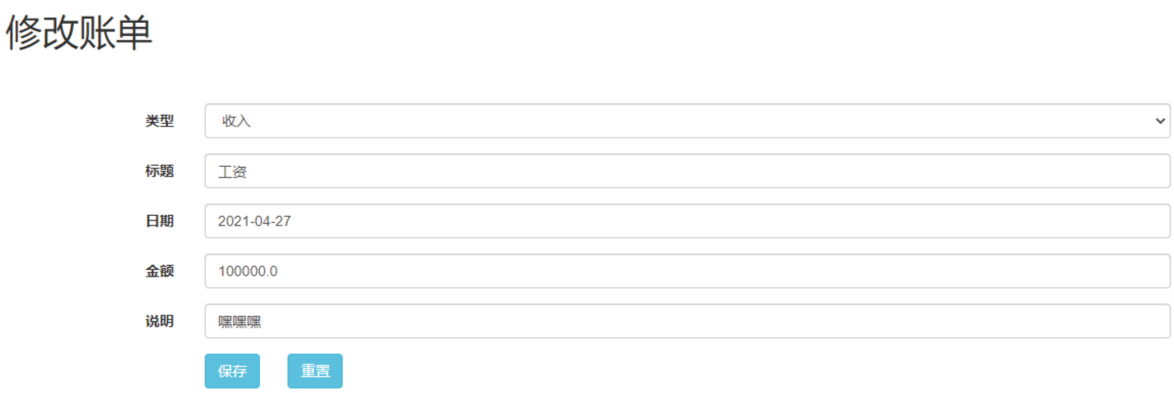
账单列表

#	标题	时间	金额	类别	说明	操作
2	饭补	2017-06-27	2.23	收入	b	修改 删除
3	出差费	2017-06-28	33.0	支出	c	修改 删除
4	奖金	2017-06-29	44.0	支出	d	修改 删除
21	采购办公用品	2019-05-15	1234.0	支出	1234	修改 删除
22	别人给的	2012-02-27	1000.0	收入		修改 删除
23	别人给的	2021-05-27	1000.0	收入		修改 删除
25	话费1111	2021-05-27	100.0	支出	abc	修改 删除
27	工资	2021-05-24	100000.0	收入	嘿嘿嘿	修改 删除
28	工资	2020-05-27	100000.0	收入	嘿嘿嘿	修改 删除
29	工资	2021-04-27	100000.0	收入	嘿嘿嘿	修改 删除



添加账单

类型	收入
标题	标题
日期	日期
金额	金额
说明	说明



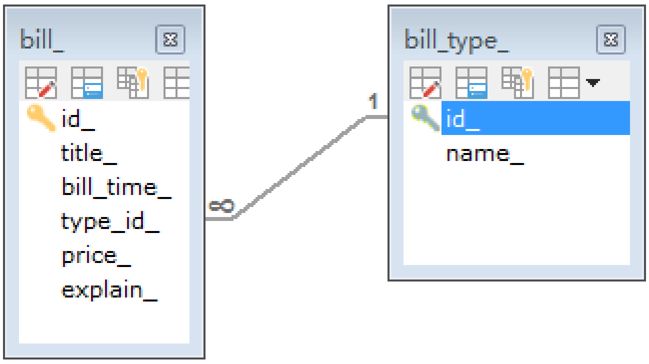
修改账单

类型	收入
标题	工资
日期	2021-04-27
金额	100000.0
说明	嘿嘿嘿

> 技术栈

- springboot
- mybatis
- tk mybatis
- thymeleaf

> 表结构



> 创建项目，引入依赖

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.yemage</groupId>
    <artifactId>bill-manager-tk</artifactId>
    <version>1.0-SNAPSHOT</version>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.0.0.RELEASE</version>
    </parent>

    <properties>
        <java.version>1.8</java.version>
    </properties>

    <dependencies>
        <dependency>
```

```
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
        </dependency>

        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>5.1.46</version>
        </dependency>

        <!-- 通用mapper -->
        <dependency>
            <groupId>tk.mybatis</groupId>
            <artifactId>mapper-spring-boot-starter</artifactId>
            <version>2.0.2</version>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-thymeleaf</artifactId>
        </dependency>

    </dependencies>

</project>
```

> spring boot 启动器

```
@SpringBootApplication
@MapperScan("com.yemage.dao")
public class BillManagerApplication {

    public static void main(String[] args) {

        SpringApplication.run(BillManagerApplication.class, args);
    }
}
```

> application.properties：springboot全局配置文件(可以拷贝)

```
#连接池
spring:
  datasource:
    driver-class-name: com.mysql.jdbc.Driver
    username: root
    password: 123456
    url: jdbc:mysql://localhost:3306/bill-manager
  thymeleaf:
    cache: false #关闭thymeleaf缓存

#整合mybatis
```

```
mybatis:
  type-aliases-package:  com.yemage.entity #别名搜索
  mapper-locations: classpath:/mybatis/*.xml
```

> 实体类

账单类型

```
@Table(name = "bill_type_")
public class BillType implements Serializable {

   @Id
   @GeneratedValue(strategy = GenerationType.IDENTITY)
   @Column(name = "id_")
   private Long id;

   @Column(name = "name_")
   private String name;
```

账单

```
@Table(name = "bill_")
public class Bill implements Serializable {

   @Id
   @GeneratedValue(strategy = GenerationType.IDENTITY)
   @Column(name = "id_")
   private Long id;
   @Column(name = "title_")
   private String title;
   @Column(name = "bill_time_")
   private Date  billTime;
   @Column(name = "type_id_")
   private Long  typeId;
   @Column(name = "price_")
   private Double price;
   @Column(name = "explain_")
   private String explain;

   /**
    * 类别名称
    */
   @Transient
   private String typeName;

   /**
    * 开始时间:用于查询
    */
   @Transient
   private Date date1;

   /**
    * 结束时间：用于查询
    */
   @Transient
   private Date date2;
```

注意 @Transient注解表示当前属性为瞬时属性，跟字段没有映射，上面实体类中date1和date2只作为查询条件存在，跟表中的字段没有关联

> dao

```
public interface BillMapper extends Mapper<Bill> {

   public List<Bill> select(Bill b);

}
```

```
public interface TypeMapper extends Mapper<BillType> {

}
```

> 查询修改的映射文件

tk mybatis内置查询比较单一固定，不灵活，不能实现动态查询，动态修改语句， 所以特定需求的查询需要写映射文件

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.yemage.dao.BillMapper">

    <sql id="selectSql">
        SELECT
            b.id_ as id,
            b.title_ as title,
            b.bill_time_ as billTime,
            b.type_id_ as typeId,
            b.price_ as price,
            b.explain_ as `explain`,
            t.name_ as typeName
        FROM
            bill_ as b
        left join
            bill_type_ as t
        on
            b.type_id_ = t.id_
    </sql>

    <select id="select" resultType="bill">
        <include refid="selectSql"/>
        <where>
            <if test="typeId !=null">
                b.type_id_ = #{typeId}
            </if>
            <if test="title !=null">
                and b.title_ like '%${title}%'
            </if>
            <if test="date1 !=null">
                and b.bill_time_ &gt;= #{date1}
            </if>
            <if test="date2 !=null">
                and b.bill_time_ &lt;= #{date2}
            </if>
        </where>
    </select>

</mapper>
```

注意：explain字段名在mysql 中是关键词必须使用``包裹

> service

```
@Service
public class BillService {

   @Autowired
   private BillMapper billMapper;

   public List<Bill> list(Bill b) {
      return billMapper.select(b);
   }

   public int add(Bill b) {
      return billMapper.insert(b);
   }

   public Bill get(Long id) {
      return billMapper.selectByPrimaryKey(id);
   }

   public int update(Bill b) {
      return billMapper.updateByPrimaryKey(b);
   }

   public int delete(Long id) {
      return billMapper.deleteByPrimaryKey(id);
   }


}
```

```
@Service
public class TypeService {

   @Resource
   private TypeMapper typeMapper;

   public List<BillType> list() {
      return typeMapper.selectAll();
   }

}
```

> 测试

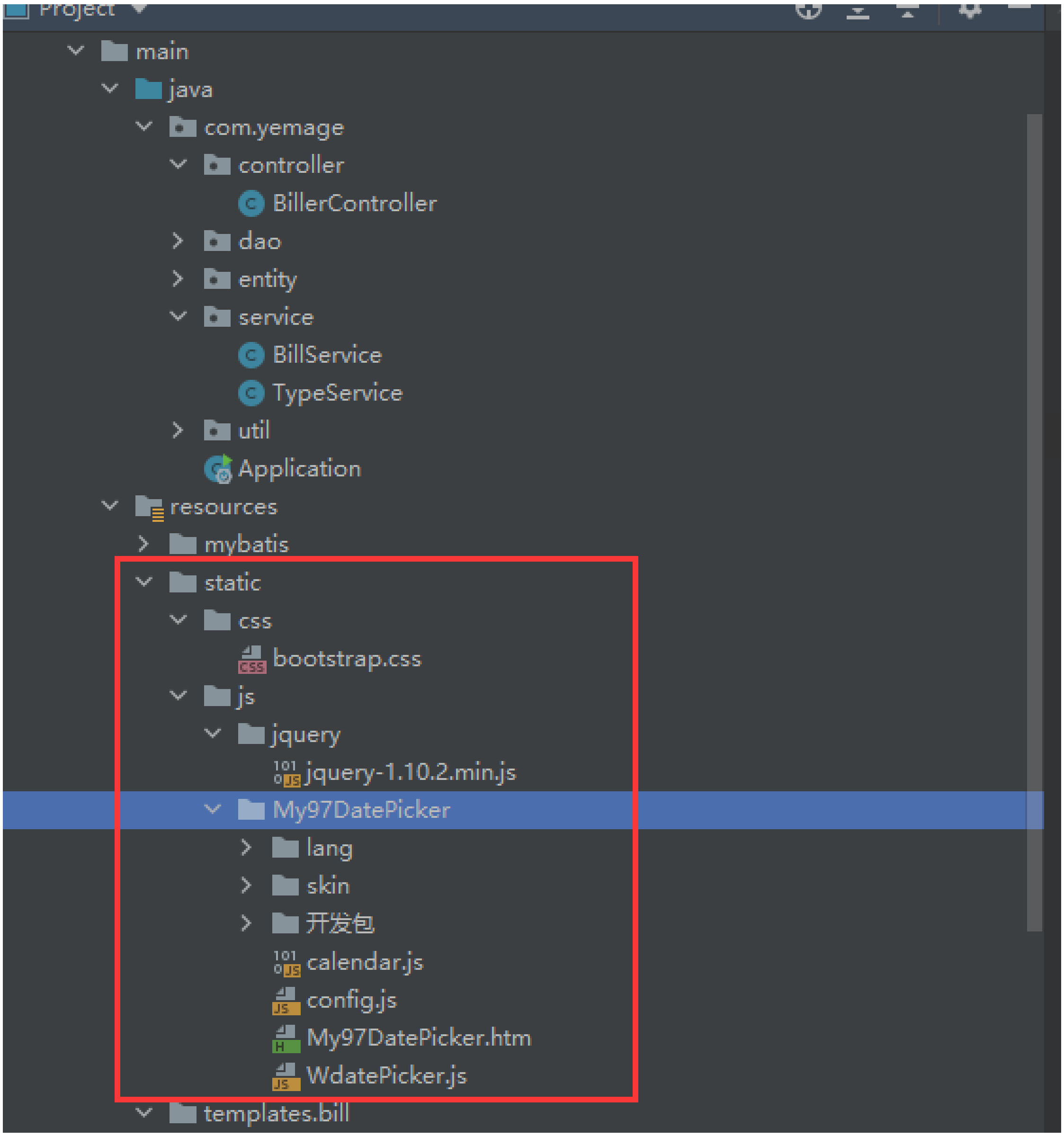
```
@RunWith(SpringRunner.class)
@SpringBootTest
public class BillServiceTest {

    @Autowired
    private BillService billService;

    @Test
    public void list() {
        Bill b = new Bill();
        b.setTitle("a");
        List<Bill> list = billService.list(b);
        for (Bill bill : list) {
            System.out.println(bill.getTitle());
        }
    }
}
```

> controller

```
@Controller
@RequestMapping("/bill")
public class BillController {

    @Resource
    private TypeService typeService;

    @Resource
    private BillService billService;

    /**
     * 查询
     * @param b
     * @param model
     * @return
     */
    @RequestMapping("/list")
    public String list(Bill b, Model model) {
        List<BillType> types = typeService.list();
        model.addAttribute("types", types);

        List<Bill> list = billService.list(b);
        model.addAttribute("list", list);

        model.addAttribute("bill", b);

        return "/bill/list";
    }

    /**
     * 调到添加页面
     * @return
     */
    @RequestMapping("/toAdd")
    public String toAdd(Model model) {
        List<BillType> types = typeService.list();
        model.addAttribute("types", types);
        return "/bill/add";
    }

    /**
     * 添加
     * @param b
     * @return
     */
    @RequestMapping("/add")
    public String add(Bill b) {
        billService.add(b);
        return "redirect:/bill/list";
    }

    /**
     * 删除
     * @param id
     * @return
     */
    @RequestMapping("/delete/{id}")
    public String delete(@PathVariable("id") Long id) {
        billService.delete(id);
        return "redirect:/bill/list";
    }

    /**
     * 修改
     * @param id
     * @return
     */
    @RequestMapping("/toUpdate/{id}")
    public String toUpdate(@PathVariable("id") Long id, Model model) {
        List<BillType> types = typeService.list();
        model.addAttribute("types", types);

        Bill bill = billService.get(id);
        model.addAttribute("bill", bill);

        return "/bill/update";
    }

    /**
     * 修改
     * @param b
     * @return
     */
    @RequestMapping("/update")
    public String update(Bill b) {
        billService.update(b);
        return "redirect:/bill/list";
    }

}
```

> 页面

拷贝素材中我们用到的bootstrap.css样式和jquery和my97日历控件相关静态资源到/resources/static目录



```
Project
    main
        java
            com.yemage
                controller
                    BillerController
                dao
                entity
                service
                    BillService
                    TypeService
                util
                Application
        resources
            mybatis
            static
                css
                    bootstrap.css
                js
                    jquery
                        jquery-1.10.2.min.js
                    My97DatePicker
                        lang
                        skin
                        开发包
                        calendar.js
                        config.js
                        My97DatePicker.htm
                        WdatePicker.js
            templates.bill
```

list.html

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8"/>
    <title>userList</title>
    <link rel="stylesheet" th:href="@{/css/bootstrap.css}"></link>
    <script type="text/javascript"
th:src="@{/js/My97DatePicker/WdatePicker.js}"></script>
    <script type="text/javascript" src="/js/My97DatePicker/lang/zh-cn.js">
</script>
    <script type="text/javascript" th:src="@{/js/jquery/jquery-1.10.2.min.js}">
</script>
</head>
<body class="container">
<br/>
<h1>账单列表</h1>
<br/><br/>
<div class="with:80%">
```

```
    <form class="form-inline" id="qf"   th:action="@{/bill/list}"
 method="post">

        <div class="form-group">
            <label for="typeId" class="control-label">类型</label>
            <select name="typeId" id="typeId" class="form-control">
                <option value="">全部</option>
                <option th:each="t:${types}" th:value="${t.id}"
th:text="${t.name}" th:selected="(${bill.typeId} == ${t.id})"  ></option>
            </select>
        </div>
        <div class="form-group">
            <label for="date1" class="control-label" >开始时间</label>
            <input type="text" class="form-control" name="date1" id="date1"
placeholder="开始时间" th:value="${bill.date1} ? ${#dates.format(bill.date1,
'yyyy-MM-dd')}" onclick="WdatePicker()"/>
        </div>
        <div class="form-group">
            <label for="date2" class="control-label">结束时间</label>
            <input type="text" class="form-control" name="date2"  id="date2"
placeholder="结束时间" th:value="${bill.date2} ? ${#dates.format(bill.date2,
'yyyy-MM-dd')}" onclick="WdatePicker()"/>
        </div>
        <div class="form-group">
            <input type="submit" value="查询" class="btn btn-info" />
            &nbsp; &nbsp;
            <input type="reset" value="重置" class="btn btn-info" />
            &nbsp; &nbsp;
            <a href="/bill/toAdd" th:href="@{/bill/toAdd}" class="btn btn-info">
添加</a>
        </div>
    </form>
</div>
<br/>

<div class="with:80%">
    <table class="table table-striped table-bordered">
        <thead>
        <tr>
            <th>#</th>
            <th>标题</th>
            <th>时间</th>
            <th>金额</th>
            <th>类别</th>
            <th>说明</th>
            <th>操作</th>
        </tr>
        </thead>
        <tbody>
        <tr  th:each="b, bstatus : ${list}" th:style="${bstatus.odd} ?
'background-color: #A3C6C8'">
            <th scope="row" th:text="${bill.id}">id</th>
            <td th:text="${b.title}">name</td>
            <td th:text="${b.billTime} ? ${#dates.format(b.billTime, 'yyyy-MM-
dd')}">time</td>
            <td th:text="${b.price}">price</td>
            <td th:text="${b.typeName}">typeName</td>
            <td th:text="${b.explain}">explain</td>
            <td>
                <a th:href="|/bill/delete/*{id}|">删除</a>
                <a th:href="|/bill/toUpdate/*{id}|">修改</a>
            </td>
        </tr>
        </tbody>
    </table>
</div>

</body>
</html>
```

> add.html

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8"/>
    <title>user</title>
    <link rel="stylesheet" th:href="@{/css/bootstrap.css}"></link>
    <script type="text/javascript"
th:src="@{/js/My97DatePicker/WdatePicker.js}"></script>
    <script type="text/javascript" src="/js/My97DatePicker/lang/zh-cn.js">
</script>


</head>
<body class="container">
<br/>
<h1>添加账单</h1>
<br/><br/>
<div class="with:80%">
    <form class="form-horizontal"   th:action="@{/bill/add}"  method="post"
enctype="multipart/form-data">
        <div class="form-group">
            <label for="typeId" class="col-sm-2 control-label">类型</label>
            <div class="col-sm-10">
                <select name="typeId" id="typeId" class="form-control">
                    <option th:each="t:${types}" th:value="${t.id}"
th:text="${t.name}" ></option>
                </select>
            </div>
        </div>
        <div class="form-group">
            <label for="title" class="col-sm-2 control-label" >标题</label>
            <div class="col-sm-10">
                <input type="text" class="form-control" name="title" id="title"
placeholder="标题"/>
            </div>
        </div>
        <div class="form-group">
            <label for="billTime" class="col-sm-2 control-label">日期</label>
            <div class="col-sm-10">
                <input type="text" class="form-control" name="billTime"
 id="billTime" placeholder="日期" onclick="WdatePicker()"/>
            </div>
        </div>
        <div class="form-group">
            <label for="price" class="col-sm-2 control-label">金额</label>
            <div class="col-sm-10">
                <input type="text" class="form-control" name="price"  id="price"
placeholder="金额"/>
            </div>
        </div>
        <div class="form-group">
            <label for="explain" class="col-sm-2 control-label">说明</label>
            <div class="col-sm-10">
                <input type="text" class="form-control" name="explain"
 id="explain" placeholder="说明"/>
            </div>
        </div>

        <div class="form-group">
            <div class="col-sm-offset-2 col-sm-10">
                <input type="submit" value="保存" class="btn btn-info" />
                &nbsp; &nbsp; &nbsp;
                <input type="reset" value="重置" class="btn btn-info" />
            </div>

        </div>
    </form>
</div>
</body>
</html>
```

> update.html

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8"/>
    <title>user</title>
    <link rel="stylesheet" th:href="@{/css/bootstrap.css}"></link>
    <script type="text/javascript"
th:src="@{/js/My97DatePicker/WdatePicker.js}"></script>
    <script type="text/javascript" src="/js/My97DatePicker/lang/zh-cn.js">
</script>
</head>
<body class="container">
<br/>
<h1>修改账单</h1>
<br/><br/>
<div class="with:80%">
    <form class="form-horizontal"   th:action="@{/bill/update}"  method="post"
th:object="${bill}">
        <input type="hidden" name="id" th:value="*{id}" />
        <div class="form-group">
            <label for="typeId" class="col-sm-2 control-label">类型</label>
            <div class="col-sm-10">
                <select name="typeId" id="typeId" class="form-control">
                    <option th:each="t:${types}" th:value="${t.id}"
th:text="${t.name}" th:selected="${t.id} == *{typeId}" ></option>
```

```
                </select>
            </div>
        </div>
        <div class="form-group">
            <label for="title" class="col-sm-2 control-label" >标题</label>
            <div class="col-sm-10">
                <input type="text" class="form-control" name="title" id="title"
th:value="*{title}" placeholder="标题"/>
            </div>
        </div>
        <div class="form-group">
            <label for="billTime" class="col-sm-2 control-label">日期</label>
            <div class="col-sm-10">
                <input type="text" class="form-control" name="billTime"
 id="billTime"  th:value="*{billTime} ? *{#dates.format(billTime, 'yyyy-MM-
dd')}" placeholder="日期" onclick="WdatePicker()"/>
            </div>
        </div>
        <div class="form-group">
            <label for="price" class="col-sm-2 control-label">金额</label>
            <div class="col-sm-10">
                <input type="text" class="form-control" name="price"  id="price"
th:value="*{price}" placeholder="金额"/>
            </div>
        </div>
        <div class="form-group">
            <label for="explain" class="col-sm-2 control-label">说明</label>
            <div class="col-sm-10">
                <input type="text" class="form-control" name="explain"
 id="explain" th:value="*{explain}" placeholder="说明"/>
            </div>
        </div>

        <div class="form-group">
            <div class="col-sm-offset-2 col-sm-10">
                <input type="submit" value="保存" class="btn btn-info" />
                &nbsp; &nbsp; &nbsp;
                <input type="reset" value="重置" class="btn btn-info" />
            </div>

        </div>
    </form>
</div>
</body>
</html>
```

> 注意在springboot中配置日期类型转换器

```
package com.yemage.util;

import org.springframework.core.convert.converter.Converter;
import org.springframework.stereotype.Component;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

@Component
public class DateConverterConfig implements Converter<String, Date> {

    private static final List<String> formarts = new ArrayList<>(4);
    static{
        formarts.add("yyyy-MM");
        formarts.add("yyyy-MM-dd");
        formarts.add("yyyy-MM-dd hh:mm");
        formarts.add("yyyy-MM-dd hh:mm:ss");
    }

    @Override
    public Date convert(String source) {
        String value = source.trim();
        if ("".equals(value)) {
            return null;
        }
        if(source.matches("^\\d{4}-\\d{1,2}$")){
            return parseDate(source, formarts.get(0));
        }else if(source.matches("^\\d{4}-\\d{1,2}-\\d{1,2}$")){
            return parseDate(source, formarts.get(1));
        }else if(source.matches("^\\d{4}-\\d{1,2}-\\d{1,2}
{1}\\d{1,2}:\\d{1,2}$")){
            return parseDate(source, formarts.get(2));
        }else if(source.matches("^\\d{4}-\\d{1,2}-\\d{1,2}
{1}\\d{1,2}:\\d{1,2}:\\d{1,2}$")){
            return parseDate(source, formarts.get(3));
        }else {
            throw new IllegalArgumentException("Invalid boolean value '" +
source + "'");
        }
    }

    /**
     * 格式化日期
     * @param dateStr String 字符型日期
     * @param format String 格式
     * @return Date 日期
     */
    public  Date parseDate(String dateStr, String format) {
        Date date=null;
        try {
            DateFormat dateFormat = new SimpleDateFormat(format);
            date = dateFormat.parse(dateStr);
        } catch (Exception e) {

        }
        return date;
    }

}
```

> 删除

```
<a th:href="|/bill/delete/*{id}|">删除</a>
```

> 分页

引入pageHelper分页插件的starter，**注意：必须引入pagehelper的启动器，不能直接引入pagehelper**

```
<!-- 分页插件-->
<dependency>
    <groupId>com.github.pagehelper</groupId>
    <artifactId>pagehelper-spring-boot-starter</artifactId>
    <version>1.2.3</version>
</dependency>
```

分页方法的测试

```
    @Test
    public void findPage() {
        Bill b = new Bill();
        b.setTitle("a");

        PageInfo<Bill> page = PageHelper.startPage(2, 10).doSelectPageInfo(() ->
{
            billService.list(b);
        });

        page.getList().forEach(bill -> {
            System.out.println(bill.getId() + " " + bill.getTitle());
        });

        System.out.println("总行数=" + page.getTotal());
        System.out.println("当前页=" + page.getPageNum());
        System.out.println("每页行数=" + page.getPageSize());
        System.out.println("总页数=" + page.getPages());
        System.out.println("起始行数=" + page.getStartRow());

        System.out.println("是第一页=" + page.isIsFirstPage());
        System.out.println("是最后页=" + page.isIsLastPage());

        System.out.println("还有下一页=" + page.isHasNextPage());
        System.out.println("还有上一页=" + page.isHasPreviousPage());
        System.out.println("页码列表" +
Arrays.toString(page.getNavigatepageNums()));

    }
```

service中的分页方法

```
    public PageInfo<Bill> listPage(Bill b, int pageNum, int pageSize) {
        return PageHelper.startPage(pageNum, pageSize).doSelectPageInfo(() -> {
            billMapper.select(b);
        });
    }
```

controller中的方法

```
/**
 * 查询
 * @param b
 * @param model
 * @return
 */
@RequestMapping("/list-page")
public String listPage(
        @RequestParam(defaultValue = "1") int pageNum,
        @RequestParam(defaultValue = "10") int pageSize,
        Bill b,
        Model model) {

    List<BillType> types = typeService.list();
    model.addAttribute("types", types);

    PageInfo<Bill> pageInfo = billService.listPage(b, pageNum, pageSize);
    model.addAttribute("page", pageInfo);

    model.addAttribute("bill", b);

    return "/bill/list-page";
}
```

前端分页

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8"/>
    <title>userList</title>
    <link rel="stylesheet" th:href="@{/css/bootstrap.css}"></link>
    <script type="text/javascript"
th:src="@{/js/My97DatePicker/WdatePicker.js}"></script>
    <script type="text/javascript" th:src="@{/js/jquery/jquery-1.10.2.min.js}">
</script>
</head>
<body class="container">
<br/>
<h1>账单列表</h1>
<br/><br/>
<div class="with:80%">
    <form class="form-inline" id="qf"   th:action="@{/bill/list-page}"
 method="post">
        <input type="hidden" name="pageNum" id="pageNum"
 th:value="${page.pageNum }" />
        <input type="hidden" name="pageSize" id="pageSize"
th:value="${page.pageSize }" />


        <div class="form-group">
            <label for="typeId" class="control-label">类型</label>
            <select name="typeId" id="typeId" class="form-control">
```

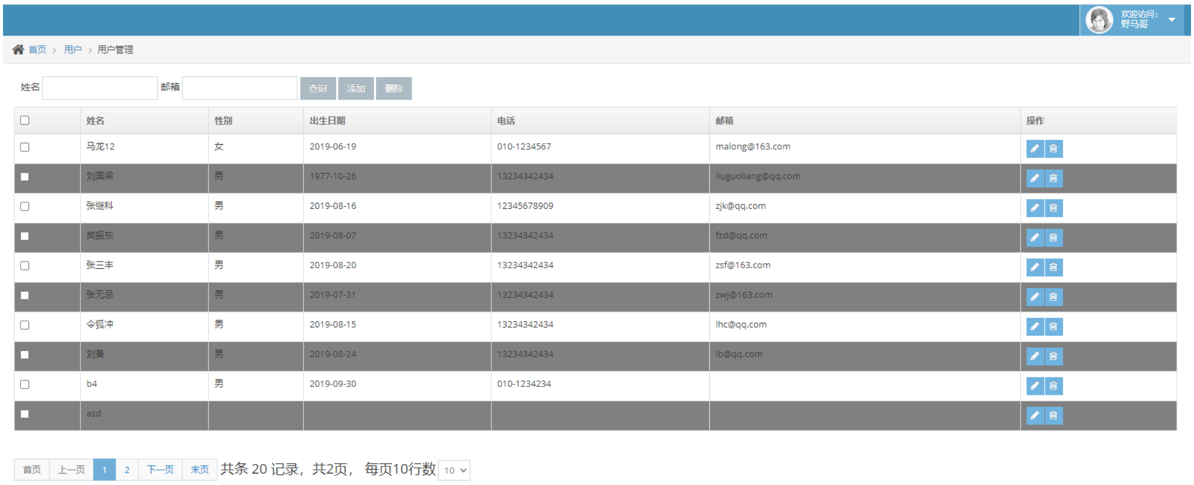
```
                <option value="">全部</option>
                <option th:each="t:${types}" th:value="${t.id}"
th:text="${t.name}" th:selected="(${bill.typeId} == ${t.id})"  ></option>
            </select>
        </div>
        <div class="form-group">
            <label for="date1" class="control-label" >开始时间</label>
            <input type="text" class="form-control" name="date1" id="date1"
placeholder="开始时间" th:value="${bill.date1} ? ${#dates.format(bill.date1,
'yyyy-MM-dd')}" onclick="WdatePicker()"/>
        </div>
        <div class="form-group">
            <label for="date2" class="control-label">结束时间</label>
            <input type="text" class="form-control" name="date2"  id="date2"
placeholder="结束时间" th:value="${bill.date2} ? ${#dates.format(bill.date2,
'yyyy-MM-dd')}" onclick="WdatePicker()"/>
        </div>
        <div class="form-group">
            <input type="submit" value="查询" class="btn btn-info" />
            &nbsp; &nbsp;
            <input type="reset" value="重置" class="btn btn-info" />
            &nbsp; &nbsp;
            <a href="/bill/toAdd" th:href="@{/bill/toAdd}" class="btn btn-info">
添加</a>
        </div>
    </form>
</div>
<br/>

<div class="with:80%">
    <table class="table table-striped table-bordered">
        <thead>
        <tr>
            <th>#</th>
            <th>标题</th>
            <th>时间</th>
            <th>金额</th>
            <th>类别</th>
            <th>说明</th>
            <th>操作</th>
        </tr>
        </thead>
        <tbody>
        <tr  th:each="b, bstatus : ${page.list}" th:style="${bstatus.odd} ?
'background-color: #A3C6C8'">
            <th scope="row" th:text="${bill.id}">id</th>
            <td th:text="${b.title}">name</td>
            <td th:text="${b.billTime} ? ${#dates.format(b.billTime, 'yyyy-MM-
dd')}">time</td>
            <td th:text="${b.price}">price</td>
            <td th:text="${b.typeName}">typeName</td>
            <td th:text="${b.explain}">explain</td>

            <td>
                <a th:href="|/bill/toUpdate/${b.id}|">修改</a>
                <a th:href="|/bill/download/${b.id}|">下载</a>
            </td>
        </tr>
        </tbody>
    </table>
</div>

<ul class="pagination">
    <li><button class="btn btn-default" id="first">第一页</button></li>
    <li><button class="btn btn-default" id="prev">上一页</button></li>
    <li th:each="p:${page.navigatepageNums}">
        <button class="btn btn-default" name="pn" th:text="${p}" th:disabled="(${p} == ${page.pageNum})"></button>
    </li>
    <li><button class="btn btn-default" id="next">下一页</button></li>
    <li><button class="btn btn-default" id="last">最后页</button></li>
</ul>

<script th:inline="javascript">
    /*<![CDATA[*/
    $(function(){
        //初始化变量
        var pageNum = [[${page.pageNum}]]; //当前页
        var pageCount = [[${page.pages}]];//最后页
        var hasNextPage = [[${page.hasNextPage}]];//还有下一页
        var hasPreviousPage = [[${page.hasPreviousPage}]];//还有上一页

        $("#next").click(function(){
            $("#pageNum").val(pageNum + 1);
            $("#qf").submit();
        });

        $("#prev").click(function(){
            $("#pageNum").val(pageNum - 1);
            $("#qf").submit();
        });

        $("#first").click(function(){
            $("#pageNum").val(1);
            $("#qf").submit();
        });

        $("#last").click(function(){
            $("#pageNum").val(pageCount);
            $("#qf").submit();
        });

        //没有上一页
        if (!hasPreviousPage) {
            $("#prev").prop("disabled", true);
            $("#first").prop("disabled", true);
        };

        //没有下一页
        if (!hasNextPage) {
            $("#next").prop("disabled", true);
            $("#last").prop("disabled", true);
        };

        $("button[name='pn']").click(function(){
            $("#pageNum").val($(this).html());
            $("#qf").submit();
        });

    });
    /*]]>*/
</script>


</body>
</html>
```

旅游网站后台管理

项目简介

旅游网站后台管理,包括如下

用户



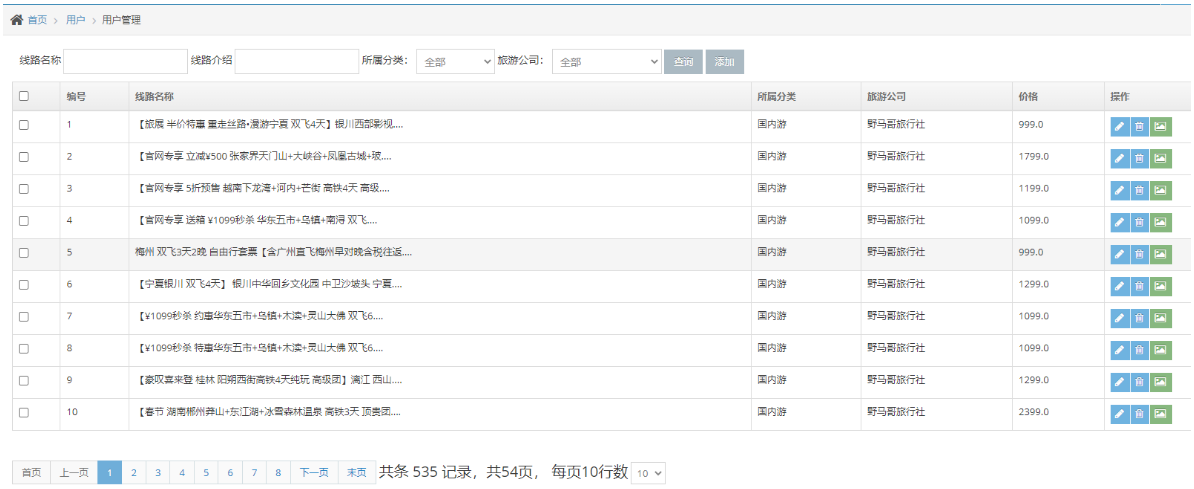
首页 › 用户 › 用户管理

姓名 ______ 邮箱 ______ 查询 添加 删除

☐	姓名	性别	出生日期	电话	邮箱	操作
☐	马龙12	女	2019-06-19	010-1234567	malong@163.com	✎ 🗑
☐	刘国梁	男	1977-10-26	13234342434	liuguoliang@qq.com	✎ 🗑
☐	张继科	男	2019-08-16	12345678909	zjk@qq.com	✎ 🗑
☐	樊振东	男	2019-08-07	13234342434	fzd@qq.com	✎ 🗑
☐	张三丰	男	2019-08-20	13234342434	zsf@163.com	✎ 🗑
☐	张无忌	男	2019-07-31	13234342434	zwj@163.com	✎ 🗑
☐	令狐冲	男	2019-08-15	13234342434	lhc@qq.com	✎ 🗑
☐	刘备	男	2019-08-24	13234342434	lb@qq.com	✎ 🗑
☐	b4	男	2019-09-30	010-1234234		✎ 🗑
☐	asd					✎ 🗑

首页 上一页 1 2 下一页 末页 共条 20 记录，共2页， 每页10行数 10

旅游线路

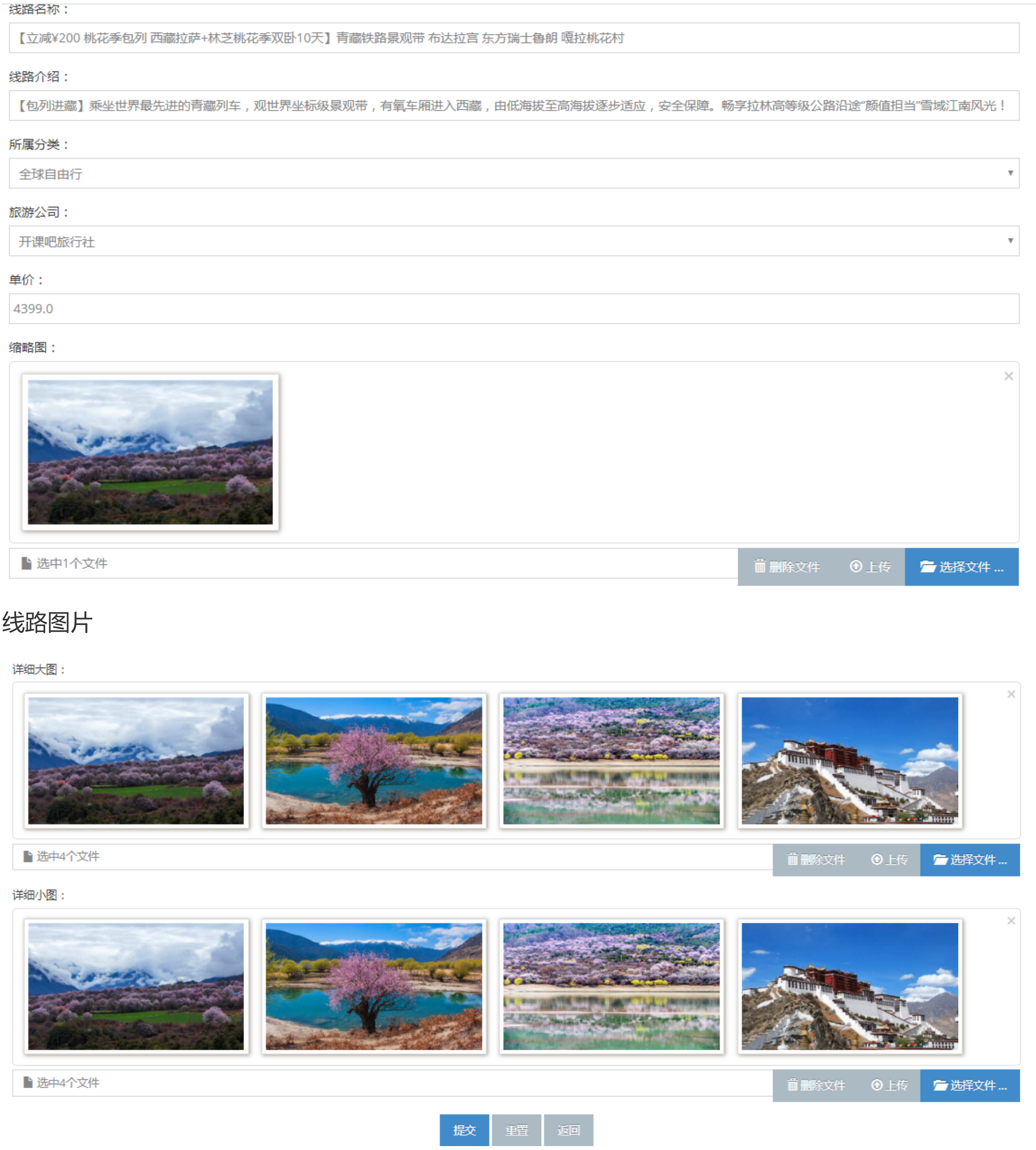


首页 › 用户 › 用户管理

线路名称 ______ 线路介绍 ______ 所属分类： 全部 旅游公司： 全部 查询 添加

☐	编号	线路名称	所属分类	旅游公司	价格	操作
☐	1	【旅展 半价特惠 重走丝路•漫游宁夏 双飞4天】银川西部影视....	国内游	野马哥旅行社	999.0	✎ 🗑 🖼
☐	2	【官网专享 立减¥500 张家界天门山+大峡谷+凤凰古城+玻....	国内游	野马哥旅行社	1799.0	✎ 🗑 🖼
☐	3	【官网专享 5折预售 越南下龙湾+河内+芒街 高铁4天 高级....	国内游	野马哥旅行社	1199.0	✎ 🗑 🖼
☐	4	【官网专享 送箱 ¥1099秒杀 华东五市+乌镇+南浔 双飞....	国内游	野马哥旅行社	1099.0	✎ 🗑 🖼
☐	5	梅州 双飞3天2晚 自由行套票【含广州直飞梅州早对晚含税往返....	国内游	野马哥旅行社	999.0	✎ 🗑 🖼
☐	6	【宁夏银川 双飞4天】 银川中华回乡文化园 中卫沙坡头 宁夏....	国内游	野马哥旅行社	1299.0	✎ 🗑 🖼
☐	7	【¥1099秒杀 约惠华东五市+乌镇+木渎+灵山大佛 双飞6....	国内游	野马哥旅行社	1099.0	✎ 🗑 🖼
☐	8	【¥1099秒杀 特惠华东五市+乌镇+木渎+灵山大佛 双飞6....	国内游	野马哥旅行社	1099.0	✎ 🗑 🖼
☐	9	【豪叹喜来登 桂林 阳朔西街高铁4天纯玩 高级团】漓江 西山....	国内游	野马哥旅行社	1299.0	✎ 🗑 🖼
☐	10	【春节 湖南郴州莽山+东江湖+冰雪森林温泉 高铁3天 顶贵团....	国内游	野马哥旅行社	2399.0	✎ 🗑 🖼

首页 上一页 1 2 3 4 5 6 7 8 下一页 末页 共条 535 记录，共54页， 每页10行数 10



线路名称：

【立减¥200 桃花季包列 西藏拉萨+林芝桃花季双卧10天】青藏铁路景观带 布达拉宫 东方瑞士鲁朗 嘎拉桃花村

线路介绍：

【包列进藏】乘坐世界最先进的青藏列车，观世界坐标级景观带，有氧车厢进入西藏，由低海拔至高海拔逐步适应，安全保障。畅享拉林高等级公路沿途"颜值担当"雪域江南风光！

所属分类：

全球自由行

旅游公司：

开课吧旅行社

单价：

4399.0

缩略图：

选中1个文件

线路图片

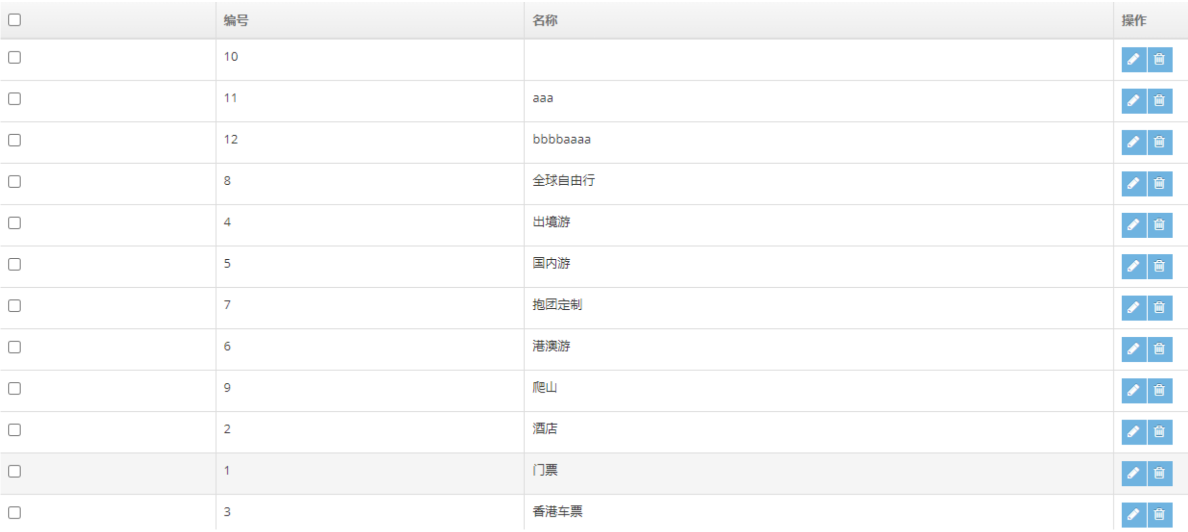
详细大图：

选中4个文件

详细小图：

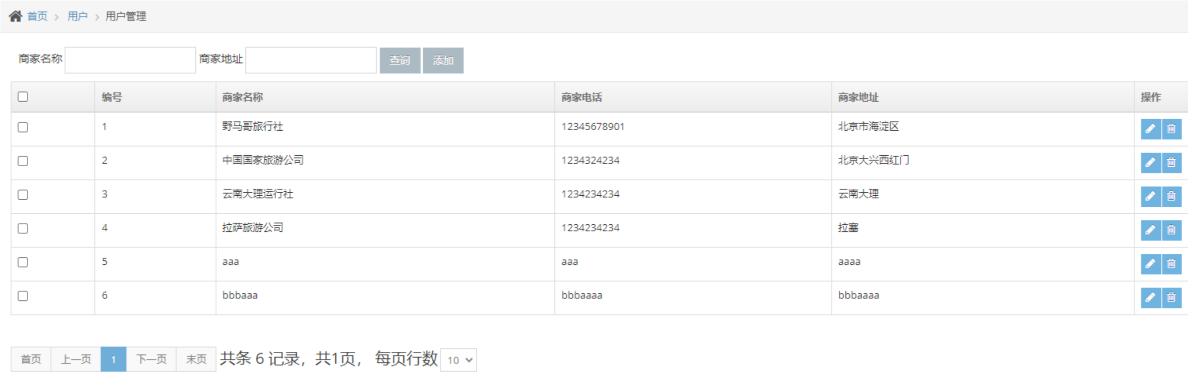
选中4个文件

线路分类



□	编号	名称	操作
□	10		
□	11	aaa	
□	12	bbbbaaaa	
□	8	全球自由行	
□	4	出境游	
□	5	国内游	
□	7	抱团定制	
□	6	港澳游	
□	9	爬山	
□	2	酒店	
□	1	门票	
□	3	香港车票	

旅行社



首页 › 用户 › 用户管理

商家名称 商家地址 查询 添加

□	编号	商家名称	商家电话	商家地址	操作
□	1	野马哥旅行社	12345678901	北京市海淀区	✎ 🗑
□	2	中国国家旅游公司	1234324234	北京大兴西红门	✎ 🗑
□	3	云南大理运行社	1234234234	云南大理	✎ 🗑
□	4	拉萨旅游公司	1234234234	拉塞	✎ 🗑
□	5	aaa	aaa	aaaa	✎ 🗑
□	6	bbbaaa	bbbaaaa	bbbaaaa	✎ 🗑

首页 上一页 1 下一页 末页 共条 6 记录，共1页， 每页行数 10

后台技术

springboot、mybatis、mybatis plus

> 前台

bootstrap、jquery、thymeleaf、fileinput插件

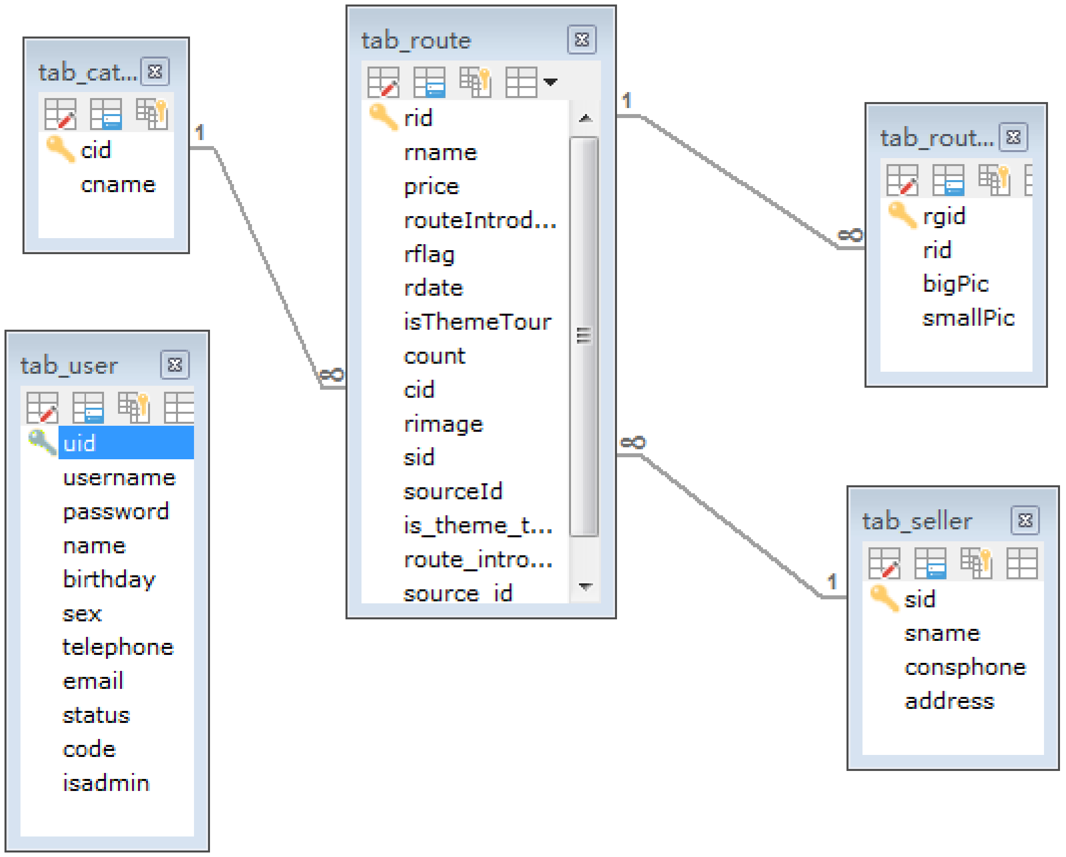
> 数据库

mysql

> 数据库设计

从素材中直接引入数据库脚本即可

表结构



)

创建项目

> 依赖

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>cn.yemage</groupId>
    <artifactId>travel-springboot-mp</artifactId>
    <version>1.0-SNAPSHOT</version>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.3.4.RELEASE</version>
        <relativePath/> <!-- 依赖spring-boot-starter-parent构件时，从本地仓库-> 远程仓库获取,不从本地目录获取 -->
    </parent>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
        <java.version>1.8</java.version>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-jdbc</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-thymeleaf</artifactId>
        </dependency>
        <!--mybatis pulus-->
        <dependency>
            <groupId>com.baomidou</groupId>
            <artifactId>mybatis-plus-boot-starter</artifactId>
            <version>3.3.2</version>
        </dependency>

        <dependency>
            <groupId>com.github.pagehelper</groupId>
            <artifactId>pagehelper</artifactId>
            <version>5.1.11</version>
        </dependency>
        <dependency>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
            <scope>provided</scope>
        </dependency>

        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>5.1.46</version>
        </dependency>

        <!--fileupload-->
        <dependency>
            <groupId>commons-fileupload</groupId>
            <artifactId>commons-fileupload</artifactId>
            <version>1.3.3</version>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>

    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>


</project>
```

> 启动器

```
@SpringBootApplication
public class TravelApplication {

    public static void main(String[] args) {

        SpringApplication.run(TravelApplication.class, args);
    }
}
```

> application.yml

具体语法参考YAML快速入门.html

```
server:
  port: 80
  servlet:
    context-path: / #项目的上下文路径
spring:
  datasource:
    url: jdbc:mysql://localhost:3306/travel
    username: root
    password: 123456
    driver-class-name: com.mysql.jdbc.Driver
    hikari:
      idle-timeout: 60000
      maximum-pool-size: 30
      minimum-idle: 10
  thymeleaf:
    cache: false
# mybatis plus配置
mybatis-plus:
  mapper-locations: classpath:/mybatis/*.xml #加载映射文件
  type-aliases-package: com.yemage.travel.domain #别名搜索的包
  configuration:
    lazy-loading-enabled: true #打开懒加载
    aggressive-lazy-loading: false #关闭积极懒加载
```

> 整合mybatis plus

```
@Configuration
@MapperScan("com.yemage.travel.dao")
public class MybatisPlusConfig {

    @Bean
    public PageInterceptor pageInterceptor() {
        return new PageInterceptor();
    }

}
```

用户管理

> 实体类

Lombok能以简单的注解形式来简化java代码，提高开发人员的开发效率

使用lomhok简化实体类开发，具体lomhok语法和安装参考资料lomhok入门.html文档

```
@Data
@TableName("tab_user")
public class User implements Serializable {

    @TableId(type = IdType.AUTO)
    private Integer uid;
    private String username;
    private String password;
    private String name;
    private Date birthday;
    private String sex;
    private String telephone;
    private String email;
    private String status;
    private String code;
    private Boolean isadmin;
}
```

> dao

接口

```
public interface UserDao extends BaseMapper<User> {

}
```

注意：这里使用mybatis plus内置方法，没有映射文件

> service

接口

```
public interface UserService {

    /**
     * 分页查询
     * @param conditioin 查询条件
     * @return
     */
    public PageInfo<User> findPage(User conditioin, int pageNum, int pageSize);

    /**
     * 查询
     * @param conditioin 查询条件
     * @return
     */
    public List<User> find(User conditioin);

    /**
     * 添加
     * @param user
     * @return
     */
    public int add(User user);

    /**
     * 根据ID查询用户
     * @param id
     * @return
     */
    public User findById(Integer id);

    /**
     * 修改
     * @param user
     * @return
     */
    public int update(User user);

    /**
     * 删除
     * @param id
     * @return
     */
    public int delete(Integer id);

}
```

实现类

```
@Service
public class UserServiceImpl implements UserService {

    @Autowired
    private UserDao userDao;

    @Override
    public PageInfo<User> findPage(User conditioin, int pageNum, int pageSize) {
        return PageHelper.startPage(pageNum, pageSize).doSelectPageInfo(() -> {
           userDao.selectList(Wrappers.<User>query());
        });
    }

    @Override
    public List<User> find(User condition) {
        return userDao.selectList(Wrappers.query());
    }

    @Override
    public int add(User user) {
        return userDao.insert(user);
    }

    @Override
    public User findById(Integer id) {
        return userDao.selectById(id);
    }

    @Override
    public int update(User user) {
        return userDao.updateById(user);
    }

    @Override
    public int delete(Integer id) {
        return userDao.deleteById(id);
    }
}
```

注意：分页方法采用的lambda表达式的写法

> 测试

```
@RunWith(SpringRunner.class)
@SpringBootTest
public class UserServiceImplTest {

    @Autowired
    private UserService userService;

    @Test
    public void testFindPage() {
        User u = new User();
        PageInfo<User> page = userService.findPage(u, 1, 10);
        page.getList().forEach(System.out :: println);

        System.out.println("总行数=" + page.getTotal());
        System.out.println("当前页=" + page.getPageNum());
        System.out.println("每页行数=" + page.getPageSize());
        System.out.println("总页数=" + page.getPages());
        System.out.println("起始行数=" + page.getStartRow());

        System.out.println("是第一页=" + page.isIsFirstPage());
        System.out.println("是最后页=" + page.isIsLastPage());

        System.out.println("还有下一页=" + page.isHasNextPage());
        System.out.println("还有上一页=" + page.isHasPreviousPage());
        System.out.println("页码列表" +
Arrays.toString(page.getNavigatepageNums()));

    }

}
```

> controller

```
@Controller
@RequestMapping("/admin/user")
public class UserController {

    @Autowired
    private UserService userService;

    /**
     * 分页查询
     * @param user 请求参数
     * @param pageNum 当前页
     * @param pageSize 每页行数
     * @param model
     * @return
     */
    @RequestMapping("/page")
    public String page(
            User user,
            @RequestParam(defaultValue = "1") Integer pageNum,
            @RequestParam(defaultValue = "10") Integer pageSize,
            Model model) {
        PageInfo<User> page = userService.findPage(user, pageNum, pageSize);
        model.addAttribute("page", page);

        return "/user/list";
    }

    /**
     * 跳到添加页面
     * @return
     */
    @RequestMapping("/toadd")
    public String toAdd() {
        return "/user/add";
    }

    /**
     * 添加
     * @param user
     * @return
     */
    @RequestMapping("/doadd")
    public String doAdd(User user) {
        userService.add(user);
        return "redirect:/admin/user/page";
    }

    /**
     * 跳到修改页面
     * @param id
     * @param model
     * @return
     */
    @RequestMapping("/toupdate/{id}")
    public String toUpdate(@PathVariable("id") Integer id, Model model) {
        User user = userService.findById(id);
        model.addAttribute("user", user);
        return "/user/update";
    }

    @RequestMapping("/doupdate")
    public String doUpdate(User user) {
        userService.update(user);
        return "redirect:/admin/user/page";
    }

    /**
     * 删除
     * @param id
     * @return
     */
    @RequestMapping("/delete/{id}")
    public String delete(@PathVariable("id") Integer id) {
        userService.delete(id);
        return "redirect:/admin/user/page";
    }

}
```

> 页面

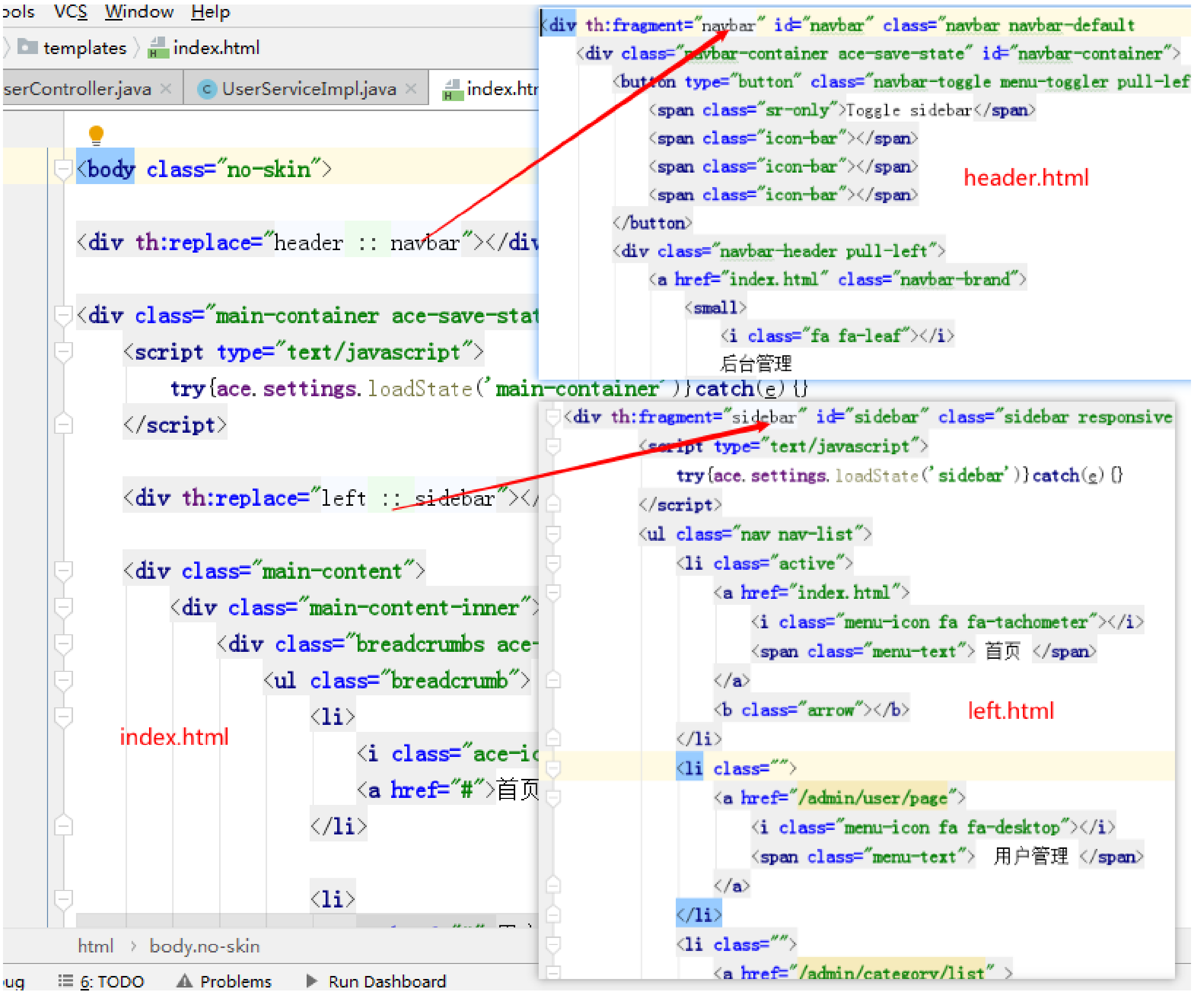
前端技术栈 bootstrap，jquery，ace, fileinput插件

拷贝资料中的static目录，里面有bootstrap, jquery, ace，fileinput前端框架

> index.html、header.html、left.html(拷贝)

springboot 项目访问http://localhost/context_path回自动跳转到templates/index.html

这里我们直接拷贝index.html,header.html,和left.html



```
<body class="no-skin">

<div th:replace="header :: navbar"></div

<div class="main-container ace-save-stat
    <script type="text/javascript">
        try{ace.settings.loadState('main-container')}catch(e){}
    </script>

    <div th:replace="left :: sidebar"></

    <div class="main-content">
        <div class="main-content-inner">
            <div class="breadcrumbs ace-
                <ul class="breadcrumb">
                    <li>
                        <i class="ace-ic
                        <a href="#">首页
                    </li>

                    <li>
```

```
<div th:fragment="navbar" id="navbar" class="navbar navbar-default
    <div class="navbar-container ace-save-state" id="navbar-container">
        <button type="button" class="navbar-toggle menu-toggler pull-lef
            <span class="sr-only">Toggle sidebar</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
        </button>
        <div class="navbar-header pull-left">
            <a href="index.html" class="navbar-brand">
                <small>
                    <i class="fa fa-leaf"></i>
                    后台管理
```

```
<div th:fragment="sidebar" id="sidebar" class="sidebar responsive
        <script type="text/javascript">
            try{ace.settings.loadState('sidebar')}catch(e){}
        </script>
        <ul class="nav nav-list">
            <li class="active">
                <a href="index.html">
                    <i class="menu-icon fa fa-tachometer"></i>
                    <span class="menu-text"> 首页 </span>
                </a>
                <b class="arrow"></b>
            </li>
            <li class="">
                <a href="/admin/user/page">
                    <i class="menu-icon fa fa-desktop"></i>
                    <span class="menu-text">  用户管理 </span>
                </a>
            </li>
            <li class="">
                <a href="/admin/category/list" >
```

> list.html

页面涉及到ace和bootstrap模板部分代码拷贝，代码中thyleaf指令和脚本部分自己写

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"/>
    <meta charset="utf-8"/>
    <title>用户管理</title>

    <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0"/>

    <!-- bootstrap & fontawesome -->
    <link rel="stylesheet" href="/assets/css/bootstrap.min.css"/>
    <link rel="stylesheet" href="/assets/font-awesome/4.5.0/css/font-
awesome.min.css"/>

    <!-- page specific plugin styles -->

    <!-- text fonts -->
    <link rel="stylesheet" href="/assets/css/fonts.googleapis.com.css"/>

    <!-- ace styles -->
    <link rel="stylesheet" href="/assets/css/ace.min.css" class="ace-main-
stylesheet" id="main-ace-style"/>
```

```
    <!--[if lte IE 9]>
    <link rel="stylesheet" href="/assets/css/ace-part2.min.css" class="ace-main-stylesheet"/>
    <![endif]-->
    <link rel="stylesheet" href="/assets/css/ace-skins.min.css"/>
    <link rel="stylesheet" href="/assets/css/ace-rtl.min.css"/>

    <!--[if lte IE 9]>
    <link rel="stylesheet" href="/assets/css/ace-ie.min.css"/>
    <![endif]-->

    <!-- inline styles related to this page -->

    <!-- ace settings handler -->
    <script src="/assets/js/ace-extra.min.js"></script>

    <!-- HTML5shiv and Respond.js for IE8 to support HTML5 elements and media queries -->

    <!--[if lte IE 8]>
    <script src="/assets/js/html5shiv.min.js"></script>
    <script src="/assets/js/respond.min.js"></script>
    <![endif]-->

    <!--[if !IE]> -->
    <script src="/assets/js/jquery-2.1.4.min.js"></script>

    <!-- <![endif]-->

    <!--[if IE]>
    <script src="/assets/js/jquery-1.11.3.min.js"></script>
    <![endif]-->
    <script src="/assets/js/bootstrap.min.js"></script>

    <!-- page specific plugin scripts -->
    <script src="/assets/js/jquery.dataTables.min.js"></script>
    <script src="/assets/js/jquery.dataTables.bootstrap.min.js"></script>
    <script src="/assets/js/dataTables.buttons.min.js"></script>
    <script src="/assets/js/buttons.flash.min.js"></script>
    <script src="/assets/js/buttons.html5.min.js"></script>
    <script src="/assets/js/buttons.print.min.js"></script>
    <script src="/assets/js/buttons.colVis.min.js"></script>
    <script src="/assets/js/dataTables.select.min.js"></script>

    <!-- ace scripts -->
    <script src="/assets/js/ace-elements.min.js"></script>
    <script src="/assets/js/ace.min.js"></script>

    <script>
    $(function(){
        $("#checkAll").click(function() {
            $(":checkbox[name='ids']").prop("checked", $(this).prop("checked"));
        });
        $("#btnDelete").click(function(){
            if ($(":checked[name='ids']").length > 0) {
                $("#df").submit();
            } else {
                alert('请选择要删除的记录')
            }
        });
    })
</script>

</head>

<body class="no-skin">

<div th:replace="header :: navbar"></div>

<div class="main-container ace-save-state" id="main-container">
    <script type="text/javascript">
        try {
            ace.settings.loadState('main-container')
        } catch (e) {
        }
    </script>

    <div th:replace="left :: sidebar"></div>

    <div class="main-content">
        <div class="main-content-inner">
            <div class="breadcrumbs ace-save-state" id="breadcrumbs">
                <ul class="breadcrumb">
                    <li>
                        <i class="ace-icon fa fa-home home-icon"></i>
                        <a href="#">首页</a>
                    </li>
                    <li>
                        <a href="#">用户</a>
                    </li>
                    <li class="active">用户管理</li>
                </ul><!-- /.breadcrumb -->

            </div>

            <div class="page-content">

                <div style="float: left; padding: 10px;">
                    <form class="form-inline" id="qf" action="/admin/user/page"
method="post">
                        <input type="hidden" name="pageNum" id="pageNum"
th:value="${page.pageNum}">
                        <input type="hidden" name="pageSize" id="pageSize"
th:value="${page.pageSize}">

                        <div class="form-group">
                            <label for="name">姓名</label>
                            <input type="text" name="name" class="form-control"
id="name" th:value="${param.name}">
                        </div>

                        <div class="form-group">
                            <label for="email">邮箱</label>
                            <input type="text" name="email" class="form-control"
id="email" th:value="${param.email}">
                        </div>
                        <button type="submit" class="btn btn-sm btn-default">查询
</button>
                        <button type="button" class="btn btn-sm btn-default"
                                onclick="location.href='/admin/user/toadd'">添加
                        </button>
                        <button type="button" class="btn btn-sm btn-default"
id="btnDelete">删除</button>
                    </form>
                </div>

                <form action="/admin/user/delete" method="post" id="df">
                <table id="simple-table" class="table  table-bordered table-
hover">
                    <thead>
                    <tr>
                        <th><input type="checkbox" id="checkAll"></th>
                        <th>姓名</th>
                        <th>性别</th>
                        <th>出生日期</th>
                        <th>电话</th>
                        <th>邮箱</th>
                        <th>操作</th>
                    </tr>
                    </thead>
                    <tbody>

                    <tr th:each="u, status : ${page.list}"
th:style="${status.even} ? 'background-color: grey'">
                        <td><input type="checkbox" name="ids"
th:value="${u.uid}"></td>
                        <td th:text="${u.name}"></td>
                        <td th:text="${u.sex}"></td>
                        <td th:text="${#dates.format(u.birthday, 'yyyy-MM-
dd')}"></td>
                        <td th:text="${u.telephone}"></td>
                        <td th:text="${u.email}"></td>
                        <td>
                            <div class="hidden-sm hidden-xs btn-group">
                                <a class="btn btn-xs btn-info"

 th:href="|/admin/user/toupdate/${u.uid}|" >
                                    <i class="ace-icon fa fa-pencil bigger-120">
</i>
                                </a>

                                <a class="btn btn-xs btn-info"
                                        th:href="|/admin/user/delete/${u.uid}|">
                                    <i class="ace-icon fa fa-trash-o bigger-
120"></i>
                                </a>
                            </div>
                        </td>
                    </tr>

                    </tbody>
                </table>
                </form>
                <div>
                    <nav aria-label="Page navigation">
                        <ul class="pagination">
                            <li id="first">
                                <a href="javascript:void(0);">
                                    <span aria-hidden="true">首页</span>
                                </a>
                            </li>
                            <li id="prev">
                                <a href="javascript:void(0);" aria-label="Previous">
                                    <span aria-hidden="true">上一页</span>
                                </a>
                            </li>
                            <li name="pageNum" th:each="i : ${page.navigatepageNums}" th:class="${i == page.pageNum} ? 'active'">
                                <a href="javascript:void(0);" th:text="${i}"></a>
                            </li>

                            <li id="next">
                                <a href="javascript:void(0);" aria-label="Next">
                                    <span aria-hidden="true">下一页</span>
                                </a>
                            </li>

                            <li id="last">
                                <a href="javascript:void(0);">
                                    <span aria-hidden="true">末页</span>
                                </a>
                            </li>

                            <span style="font-size: 20px;margin-left: 5px;"
                                  th:text="|共条 ${page.total} 记录，共 ${page.pages}页， 每页${page.pageSize}行数|">
                                            </span>
                            <select id="setRows">
                                <option value="5" th:selected="${5 == page.pageSize}">5</option>
                                <option value="10" th:selected="${10 == page.pageSize}" >10</option>
                                <option value="20" th:selected="${20 == page.pageSize}" >20</option>
                                <option value="30" th:selected="${30 == page.pageSize}" >30</option>
                            </select>

                        </ul>
                    </nav>
                </div>

                <script th:inline="javascript">
                    //得到初始化变量
                    var pageNum = [[${page.pageNum}]]; //当前页
                    var pages =[[${page.pages}]]; //总页数
                    var hasNextPage = [[${page.hasNextPage}]];//true：还有下一页
                    var hasPreviousPage =[[${page.hasPreviousPage}]]; //还有上一页
                    //判断按钮状态
                    if (!hasPreviousPage) {
                        $("#prev").addClass("disabled");
                        $("#first").addClass("disabled");
                    }
                    if (!hasNextPage) {
                        $("#next").addClass("disabled");
                        $("#last").addClass("disabled");
                    }

                    //设置按钮的监听事件
                    $("#first").click(function() {
                        if (!$("#first").hasClass("disabled")) {
                            $("#pageNum").val(1);
                            $("#qf").submit();
                        }
                    });
                    $("#prev").click(function() {
                        if (!$("#prev").hasClass("disabled")) {
                            $("#pageNum").val(pageNum - 1);
                            $("#qf").submit();
                        }
                    });
                    $("#next").click(function() {
                        if (!$("#next").hasClass("disabled")) {
                            $("#pageNum").val(pageNum + 1);
                            $("#qf").submit();
                        }
                    });
                    $("#last").click(function() {
                        if (!$("#last").hasClass("disabled")) {
                            $("#pageNum").val(pages);
                            $("#qf").submit();
                        }
                    });

                    //页码分页
                    $("li[name='pageNum']").click(function(){
                        if (!$(this).hasClass("active")) {
                            $("#pageNum").val($(this).children("a").html());
                            $("#qf").submit();
                        }
                    });

                    //设置每页行数
                    $("#setRows").change(function(){
                        $("#pageSize").val($(this).val());
                        $("#pageNum").val(1);
                        $("#qf").submit();
                    });


                </script>

            </div><!-- /.page-content -->
        </div>
    </div><!-- /.main-content -->
```

```
</div><!-- /.main-container -->

</body>
</html>
```

> add.html

```
<div class="page-content">

    <form action="/admin/user/doadd" method="post">
        <div class="form-group">
            <label for="name">姓名：</label>
            <input type="text" class="form-control" id="name" name="name"
placeholder="请输入姓名">
        </div>

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

        <div class="form-group">
            <label for="birthday">生日：</label>
            <input type="text" class="form-control" id="birthday"
name="birthday" placeholder="请输入生日"
                   onClick="WdatePicker({el:this,dateFmt:'yyyy-MM-dd'})">
        </div>

        <div class="form-group">
            <label for="telephone">电话：</label>
            <input type="text" class="form-control" name="telephone"
id="telephone" placeholder="请输入电话"/>
        </div>

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

        <div class="form-group">
            <label for="username">登录名</label>
            <input type="text" class="form-control" name="username"
id="username" placeholder="请输入登录名"/>
        </div>

        <div class="form-group">
            <label for="password">密码</label>
            <input type="text" class="form-control" name="password"
id="password" placeholder="请输入密码"/>
        </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><!-- /.page-content -->
```

> update.html

```
<div class="page-content">

    <form action="/admin/user/doupdate" method="post" th:object="${user}">
        <input type="hidden" name="uid" th:value="${user.uid}">
        <div class="form-group">
            <label for="name">姓名：</label>
            <input type="text" class="form-control" id="name" name="name"
placeholder="请输入姓名" th:value="*{name}">
        </div>

        <div class="form-group">
            <label>性别：</label>
            <input type="radio" name="sex" value="男" th:checked="'男' == *{sex}"
/>男
            <input type="radio" name="sex" value="女" th:checked="'女' == *{sex}"
/>女
        </div>

        <div class="form-group">
            <label for="birthday">生日：</label>
            <input type="text" class="form-control" id="birthday"
name="birthday" placeholder="请输入生日"
onClick="WdatePicker({el:this,dateFmt:'yyyy-MM-dd'})" th:value="*{birthday}">
        </div>

        <div class="form-group">
            <label for="telephone">电话：</label>
            <input type="text" class="form-control" name="telephone"
id="telephone" placeholder="请输入电话" th:value="*{telephone}"/>
        </div>

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

        <div class="form-group">
            <label for="username">登录名</label>
            <input type="text" class="form-control" name="username"
id="username" placeholder="请输入登录名" th:value="*{username}"/>
        </div>

        <div class="form-group">
            <label for="password">密码</label>
            <input type="text" class="form-control" name="password"
id="password" placeholder="请输入密码" th:value="*{password}"/>
        </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><!-- /.page-content -->
```

> 多选删除

controller

```
/**
 * 批量删除
 * @param ids
 * @return
 */
@RequestMapping("/delete")
public String batchDelete(@RequestParam("ids") Integer[] ids) {
    for (Integer id : ids) {
        userService.delete(id);
    }
    return "redirect:/admin/user/page";
}
```

页面

```
<script>
    $(function(){
        $("#checkAll").click(function() {
            $(":checkbox[name='ids']").prop("checked", $(this).prop("checked"));
        });
        $("#btnDelete").click(function(){
            if ($(":checked[name='ids']").length > 0) {
                $("#df").submit();
            } else {
                alert('请选择要删除的记录')
            }
        });
    })
</script>
```

> 注入在springmvc中配置日期转换器

```
@Component
public class DateConverterConfig implements Converter<String, Date> {

    private static final List<String> formarts = new ArrayList<>(4);
    static{
        formarts.add("yyyy-MM");
        formarts.add("yyyy-MM-dd");
        formarts.add("yyyy-MM-dd hh:mm");
        formarts.add("yyyy-MM-dd hh:mm:ss");
    }
    @Override
    public Date convert(String source) {
        String value = source.trim();
        if ("".equals(value)) {
            return null;
        }
        if(source.matches("^\\d{4}-\\d{1,2}$")){
            return parseDate(source, formarts.get(0));
        }else if(source.matches("^\\d{4}-\\d{1,2}-\\d{1,2}$")){
            return parseDate(source, formarts.get(1));
        }else if(source.matches("^\\d{4}-\\d{1,2}-\\d{1,2}
{1}\\d{1,2}:\\d{1,2}$")){
            return parseDate(source, formarts.get(2));
        }else if(source.matches("^\\d{4}-\\d{1,2}-\\d{1,2}
{1}\\d{1,2}:\\d{1,2}:\\d{1,2}$")){
            return parseDate(source, formarts.get(3));
        }else {
            throw new IllegalArgumentException("Invalid boolean value '" +
source + "'");
        }
    }

    /**
     * 格式化日期
     * @param dateStr String 字符型日期
     * @param format String 格式
     * @return Date 日期
     */
    public  Date parseDate(String dateStr, String format) {
        Date date=null;
        try {
            DateFormat dateFormat = new SimpleDateFormat(format);
            date = dateFormat.parse(dateStr);
        } catch (Exception e) {

        }
        return date;
    }

}
```

旅游公司

> 参考用户管理CRUD操作，具体代码和用户管理相似

> 实体类

```java
@Data
@TableName("tab_seller")
public class Seller implements Serializable {

    @TableId(type = IdType.AUTO)
    private Integer sid;//商家id
    private String sname;//商家名称
    private String consphone;//商家电话
    private String address;//商家地址

}
```

> dao

```java
public interface SellerDao extends BaseMapper<Seller> {

}
```

> service

接口

```java
public interface SellerService {

    /**
     * 分页查询
     * @param conditioin 查询条件
     * @return
     */
    public PageInfo<Seller> findPage(Seller conditioin, int pageNum, int 
pageSize);

    /**
     * 查询
     * @param conditioin 查询条件
     * @return
     */
    public List<Seller> find(Seller conditioin);

    /**
     * 添加
     * @param seller
     * @return
     */
    public int add(Seller seller);

    /**
     * 根据ID查询用户
     * @param id
     * @return
     */
    public Seller findById(Integer id);

    /**
     * 修改
     * @param seller
     * @return
     */
    public int update(Seller seller);

    /**
     * 删除
     * @param id
     * @return
     */
    public int delete(Integer id);

}
```

实现类

```
@Service
public class SellerServiceImpl implements SellerService {

    @Autowired
    private SellerDao sellerDao;

    @Override
    public PageInfo<Seller> findPage(Seller conditioin, int pageNum, int
pageSize) {
        return PageHelper.startPage(pageNum, pageSize).doSelectPageInfo(() -> {
            sellerDao.selectList(new QueryWrapper<Seller>());
        });
    }

    @Override
    public List<Seller> find(Seller condition) {
        return sellerDao.selectList(Wrappers.query());
    }

    @Override
    public int add(Seller seller) {
        return sellerDao.insert(seller);
    }

    @Override
    public Seller findById(Integer id) {
        return sellerDao.selectById(id);
    }

    @Override
    public int update(Seller seller) {
        return sellerDao.updateById(seller);
    }

    @Override
    public int delete(Integer id) {
        return sellerDao.deleteById(id);
    }

}
```

> controller

```java
@Controller
@RequestMapping("/admin/seller/")
public class SellerController {

    @Autowired
    private SellerService sellerService;

    @RequestMapping("/page")
    public String page(
            Seller seller,
            @RequestParam(defaultValue = "1") Integer pageNum,
            @RequestParam(defaultValue = "10") Integer pageSize,
            Model model) {
        PageInfo<Seller> page = sellerService.findPage(seller, pageNum, pageSize);
        model.addAttribute("page", page);

        return "seller/list";
    }

    @RequestMapping("/toadd")
    public String toAdd() {
        return "seller/add";
    }

    @RequestMapping("/doadd")
    public String doAdd(Seller seller) {
        sellerService.add(seller);
        return "redirect:/admin/seller/page";
    }

    @RequestMapping("toupdate")
    public String toUpdate(Integer id, Model model) {
        Seller seller = sellerService.findById(id);
        model.addAttribute("seller", seller);
        return "seller/update";
    }

    @RequestMapping("/doupdate")
    public String doUpdate(Seller seller) {
        sellerService.update(seller);
        return "redirect:/admin/seller/page";
    }

    @RequestMapping("/delete")
    public String delete(Integer id) {
        sellerService.delete(id);
        return "redirect:/admin/seller/page";
    }

}
```

> 页面

页面代码和用户crud基本一致，直接从素材拷贝即可

线路分类

> 实体类

```
@Data
@TableName("tab_category")
public class Category implements Serializable {

    @TableId(type = IdType.AUTO)
    private Integer cid;//分类id
    private String cname;//分类名称

}
```

> dao

```
public interface CategoryDao extends BaseMapper<Category> {


}
```

> service

接口

```
public interface CategoryService {

    public List<Category> find();

    /**
     * 添加
     * @param category
     * @return
     */
    public int add(Category category);

    /**
     * 根据ID查询
     * @param id
     * @return
     */
    public Category findById(Integer id);

    /**
     * 修改
     * @param category
     * @return
     */
    public int update(Category category);

    /**
     * 删除
     * @param id
     * @return
     */
    public int delete(Integer id);
}
```

实现类

```
@Service
public class CategoryServiceImpl implements CategoryService {

    @Autowired
    private CategoryDao categoryDao;

    @Override
    public List<Category> find() {
        return categoryDao.selectList(Wrappers.query());
    }

    @Override
    public int add(Category category) {
        return categoryDao.insert(category);
    }

    @Override
    public Category findById(Integer id) {
        return categoryDao.selectById(id);
    }

    @Override
    public int update(Category category) {
        return categoryDao.updateById(category);
    }

    @Override
    public int delete(Integer id) {
        return categoryDao.deleteById(id);
    }
}
```

> controller

```
@Controller
@RequestMapping("/admin/category/")
public class CategoryController {

    @Autowired
    private CategoryService categoryService;

    @RequestMapping("/list")
    public String find(Model model) {
        List<Category> list = categoryService.find();
        model.addAttribute("list", list);
        return "category/list";
    }

    @RequestMapping("/toadd")
    public String toAdd() {
        return "category/add";
    }

    @RequestMapping("/doadd")
    public String doAdd(Category category) {
        categoryService.add(category);
        return "redirect:/admin/category/list";
    }

    @RequestMapping("/toupdate")
    public String toUpdate(Integer id, Model model) {
        Category category = categoryService.findById(id);
        model.addAttribute("category", category);
        return "category/update";
    }

    @RequestMapping("/doupdate")
    public String doUpdate(Category category) {
        categoryService.update(category);
        return "redirect:/admin/category/list";
    }

    @RequestMapping("delete")
    public String delete(Integer id) {
        categoryService.delete(id);
        return "redirect:/admin/category/list";
    }

}
```

> 页面

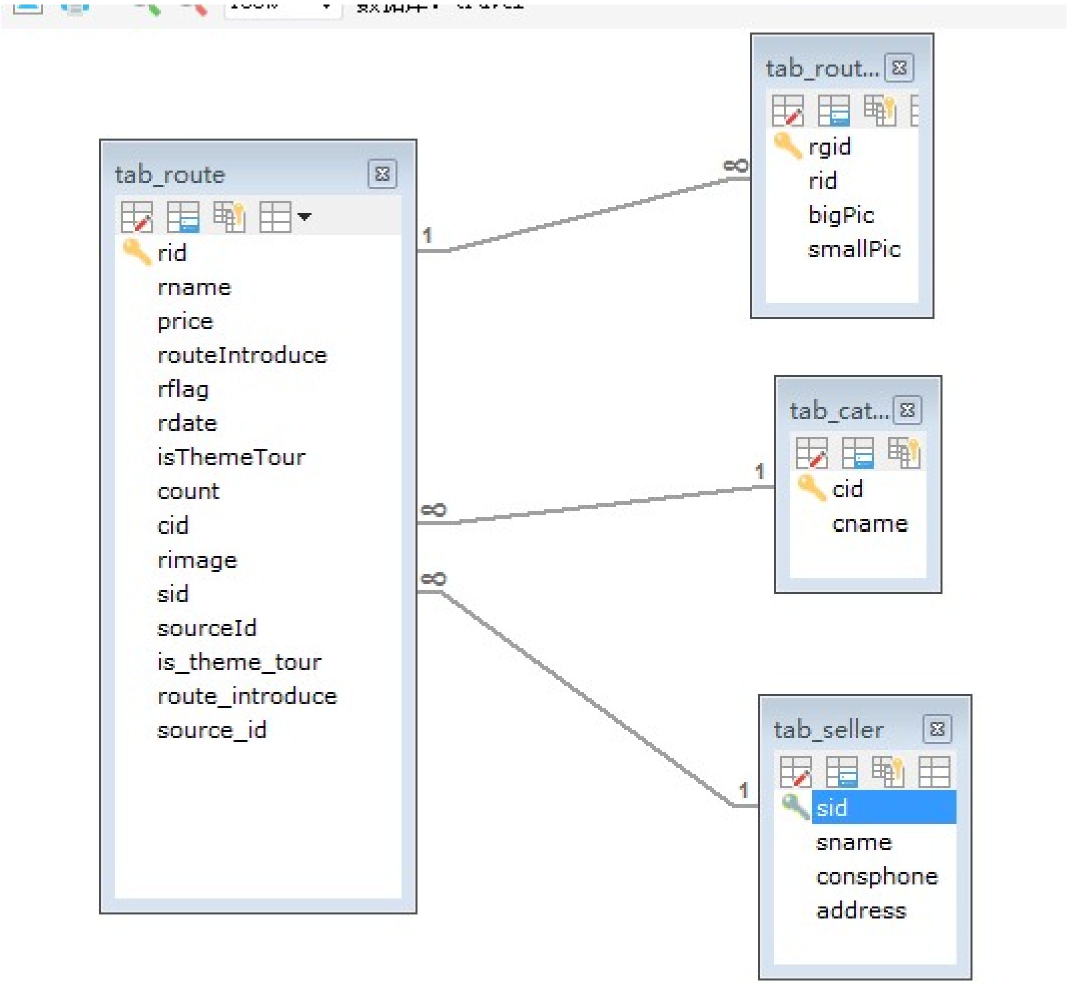
页面代码直接从素材拷贝即可，注意分类因为数据较少，可以不用分页

旅游线路

> 需求和表结构

旅游线路主表（线路名称，详情，介绍，价格，缩略图）

旅游线详细图片表(小图，大图)



> 实体类

线路实体类

```java
@Data
@TableName("tab_route")
public class Route implements Serializable {

    @TableId(type = IdType.AUTO)
    private Integer rid;//线路id，必输
    private String rname;//线路名称，必输
    private Double price;//价格，必输
    private String routeIntroduce;//线路介绍
    private String rflag;   //是否上架，必输，0代表没有上架，1代表是上架
    private String rdate;   //上架时间
    private String isThemeTour;//是否主题旅游，必输，0代表不是，1代表是
    private Integer count;//收藏数量
    private Integer cid;//所属分类，必输
    private String rimage;//缩略图
    private Integer sid;//所属商家
    private String sourceId;//抓取数据的来源id

    @TableField(exist = false)
    private Category category;//所属分类 使用resultmap的assocation处理
    @TableField(exist = false)
    private Seller seller;//所属商家 使用resultmap的assocation处理
    @TableField(exist = false)
    private List<RouteImg> routeImgList;//商品详情图片列表,关联属性，mybatis plus不能查，需要配置resultmap使用resultmap的collection处理

}
```

线路的图片列表

```
@Data
@TableName("tab_route_img")
public class RouteImg implements Serializable {

    @TableId(type = IdType.AUTO)
    private int rgid;//商品图片id
    private int rid;//旅游商品id
    private String bigpic;//详情商品大图
    private String smallpic;//详情商品小图

}
```

> dao

线路dao

```
public interface RouteDao extends BaseMapper<Route> {

    /**
     * 分页查询
     * @param conditioin
     * @return
     */
    public List<Route> find(Route conditioin);

    /**
     * 根据ID查询用户
     * @param id
     * @return
     */
    public Route findById(nteger id);

}
```

> 映射文件

因为线路需要查询线路详细图片列表`routeImgList`，线路所属分类`category`，线路所属旅行社`seller`,这里在xml映射文件中使用resultMap的association和collection查询，不能使用mybatis plus内置查询

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
        PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
        "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="cn.yemage.travel.dao.RouteDao">

    <sql id="selectSql">
        SELECT
            *
        FROM
            tab_route
    </sql>

    <select id="find" resultMap="routeMap">
        <include refid="selectSql"></include>
        <where>
            <if test="cid !=null">
                cid = #{cid}
            </if>
            <if test="sid !=null">
                and sid = #{sid}
            </if>
            <if test="rname !=null and rname.trim() != ''">
                and rname like '%${rname}%'
            </if>
            <if test="routeIntroduce != null and routeIntroduce.trim() != ''">
                and routeIntroduce like '%${routeIntroduce}%'
            </if>
        </where>
    </select>

    <resultMap id="routeMap" type="route">
        <id column="rid" property="rid"></id>
        <association property="category" javaType="category"
select="cn.yemage.travel.dao.CategoryDao.findById" column="cid">
            <id column="cid" property="cid"></id>
        </association>
        <association property="seller" javaType="seller"
select="cn.yemage.travel.dao.SellerDao.findById" column="sid">
            <id column="sid" property="sid"></id>
        </association>
        <collection property="routeImgList" javaType="java.util.List"
ofType="RouteImg" select="cn.yemage.travel.dao.RouteImgDao.findByRid"
column="rid">
            <id column="rgid" property="rgid"></id>
        </collection>
    </resultMap>

    <select id="findById" parameterType="int" resultMap="routeMap">
        <include refid="selectSql"></include>
        where
            rid = #{rid}
    </select>

</mapper>
```

注意：这里因为要查询旅游线路的所属分类（`select="cn.yemage.travel.dao.CategoryDao.findById"`），所属旅行社（`select="cn.yemage.travel.dao.SellerDao.findById"`），线路详细图片列表（`select="cn.yemage.travel.dao.RouteImgDao.findByRid"`）

所以需要分在在CategoryDao中增加findById方法如下

```
public interface CategoryDao extends BaseMapper<Category> {

    @Select("SELECT cname,cid FROM tab_category WHERE cid=#{id}")
    public Category findById(Integer id);

}
```

SellerDao增加findById方法如下：

```
public interface SellerDao extends BaseMapper<Seller> {

    @Select("SELECT * FROM tab_seller WHERE sid=#{id}")
    public Seller findById(Integer id);

}
```

RouteImgDao增加findByRid方法：如下

```
public interface RouteImgDao extends BaseMapper<RouteImg> {

    @Select("select * from tab_route_img where rid = #{rid}")
    public List<RouteImg> findByRid(Integer rid);

}
```

> service

线路service：接口

```
public interface RouteService {

    /**
     * 分页查询
     * @param conditioin 查询条件
     * @return
     */
    public PageInfo<Route> findPage(Route conditioin, int pageNum, int
pageSize);

    /**
     * 添加
     * @param route
     * @return
     */
    public int add(Route route);

    /**
     * 根据ID查询用户
     * @param id
     * @return
     */
    public Route findById(Integer id);

    /**
     * 修改
     * @param route
     * @return
     */
    public int update(Route route);

    /**
     * 删除
     * @param id
     * @return
     */
    public int delete(Integer id);

}
```

线路service实现类

```
@Service
public class RouteServiceImpl implements RouteService {

    @Autowired
    private RouteDao routeDao;

    @Override
    public PageInfo<Route> findPage(Route conditioin, int pageNum, int pageSize)
{
        return PageHelper.startPage(pageNum, pageSize).doSelectPageInfo(() -> {
            routeDao.find(conditioin);
        });
    }

    @Override
    public int add(Route route) {
        return routeDao.insert(route);
    }

    @Override
    public Route findById(Integer id) {
        return routeDao.findById(id);
    }

    @Override
    public int update(Route route) {
        return routeDao.updateById(route);
    }

    @Override
    public int delete(Integer id) {
        return routeDao.deleteById(id);
    }
}
```

> 测试

```
@RunWith(SpringRunner.class)
@SpringBootTest
public class RouteServiceImplTest {
    @Autowired
    private RouteService routeService;

    @Test
    public void findPage() {
        Route condition = new Route();
        condition.setRname("北京");
        PageInfo<Route> page = routeService.findPage(condition, 1, 10);

        page.getList().forEach((r) -> {
            System.out.println(r.getRid() + "\t" + r.getRname() + "\t" +
r.getCategory().getCname() + "\t" + r.getSeller().getSname() + "\t" +
r.getRouteImgList().size());
        });
    }

    @Test
    public void findById() {
        Route r = routeService.findById(34);
        System.out.println(r.getRid() + "\t" + r.getRname() + "\t" +
r.getCategory().getCname() + "\t" + r.getSeller().getSname() + "\t" +
r.getRouteImgList().size());
    }
}
```

> controller

```
@Controller
@RequestMapping("/admin/route/")
public class RouteController {

    @Autowired
    private RouteService routeService;
    @Autowired
    private CategoryService categoryService;
    @Autowired
    private SellerService sellerService;

    /**
     * 分页
     * @param route
     * @param pageNum
     * @param pageSize
     * @param model
     * @return
     */
    @RequestMapping("/page")
    public String page(
            Route route,
            @RequestParam(defaultValue = "1") Integer pageNum,
            @RequestParam(defaultValue = "10") Integer pageSize,
            Model model) {
        PageInfo<Route> page = routeService.findPage(route, pageNum, pageSize);
        model.addAttribute("page", page);

        //查询所有分类生成下拉框
        List<Category> categories = categoryService.find();
        model.addAttribute("categories", categories);
        List<Seller> sellers = sellerService.find(new Seller());
        model.addAttribute("sellers", sellers);

        //用于页面回显
        model.addAttribute("route", route);

        return "route/list";
    }

    /**
     * 删除
     * @param id
     * @return
     */
    @RequestMapping("/delete/{id}")
    public String delete(@PathVariable("id") Integer id) {
        routeService.delete(id);
        return "redirect:/admin/route/page";
    }

    /**
     * 跳到添加页面
     * @param model
     * @return
     */
    @RequestMapping("/toadd")
    public String toAdd(Model model) {
        //查询所有分类生成下拉框
        List<Category> categories = categoryService.find();
        model.addAttribute("categories", categories);
        List<Seller> sellers = sellerService.find(new Seller());
        model.addAttribute("sellers", sellers);

        return "route/add";
    }

    /**
     * 执行添加
     * @param route
     * @param rimageFile
     * @param request
     * @return
     * @throws IOException
     */
    @RequestMapping("/doadd")
    public String doAdd(Route route, @RequestParam("rimageFile") MultipartFile
rimageFile, HttpServletRequest request) throws IOException {
        routeService.add(route);
        return "redirect:/admin/route/page";
    }

    /**
     * 根据id查询，跳转到修改页面
     * @param id
     * @param model
     * @return
     */
    @RequestMapping("/toupdate/{id}")
    public String toUpdate(@PathVariable Integer id, Model model) {
        //查询所有分类生成下拉框
        List<Category> categories = categoryService.find();
        model.addAttribute("categories", categories);
        List<Seller> sellers = sellerService.find(new Seller());
        model.addAttribute("sellers", sellers);

        Route route = routeService.findById(id);
        model.addAttribute("route", route);

        return "route/update";
    }

    @RequestMapping("/doupdate")
    public String doUpdate(Route route, @RequestParam("rimageFile")
MultipartFile rimageFile, HttpServletRequest request) throws IOException {
        routeService.update(route);
        return "redirect:/admin/route/page";
    }

}
```

> 页面list.html、add.html、update.html

页面部分直接从素材拷贝，里面thymleaf指令和脚本自己写

```
参考素材代码
```

> 主图（缩略图）上传

> controller

调到添加页面

```
    /**
     * 执行添加
     * @param route
     * @param rimageFile
     * @param request
     * @return
     * @throws IOException
     */
    @RequestMapping("/doadd")
    public String doAdd(Route route, @RequestParam("rimageFile") MultipartFile
rimageFile, HttpServletRequest request) throws IOException {
        //项目的部署目录 + img/product/rimage/
        performRImage(route, rimageFile, request);

        routeService.add(route);
        return "redirect:/admin/route/page";
    }

    @RequestMapping("/doupdate")
    public String doUpdate(Route route, @RequestParam("rimageFile")
MultipartFile rimageFile, HttpServletRequest request) throws IOException {
        performRImage(route, rimageFile, request);

        routeService.update(route);
        return "redirect:/admin/route/page";
    }

    private void performRImage(Route route, @RequestParam("rimageFile")
MultipartFile rimageFile, HttpServletRequest request) throws IOException {
        //项目的部署目录 + img/product/rimage/
        String savePath =
request.getServletContext().getRealPath("img/product/rimage/");

        //处理随机文件名
        String fileName = UUID.randomUUID().toString().replaceAll("-", "") + "."
+ FilenameUtils.getExtension(rimageFile.getOriginalFilename());

        //上传目录如果不存在,先创建
        File savePathDir = new File(savePath);
        if (!savePathDir.exists()) {
            savePathDir.mkdirs();
        }

        //保存文件
        rimageFile.transferTo(new File(savePathDir, fileName));
        //设置线路的rimage属性=文件的相对路径
        route.setRimage("img/product/rimage/" + fileName);
    }
```

项目图片文件上传，***注意重复文件名的处理***

> 页面

页面使用基于bootstrap的图片上传插件fileinput

参考官方文档http://bootstrap-fileinput.com/

bootstrap-fileinput是一款非常优秀的HTML5文件上传插件，支持文件预览、多选等一系列特性

一款非常优秀的HTML5文件上传插件，支持bootstrap 3.x 和4.x版本，具有非常多的特性:多文件选择。这个插件能最简单的帮你完成文件上传功能，且使用bootstrap样式。还支持多种文件的预览，images, text, html, video, audio, flash。另外还支持ajax方式上传文件，可以看到上传进度。支持拖拽的方式添加和删除文件

> 用法

如果你将一个`css` `class='file'`属性赋予`input`标签，插件将自动把字段`[input type="file"]`转换为文件输入控件。但是，如果你想通过`javascript`单独初始化插件，那么请勿将`css` `class='file'`属性附加到'input'上（因为这将导致重复的初始化，并且JavaScript代码可能会被跳过不执行）

步骤一

引入相关的依赖的插件

```
<script src="/assets/js/bootstrap.min.js"></script>
<link href="/js/fileinput/css/fileinput.css" media="all" rel="stylesheet"
type="text/css" />
<script src="/js/fileinput/js/fileinput.js" type="text/javascript"></script>
<script src="/js/fileinput/js/fileinput_locale_zh.js" type="text/javascript">
</script>
```

步骤二

方法一：在文件域中设置class="file"

```
<input id="rimageFile" name="rimageFile" class="file" type="file">
```

> 注意html的input属性multiple，决定fileinput是否接受多图片上传

方法二：使用js把文件域转换成fileinput对象

```
<input id="rimageFile" name="rimageFile" type="file" >
<script >
    $(function(){
        $("#rimageFile").fileinput({
        });
    })
</script>
```

> 修改

加载数据，调到修改页面

```
    @RequestMapping("/toupdate/{id}")
    public String toUpdate(@PathVariable("id") Integer id, Model model) {
        Route route = routeService.findById(id);
        model.addAttribute("route", route);

        //查询所有分类生成下拉框
        List<Category> categories = categoryService.find();
        model.addAttribute("categories", categories);
        List<Seller> sellers = sellerService.find(new Seller());
        model.addAttribute("sellers", sellers);

        return "route/update";
    }
```

> 修改回显主图（缩略图）到页面

拷贝素材，修改响应代码练习

```
            <script th:inline="javascript">
                $(function(){
                    var rimage = '/' + [[${route.rimage}]];
                    $("#rimageFile").fileinput({
                        initialPreview: [
                            "<img src='" + rimage +"' class='file-preview-image'
>"
                        ],
                        overwriteInitial: true
                    });

                })
            </script>
```

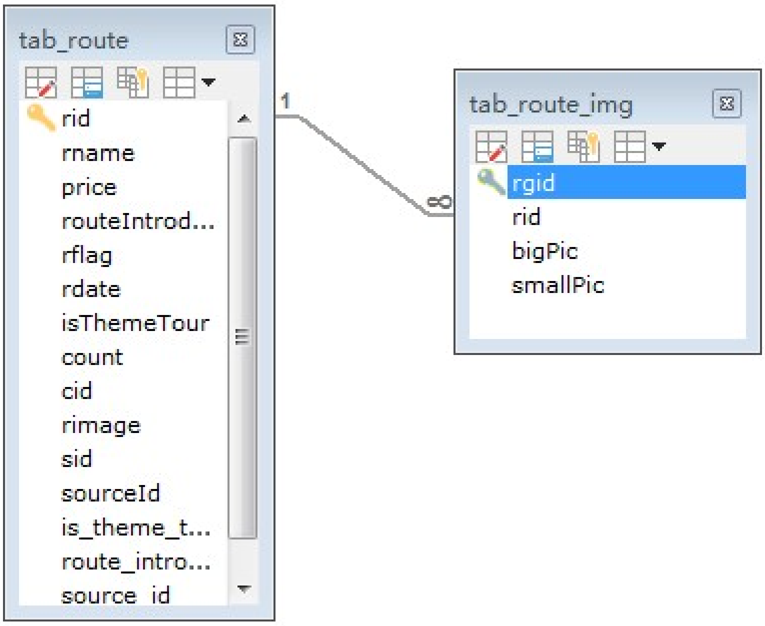
initialPreview：数组类型，指定初始化回显的图片列表

overwriteInitial：覆盖之前回显的内容

线路详细图片

需求: 一条旅游线路，有多张介绍线路的详细图片，详细图片分为小图和大图，前端选择小图可以预览大图

数据结构



tab_route

rid
rname
price
routeIntrod...
rflag
rdate
isThemeTour
count
cid
rimage
sid
sourceId
is_theme_t...
route_intro...
source_id

tab_route_img

rgid
rid
bigPic
smallPic

> service

```
public interface RouteImgService {

    /**
     * 处理线路图片
     * @param rid：线路id
     * @param routeImgs：要添加的线路图片列表
     */
    public void saveImg(Integer rid, List<RouteImg> routeImgs);

}
```

```
@Service
public class RouteImgServiceImpl implements RouteImgService {

    @Autowired
    private RouteImgDao imgDao;


    @Override
    @Transactional
    public void saveImg(Integer rid, List<RouteImg> routeImgs) {
        imgDao.delete(Wrappers.<RouteImg>query().eq("rid", rid));
        for (RouteImg ri : routeImgs) {
            imgDao.insert(ri);
        }
    }
}
```

> controller和页面

> 查看线路详细图片

```
    /**
     * 根据id加载线路图片，跳到image.html
     * @param id
     * @param model
     * @return
     */
    @RequestMapping("/toimage/{id}")
    public String toImage(@PathVariable("id") Integer id, Model model) {
        Route route = routeService.findById(id);
        model.addAttribute("route", route);

        return "route/image";
    }
```

页面回显image.html

```
            <div class="page-content">

                <form action="/admin/route/doimage" enctype="multipart/form-
data" method="post">
                    <input type="hidden" name="rid" th:value="${route.rid}">

                    <div class="form-group">
                        <label for="bigPicFile">详细大图：</label>
                        <input id="bigPicFile" name="bigPicFile" type="file"
multiple >
                    </div>
                    <div class="form-group">
                        <label for="smallPicFile">详细小图：</label>
                        <input id="smallPicFile" name="smallPicFile" type="file"
multiple >
                    </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><!-- /.page-content -->
            <script th:inline="javascript">
                var ri = [[${route.routeImgList}]]; //后台查询的图片列表
                var bi = []; //回显的大图数组
                var si = []; //回显的小图数组
                for (var i = 0; i < ri.length; i++) {
                    bi.push("<img src='/" + ri[i].bigpic +"' class='file-preview-image' >");
                    si.push("<img src='/" + ri[i].smallpic +"' class='file-preview-image' >");
                }
                $("#bigPicFile").fileinput({
                    initialPreview: bi,
                    overwriteInitial: true,
                    minFileCount: 1,
                    maxFileCount: 4
                });
                $("#smallPicFile").fileinput({
                    initialPreview: si,
                    overwriteInitial: true,
                    minFileCount: 1,
                    maxFileCount: 4
                });
            </script>
        </div>
    </div><!-- /.main-content -->
```

注意html5属性multiple，表示可以多文件上传

> 保存线路详细图片

```
    /**
     * 根据id加载线路图片，跳到image.html
     * @param id
     * @param model
     * @return
     */
    @RequestMapping("/toimage/{id}")
    public String toImage(@PathVariable("id") Integer id, Model model) {
        Route route = routeService.findById(id);
        model.addAttribute("route", route);

        return "route/image";
    }

    @RequestMapping("/doimage")
    public String doImage(
            Integer rid,
            @RequestParam("bigPicFile") MultipartFile[] bigPicFile,
            @RequestParam("smallPicFile")MultipartFile[] smallPicFile,
            HttpServletRequest request) throws  Exception {

        List<String> bigPic = new ArrayList<>();
        List<String> smallPic = new ArrayList<>();

        String path = request.getServletContext().getRealPath("/");
        for (MultipartFile f : bigPicFile) {
            File bigPath = new File(path + "img\\product\\big-pic\\");
            if (!bigPath.exists()) {
                bigPath.mkdirs();
            }
            String fileName = UUID.randomUUID().toString().replace("-", "") +
"." + FilenameUtils.getExtension(f.getOriginalFilename());
            f.transferTo(new File(bigPath, fileName));
            bigPic.add("img/product/big-pic/" + fileName);
        }

        for (MultipartFile f : smallPicFile) {
            File smallPath = new File(path + "img\\product\\small-pic\\");
            if (!smallPath.exists()) {
                smallPath.mkdirs();
            }
            String fileName = UUID.randomUUID().toString().replace("-", "") +
"." + FilenameUtils.getExtension(f.getOriginalFilename());
            f.transferTo(new File(smallPath, fileName));
            smallPic.add("img/product/small-pic/" + fileName);
        }

        //要添加的图片列表
        List<RouteImg> ris = new ArrayList<>();
        for (int i=0; i<bigPic.size(); i++) {
            RouteImg img = new RouteImg();
            img.setRid(rid);
            img.setBigpic(bigPic.get(i));
            img.setSmallpic(smallPic.get(i));
            ris.add(img);
        }

        imgService.saveImg(rid, ris);

        return  "redirect:/admin/route/page";
    }
```