中北大学软件学院

**实验报告**

|  |  |
| --- | --- |
| 专 业： | 软件工程 |
| 方 向： | 云计算与大数据技术 |
| 课程名称： | NoSQL数据库技术 |
| 班 级： |  |
| 学 号： |  |
| 姓 名： |  |
| 辅导教师： |  |

2022年9月制

成绩：

|  |  |  |  |
| --- | --- | --- | --- |
| 实验时间 | 2022年11 月13 日19 时22 时 | 学时数 | 4学时 |
| 1.实验名称  小型非关系型数据库系统应用开发 | | | |
| 2.实验目的  (1) 掌握文档数据库数据存储模式特点；  (2) 掌握使用MongoDB数据库管理数据的方法；  (3) 掌握Java应用程序连接MongoDB的方法以及操作文档的方法。 | | | |
| 3.实验内容  开发小型非关系型数据库应用系统：模拟某大型电商平台业务需求，使用非关系型数据库存储数据，实现以下业务功能模块：日志存储、用户扩展信息管理、商品信息管理、商品评价、订单信息管理、历史订单、点击量存储。  借助MongoDB和Eclipse等开发工具，实现相关业务功能模块开发（至少完成上面功能模块其中3个子模块）。 | | | |
| 4.文件描述：此处写清楚共涉及到几个文件（名字和功能）  后端使用spring boot框架，采用MVC架构，service业务层，dao数据库交互层实现mongo数据库操作，前端使用静态HTML页面，使用layui以及jQuery美化，数据交互采用Ajax交互。  本项目实现了以下四个模块：日志存储，用户扩展信息管理，订单信息管理，历史订单。  涉及文件：com/mongodemov3/config：日志配置文件  com/mongodemov3/controller：mvc控制层  com/mongodemov3/pojo：实体类  com/mongodemov3/service：mvc业务层  com/mongodemov3/dao：数据库操作层  com/mongodemov3/utils：日志IP工具类  com/mongodemov3/Mongodemov3Application.java：项目启动类  static/js：jQuery存放地址  static/layui：layui存放地址  static/addlogistics.html：物流信息表单页面  static/addorder.html：添加订单信息表单页面  static/editexuser.html：修改用户拓展信息表单页面  static/editorder.html：修改订单信息表单页面  static/edituser.html：修改用户基本信息表单页面  static/hisordermanage.html：展示历史订单页面  static/index.html：登陆页面  static/ordermanage.html：订单信息管理页面  static/system.html：系统主页面  static/user.html：用户基本信息表单页面  static/userall.html：用户所有信息表单页面  static/userex.html：用户拓展信息表单页面  static/usermanager.html：展示用户基本信息页面  static/usermanagerdetails.html：展示用户拓展信息页面  src/main/resources/application.properties:springboot框架配置文件  src/main/resources/logback-spring.xml：日志文件配置文件  pom.xml：springboot maven依赖配置文件 | | | |
| 1. 核心代码及主要运行界面截图 2. 数据库数据据采集   数据库名称： springboot  集合名称：historyorder  职责：存储历史订单信息    集合名称：logdb  职责：日志信息存储（logback框架类）    集合名称：logs  职责：存储日志信息（IP，请求方法，请求名称类）    集合名称：order  职责：订单信息存储    集合名称：user  职责：用户信息存储     1. 前端登陆界面   代码：  index.html  <!DOCTYPE *html*> <html *lang*="en"> <head>  <meta *charset*="UTF-8">  <title>登录</title>   <link *rel*="stylesheet" *href*="layui/css/layui.css">  <style>  .*login\_line1* img{  margin-left: 155px;  }  form{  margin-left: 15px;  }  .*login\_line3* i{  cursor: pointer;  }  .*login\_other\_l*{  display: inline-block;  transform: translateY(-90%);  margin-top: 50%;  width: 140px;  height: 1px;  border-bottom: 1px solid #eee;  margin-right: 20px;  }  .*login\_other\_r*{  display: inline-block;  transform: translateY(-90%);  width: 140px;  height: 1px;  border-bottom: 1px solid #eee;  margin-left: 20px;  }  .*login\_line3* i{  font-size: 50px;   }  .*login\_line3* i:*nth-child*(1){  margin-left: 70px;  color: green;  }  .*login\_line3* i:*nth-child*(2){  margin-left: 80px;  color: #FE5134;  }  .*login\_line3* i:*nth-child*(3){  margin-left: 80px;  color: #12B7F5;  }  .*login\_line7* .*login\_rem*{  margin-left: 250px;   }  input{  max-width: 350px;  }  .*login\_rem* a{  color: #FD5353;  }  .*login\_line8* button{  text-align: center;  width: 450px;  }  .*login\_line8* button:*first-child*{  margin-left: auto;  margin-right: auto;  background-color: #fff;  border:1px solid #FD5353;  color: #FD5353;  }  .*login\_line8* button:*last-child*{  margin-left: auto;  margin-right: auto;  background-color: #FD5353;  border:1px solid #FD5353;  }  .*layui-form-checked*[lay-skin=primary] i{  border-color:#FD5353 *!important*;  }  .*layui-form-checked* i, .*layui-form-checked*:*hover* i{  background-color: #FD5353 *!important*;  }  </style>  </head> <body> <div *class*="layui-row">  <form *action*="" *class*="layui-form layui-form-pane layui-col-md4 layui-col-md-offset3" *name*="f">  *<!--第四行-->* <div *class*="layui-form-item login\_line4">  <i *class*="login\_other\_l"></i>登录<i *class*="login\_other\_r"></i>  </div>  *<!--第五行-->* <div *class*="layui-form-item login\_line5">  <label *class*="layui-form-label"><i  *class*="layui-icon layui-icon-username"></i></label>  <div *class*="layui-input-block">  <input *type*="text" *id*='username' *name*="username" *lay-verify*="username" *class*="layui-input" *placeholder*="请输入用户名"/>  </div>  </div>  *<!-- 第六行-->* <div *class*="layui-form-item login\_line6">  <label *class*="layui-form-label"><i *class*="layui-icon layui-icon-password"></i></label>  <div *class*="layui-input-block">  <input *type*="password" *id*="password" *name*="password" *lay-verify*="password" *class*="layui-input" *placeholder*="请输入密码"/>  </div>  </div>  <div *class*="layui-form-item login\_line8">  <button></button>  <button *class*="layui-btn login\_btn" *lay-submit lay-filter*="login" *onclick*="*check*(*this*)">登陆</button>  *<!--<button type="button" class="layui-btn" lay-submit lay-filter="login">登录</button>-->* <button></button>  </div>   </form> </div> <script *src*="js/jquery-3.5.1.min.js"></script> <script *src*="layui/layui.js"></script> <script>  *function check*(thisform) {   *var* name=document.getElementById("username").value; *//读取表单数据，创建变量  var* pass=document.getElementById("password").value;   *if* (name==="2013040544q" && pass==="qsj0308") {  *//验证变量。此处设置账号、密码（可设置多组，用||隔开）  alert*("登录成功！");  window.document.f.action="system.html"; *//此处设置登录后跳转页面* window.document.f.submit();  *return* self.location='system.html';  }  *else*{  *alert*("用户名或密码错误！");  }   } </script> <script>  layui.use(['form','layer'],*function* (){  *var* layer=layui.layer;  *var* form =layui.form;   form.verify({  username: *function*(value) {  *if* (value.length == 0) {  *return* '用户名不能为空';  }  *if* (!*new* RegExp("^[a*-*zA*-*Z0*-*9\_\u4e00*-*\u9fa5\\s·]+$").test(value)) {  *return* '用户名不能有特殊字符';  }  *if* (/(^\\_)|(\\_\_)|(\\_+$)/.test(value)) {  *return* '用户名首尾不能出现下划线\'\_\'';  }  *if* (/^\d+\d+\d$/.test(value)) {  *return* '用户名不能全为数字';  }   *//如果不想自动弹出默认提示框，可以直接返回 true，这时你可以通过其他任意方式提示（v2.5.7 新增）  if* (value === 'xxx') {  *alert*('用户名不能为敏感词');  *return true*;  }  }  ,password: [  /^[\S]{6,12}$/  ,'密码必须6到12位，且不能出现空格'  ],    *// ,phone:[  // /^[1][0-9]{10}$/  // ,'手机号格式不正确'  // ]* });  $(".*login*").on('click',*function* (){  *var* index = parent.layer.getFrameIndex(window.name);  parent.layer.close(index);  parent.layer.open({  url:'/system.html',  type:2, *//弹出一个页面层* title:"system",  content:'system.html',  area: ['500px', '550px'],  anim:1  })  })  }) </script>  </body> </html>   1. 系统页面     代码：  system.html  <!DOCTYPE *html*> <html> <head>  <meta *charset*="utf-8">  <meta *name*="viewport" *content*="width=device-width, initial-scale=1, maximum-scale=1">  <title>订单管理系统</title>  <link *rel*="stylesheet" *href*="layui/css/layui.css" *rel*="external nofollow" *rel*="external nofollow" *rel*="external nofollow" *rel*="external nofollow" > </head> <body *class*="layui-layout-body"> <div *class*="layui-layout layui-layout-admin">  <div *class*="layui-header">  <div *class*="layui-logo">system</div>  *<!-- 头部区域（可配合layui已有的水平导航） -->* <ul *class*="layui-nav layui-layout-right">  <li *class*="layui-nav-item">  <a *href*="javascript:;" *rel*="external nofollow" *rel*="external nofollow" >  </a>  </li>  </ul>  </div>  <div *class*="layui-side layui-bg-black">  <div *class*="layui-side-scroll">  *<!-- 左侧导航区域（可配合layui已有的垂直导航） -->* <ul *class*="layui-nav layui-nav-tree" *lay-filter*="test">  <li *class*="layui-nav-item layui-nav-itemed">  <a *class*="" *href*="javascript:;" *rel*="external nofollow" *rel*="external nofollow" >订单管理</a>  <dl *class*="layui-nav-child">  <dd><a *href*="ordermanage.html" *rel*="external nofollow" *target*="container">订单管理</a></dd>  <dd><a *href*="addorder.html" *rel*="external nofollow" *target*="container">订单添加</a></dd>  <dd><a *href*="addlogistics.html" *rel*="external nofollow" *target*="container">物流添加</a></dd>  <dd><a *href*="hisordermanage.html" *rel*="external nofollow" *target*="container">历史订单管理</a></dd>  </dl>  </li>  </ul>  <ul *class*="layui-nav layui-nav-tree" *lay-filter*="test">  <<li *class*="layui-nav-item layui-nav-itemed">  <a *class*="" *href*="javascript:;" *rel*="external nofollow" *rel*="external nofollow" >用户扩展信息管理</a>  <dl *class*="layui-nav-child">  <dd><a *href*="usermanager.html" *rel*="external nofollow" *target*="container">用户基本信息展示</a></dd>  <dd><a *href*="usermanagerdetails.html" *rel*="external nofollow" *target*="container">用户拓展信息展示</a></dd>  <dd><a *href*="userall.html" *rel*="external nofollow" *target*="container">用户全部基本信息展示</a></dd>  <dd><a *href*="user.html" *rel*="external nofollow" *target*="container">添加用户信息</a></dd>  <dd><a *href*="userex.html" *rel*="external nofollow" *target*="container">用户信息拓展</a></dd>  </dl>  </li>  </ul>  </div>  </div>  <div *class*="layui-body">  *<!-- 内容主体区域 -->* <iframe *src*="addorder.html" *name*="container" *width*="100%" *height*="100%"></iframe>  </div>  <div *class*="layui-footer">  *<!-- 底部固定区域 -->* </div> </div> <script *src*="layui/layui.js"></script> <script *src*="layui/modules/jquery.js"></script> *<!--<script src="layui/main.js"></script>-->* <script>  *// JavaScript代码区域* layui.use('element', *function*(){  *var* $ = layui.jquery  ,element = layui.element; *//Tab的切换功能，切换事件监听等，需要依赖element模块 //触发事件  var* active = {  tabAdd: *function*(){ *//新增一个Tab项* element.tabAdd('demo', {  title: '新选项'+ (Math.random()\*1000|0) *//用于演示* ,content: '内容'+ (Math.random()\*1000|0)  ,id: *new* Date().getTime() *//实际使用一般是规定好的id，这里以时间戳模拟下* })  }  ,tabDelete: *function*(othis){ *//删除指定Tab项* element.tabDelete('demo', '44'); *//删除：“商品管理”* othis.addClass('layui-btn-disabled');  }  ,tabChange: *function*(){ *//切换到指定Tab项* element.tabChange('demo', '22'); *//切换到：用户管理* }  };  }); </script> <script>  *//JavaScript代码区域* layui.use(['element', 'layer'], *function* () {  *var* element = layui.element;  *var* layer = layui.layer;  }); </script> </body> </html>   1. 订单管理前端界面     代码：  addorder.html  <!DOCTYPE *html*> <html *lang*="en"> <head>  <meta *charset*="UTF-8">  <title>Title</title>  <link *rel*="stylesheet" *href*="layui/css/layui.css"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow" > </head> <body> <blockquote *class*="layui-elem-quote layui-text"> 订单管理 </blockquote> <div *class*="layui-form-item">  <div *class*="layui-inline">  <label *class*="layui-form-label">订单号</label>  <div *class*="layui-input-inline">  <input *type*="tel" *name*="orderid" *id*="orderid" *autocomplete*="off" *class*="layui-input">  </div>  </div>  <button *class*="layui-btn" *id*="seach" *onclick*="*search*()">搜索</button><br>  <div *style*="padding: 20px; background-color: #F2F2F2;">  <div *class*="layui-row layui-col-space15">  <div *class*="layui-col-md6">  <div *class*="layui-card">  <div *class*="layui-card-header" *id*="order"></div>  <div *class*="layui-card-body" *id*="orderbody">  </div>  </div>  </div>  </div>  </div>  <ul *class*="layui-timeline" *id*="timezhou"></ul>  <div *class*="larry-personal-body clearfix">  <table *class*="layui-hide" *id*="ordertable" *lay-filter*="ordertable"></table>  </div> </div> </body> <script *type*="text/javascript" *src*="layui/layui.js"></script> <script *type*="text/javascript" *src*="js/jquery-3.5.1.min.js"></script> <script *type*="text/html" *id*="barDemo">  <a *class*="layui-btn layui-btn-xs" *lay-event*="edit">编辑</a>  <a *class*="layui-btn layui-btn-danger layui-btn-xs" *lay-event*="del">删除</a> </script> <script *type*="text/html" *id*="toolbarDemo">  <div *class*="layui-btn-container">  <button *class*="layui-btn layui-btn-sm" *lay-event*="getCheckData">右侧进行筛选导出</button>  </div> </script> <script *type*="text/javascript">  *function search*() {*//根据  var* orderid = $("#orderid").val();  $("#orderbody").html('');  $("#timezhou").html('');  $.ajax( {  url:"getorderbyid",  data:{  'id':orderid  },method:'GET',  success:*function* (order) {  $("#order").html('订单号：'+orderid+'('+order['status']+')');  $("#orderbody").append('发件人：'+order['shipper']+'&nbsp;发件人手机：'+order['shipperPhone']+'&nbsp;发件人地址：'+order['shippingAdress']+'&nbsp;下单时间：'+order['shipTime']);  $("#orderbody").append('<br>收件人：'+order['recevier']+'&nbsp;收获人手机：'+order['receviePhone']+'&nbsp;收获人地址：'+order['recevierAddress']);  */\* var logistics=order['logistics'];  console.log(logistics);  for(var i=logistics.length-1;i>=0;i--)  {  console.log(logistics[i]);  $("#timezhou").append(' <li class="layui-timeline-item">\n' +  ' <i class="layui-icon layui-timeline-axis"> </i>\n' +  ' <div class="layui-timeline-content layui-text">\n' +  ' <h3 class="layui-timeline-title">'+'('+logistics[i].operation+')'+logistics[i].operationTime+  ' </h3><p>'+logistics[i].operator+'&nbsp;'+logistics[i].details+'<br>'+logistics[i].adress);  if(logistics[i].phone!=0)  {  $("#timezhou").append('<br>'+logistics[i].phone);  }  $("#timezhou").append(' </p>\n' +  ' </div>\n' +  ' </li>');  }\*/* },  error:*function* (order) {  layer.msg(order)  }  })  }  layui.use('table', *function*(){  *var* table = layui.table;*//高版本建议把括号去掉，有的低版本，需要加()* table.render({  elem: '#ordertable'  ,url: 'getallorders' *//数据接口* ,page: *true //开启分页* ,toolbar: '#toolbarDemo'  ,cols: [[ *//表头* {field: 'id', title: 'id', sort: *true*, fixed: 'left',width:80}  ,{field: 'orderTime', title: '下单时间',sort:*true*,width:80}  ,{field: 'recevierAddress', title: '收货地址',edit:'text'}  ,{field: 'recevier', title: '收货人' ,edit:'text'}  ,{field: 'receviePhone', title: '收货人手机' ,edit:'text'}  ,{field: 'shippingAdress', title: '发货地址'}  ,{field: 'shipper', title: '发货人'}  ,{field: 'shipperPhone', title: '发货人手机'}  ,{field: 'status', title: '物流状态'}  ,{fixed: 'right', title:'操作', toolbar: '#barDemo', width:150}  ]] }); *//头工具栏事件 //监听单元格编辑* table.on('tool(ordertable)', *function*(obj){  *var* data = obj.data;  *var* recevierAddress = obj.recevierAddress;  *var* recevier=obj.recevier;  *var* receviePhone=obj.receviePhone;  console.log(obj)  *if*(obj.event === 'del'){  layer.confirm('真的删除行么', *function*(index){  $.ajax({  url:'deletebyid',  data: {  'id':data.id,  },method:'GET',  traditional: *true*,  success:*function* (msg) {  layer.msg(msg);  obj.del();  },  error:*function* (msg) {  layer.msg(msg)  }  });  layer.close(index);  });  } *else if*(obj.event === 'edit'){  self.location='editorder.html';  layer.msg(JSON.stringify("您可以直接单击单元格进行编辑"))   }  layer.close(index);  });  }); </script>   1. 订单添加页面：     代码：  addorder.html  <!DOCTYPE *html*> <html *lang*="en"> <head>  <meta *charset*="UTF-8">  <title>Title</title>  <link *rel*="stylesheet" *href*="layui/css/layui.css"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow" > </head> <body> <section *class*="layui-larry-box">  <div *class*="larry-personal">  <blockquote *class*="layui-elem-quote layui-text">  <span>增加订单</span>  </blockquote>  <form *class*="layui-form col-lg-5 " *action*="addorder" *method*="post">  <div *class*="layui-form-item">  <label *class*="layui-form-label">订单id</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="id" *autocomplete*="off" *class*="layui-input" *value*="" >  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">发货人姓名</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="shipper" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">发货人地址</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="shippingAdress" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">发货人电话</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="shipperPhone" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">收件人姓名</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="recevier" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">收件人地址</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="recevierAddress" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">收件人手机</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="receviePhone" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <div *class*="layui-input-block">  <button *class*="layui-btn" *lay-submit lay-filter*="formDemo">添加</button>  <button *type*="reset" *class*="layui-btn layui-btn-primary">重置</button>  </div>  </div>  </form>  </div> </section> </body> <script *type*="text/javascript" *src*="layui/layui.js"></script>   1. 物流添加页面：     代码：  addlogistics.html  <!DOCTYPE *html*> <html *lang*="en"> <head>  <meta *charset*="UTF-8">  <title>Title</title>  <link *rel*="stylesheet" *href*="layui/css/layui.css" *rel*="external nofollow" *rel*="external nofollow" *rel*="external nofollow" *rel*="external nofollow" > </head> <body> <section *class*="layui-larry-box">  <div *class*="larry-personal">  <blockquote *class*="layui-elem-quote layui-text">  <span>增加物流信息</span>  </blockquote>  <form *class*="layui-form col-lg-5 " *action*="updateorder" *method*="post">  <div *class*="layui-form-item">  <label *class*="layui-form-label">订单id</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="orderId" *autocomplete*="off" *class*="layui-input" *value*="" >  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">操作名称</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="operation" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">操作员</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="operator" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">操作地址</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="adress" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">备注</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="details" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <div *class*="layui-input-block">  <button *class*="layui-btn" *lay-submit lay-filter*="formDemo">添加</button>  <button *type*="reset" *class*="layui-btn layui-btn-primary">重置</button>  </div>  </div>  </form>  </div> </section> </body> <script *type*="text/javascript" *src*="layui/layui.js"></script>   1. 历史订单管理页面     代码：  hisordermanage.html  <!DOCTYPE *html*> <html *lang*="en"> <head>  <meta *charset*="UTF-8">  <title>Title</title>  <link *rel*="stylesheet" *href*="layui/css/layui.css" *\  rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow" > </head> <body> <blockquote *class*="layui-elem-quote layui-text"> 历史订单管理 </blockquote> <div *class*="layui-form-item">  <ul *class*="layui-timeline" *id*="timezhou"></ul>  <div *class*="larry-personal-body clearfix">  <table *class*="layui-hide" *id*="ordertables" *lay-filter*="ordertables"></table>  </div> </div> </body> <script *type*="text/javascript" *src*="layui/layui.js"></script> <script *type*="text/javascript" *src*="js/jquery-3.5.1.min.js"></script> <script *type*="text/html" *id*="barDemo"> </script> <script *type*="text/html" *id*="toolbarDemo">  <div *class*="layui-btn-container">  <button *class*="layui-btn layui-btn-sm" *lay-event*="getCheckData">右侧进行筛选导出</button>  </div> </script> <script *type*="text/javascript">  layui.use('table', *function*(){  *var* table = layui.table;*//高版本建议把括号去掉，有的低版本，需要加()* table.render({  elem: '#ordertables'  ,url: 'getallhistoryorders' *//数据接口* ,page: *true //开启分页* ,toolbar: '#toolbarDemo'  ,cols: [[ *//表头* {field: 'id', title: 'id', sort: *true*, fixed: 'left',width:80}  ,{field: 'orderTime', title: '下单时间',sort:*true*,width:80}  ,{field: 'recevierAddress', title: '收货地址'}  ,{field: 'recevier', title: '收货人' }  ,{field: 'receviePhone', title: '收货人手机' }  ,{field: 'shippingAdress', title: '发货地址'}  ,{field: 'shipper', title: '发货人'}  ,{field: 'shipperPhone', title: '发货人手机'}  ,{field: 'status', title: '物流状态'}  *//,{field: 'status', title: '物流状态',toolbar: '#barDemo', width:150}* ]] });  }); </script>   1. 用户基本信息展示页面：     代码：  usermanager.html  <!DOCTYPE *html*>  <html *lang*="en">  <head>  <meta *charset*="UTF-8">  <title>Title</title>  <link *rel*="stylesheet" *href*="layui/css/layui.css"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow" > </head> <body> <blockquote *class*="layui-elem-quote layui-text"> 用户基本信息展示 </blockquote> <div *class*="layui-form-item">  <div *class*="layui-inline">  <label *class*="layui-form-label">用户id</label>  <div *class*="layui-input-inline">  <input *type*="tel" *name*="userid" *id*="userid" *autocomplete*="off" *class*="layui-input">  </div>  </div>  <button *class*="layui-btn" *id*="seach" *onclick*="search()">搜索</button><br>  <div *style*="padding: 20px; background-color: #F2F2F2;">  <div *class*="layui-row layui-col-space15">  <div *class*="layui-col-md6">  <div *class*="layui-card">  <div *class*="layui-card-header" *id*="user"></div>  <div *class*="layui-card-body" *id*="userbody">  </div>  </div>  </div>  </div>  </div>  <ul *class*="layui-timeline" *id*="timezhou"></ul>  <div *class*="larry-personal-body clearfix">  <table *class*="layui-hide" *id*="usertable" *lay-filter*="usertable"></table>  </div> </div> </body> <script *type*="text/javascript" *src*="layui/layui.js"></script> <script *type*="text/javascript" *src*="js/jquery-3.5.1.min.js"></script> <script *type*="text/html" *id*="barDemo">  <a *class*="layui-btn layui-btn-xs" *lay-event*="edit">编辑</a>  <a *class*="layui-btn layui-btn-danger layui-btn-xs" *lay-event*="del">删除</a> </script> <script *type*="text/html" *id*="toolbarDemo">  <div *class*="layui-btn-container">  <button *class*="layui-btn layui-btn-sm" *lay-event*="getCheckData">右侧进行筛选导出</button>  </div> </script> <script *type*="text/javascript">  *function* search() {*//根据  var* userid = $("#userid").val();  $("#userbody").html('');  $("#timezhou").html('');  $.ajax( {  url:"getuserbyid",  data:{  'id':userid  },method:'GET',  success:*function* (user) {  $("#user").html('用户id：'+userid);  $("#userbody").append('姓名：'+user['name']+'&nbsp;昵称：'+user['nickname']+'&nbsp;年龄：'+user['age']+'&nbsp;性别：'+user['sex']);  *//$("#userbody").append('<br>收件人：'+user['codeNum']+'&nbsp;收获人手机：'+user['phone']+'&nbsp;收获人地址：'+user['recevierAddress']);  /\* var logistics=user['logistics'];  console.log(logistics);  for(var i=logistics.length-1;i>=0;i--)  {  console.log(logistics[i]);  $("#timezhou").append(' <li class="layui-timeline-item">\n' +  ' <i class="layui-icon layui-timeline-axis"> </i>\n' +  ' <div class="layui-timeline-content layui-text">\n' +  ' <h3 class="layui-timeline-title">'+'('+logistics[i].operation+')'+logistics[i].operationTime+  ' </h3><p>'+logistics[i].operator+'&nbsp;'+logistics[i].details+'<br>'+logistics[i].adress);  if(logistics[i].phone!=0)  {  $("#timezhou").append('<br>'+logistics[i].phone);  }  $("#timezhou").append(' </p>\n' +  ' </div>\n' +  ' </li>');  }\*/* },  error:*function* (user) {  layer.msg(user)  }  })  }  layui.use('table', *function*(){  *var* table = layui.table;*//高版本建议把括号去掉，有的低版本，需要加()* table.render({  elem: '#usertable'  ,url: 'getalluser' *//数据接口* ,page: *true //开启分页* ,toolbar: '#toolbarDemo'  ,cols: [[ *//表头* {field: 'id', title: '用户id', sort: *true*, fixed: 'left',width:80}  ,{field: 'name', title: '姓名',width:80}  ,{field: 'nickname', title: '昵称',edit: 'text'}  ,{field: 'age', title: '年龄' ,edit:'text'}  ,{field: 'sex', title: '性别',edit: 'text' }  ,{fixed: 'right', title:'操作', toolbar: '#barDemo', width:150}  ]] }); *//头工具栏事件 //监听单元格编辑* table.on('tool(usertable)', *function*(obj){  *var* data = obj.data;  console.log(obj)  *if*(obj.event === 'del'){  layer.confirm('真的删除行么', *function*(index){  $.ajax({  url:'deletebyuserid',  data: {  'id':data.id,  },method:'GET',  traditional: *true*,  success:*function* (msg) {  layer.msg(msg);  obj.del();  },  error:*function* (msg) {  layer.msg(msg)  }  });  layer.close(index);  });  } *else if*(obj.event === 'edit'){  self.location='edituser.html';  layer.msg(JSON.stringify("您可以直接单击单元格进行编辑"))   }  layer.close(index);  });  }); </script>   1. 用户拓展信息展示页面：     代码：  usermanagerdetails.html  <!DOCTYPE *html*> <html *lang*="en"> <head>  <meta *charset*="UTF-8">  <title>Title</title>  <link *rel*="stylesheet" *href*="layui/css/layui.css"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow" > </head> <body> <blockquote *class*="layui-elem-quote layui-text"> 用户拓展信息展示 </blockquote> <div *class*="layui-form-item">  <div *class*="layui-inline">  <label *class*="layui-form-label">用户id</label>  <div *class*="layui-input-inline">  <input *type*="tel" *name*="userid" *id*="userid" *autocomplete*="off" *class*="layui-input">  </div>  </div>  <button *class*="layui-btn" *id*="seach" *onclick*="*search*()">搜索</button><br>  <div *style*="padding: 20px; background-color: #F2F2F2;">  <div *class*="layui-row layui-col-space15">  <div *class*="layui-col-md6">  <div *class*="layui-card">  <div *class*="layui-card-header" *id*="user"></div>  <div *class*="layui-card-body" *id*="userbody">  </div>  </div>  </div>  </div>  </div>  <ul *class*="layui-timeline" *id*="timezhou"></ul>  <div *class*="larry-personal-body clearfix">  <table *class*="layui-hide" *id*="usertable" *lay-filter*="usertable"></table>  </div> </div> </body> <script *type*="text/javascript" *src*="layui/layui.js"></script> <script *type*="text/javascript" *src*="js/jquery-3.5.1.min.js"></script> <script *type*="text/html" *id*="barDemo">  <a *class*="layui-btn layui-btn-xs" *lay-event*="edit">编辑</a>  <a *class*="layui-btn layui-btn-danger layui-btn-xs" *lay-event*="del">删除</a> </script> <script *type*="text/html" *id*="toolbarDemo">  <div *class*="layui-btn-container">  <button *class*="layui-btn layui-btn-sm" *lay-event*="getCheckData">右侧进行筛选导出</button>  </div> </script> <script *type*="text/javascript">  *function search*() {*//根据  var* userid = $("#userid").val();  $("#userbody").html('');  $("#timezhou").html('');  $.ajax( {  url:"getuserexbyid",  data:{  'id':userid  },method:'GET',  success:*function* (user) {  $("#user").html('用户id：'+userid);  $("#userbody").append('身份证号：'+user['codeNum']+'&nbsp;电话：'+user['phone']+'&nbsp;学校：'+user['school']+'&nbsp;住址：'+user['address']);  *//$("#userbody").append('<br>收件人：'+user['codeNum']+'&nbsp;收获人手机：'+user['phone']+'&nbsp;收获人地址：'+user['recevierAddress']);  /\* var logistics=user['logistics'];  console.log(logistics);  for(var i=logistics.length-1;i>=0;i--)  {  console.log(logistics[i]);  $("#timezhou").append(' <li class="layui-timeline-item">\n' +  ' <i class="layui-icon layui-timeline-axis"> </i>\n' +  ' <div class="layui-timeline-content layui-text">\n' +  ' <h3 class="layui-timeline-title">'+'('+logistics[i].operation+')'+logistics[i].operationTime+  ' </h3><p>'+logistics[i].operator+'&nbsp;'+logistics[i].details+'<br>'+logistics[i].adress);  if(logistics[i].phone!=0)  {  $("#timezhou").append('<br>'+logistics[i].phone);  }  $("#timezhou").append(' </p>\n' +  ' </div>\n' +  ' </li>');  }\*/* },  error:*function* (user) {  layer.msg(user)  }  })  }  layui.use('table', *function*(){  *var* table = layui.table;*//高版本建议把括号去掉，有的低版本，需要加()* table.render({  elem: '#usertable'  ,url: 'getallexuser' *//数据接口* ,page: *true //开启分页* ,toolbar: '#toolbarDemo'  ,cols: [[ *//表头* {field: 'id', title: '用户id', sort: *true*, fixed: 'left',width:80}  ,{field: 'codeNum', title: '身份证',width:80}  ,{field: 'phone', title: '电话'}  ,{field: 'school', title: '学校' }  ,{field: 'address', title: '住址'}  ,{fixed: 'right', title:'操作', toolbar: '#barDemo', width:150}  ]] }); *//头工具栏事件 //监听单元格编辑* table.on('tool(usertable)', *function*(obj){  *var* data = obj.data;  console.log(obj)  *if*(obj.event === 'del'){  layer.confirm('真的删除行么', *function*(index){  $.ajax({  url:'deletebyuserid',  data: {  'id':data.id,  },method:'GET',  traditional: *true*,  success:*function* (msg) {  layer.msg(msg);  obj.del();  },  error:*function* (msg) {  layer.msg(msg)  }  });  layer.close(index);  });  } *else if*(obj.event === 'edit'){  self.location='edituser.html';   }  layer.close(index);  });  }); </script>   1. 用户全部信息展示页面：     代码：  userall.html  <!DOCTYPE *html*> <html *lang*="en"> <head>  <meta *charset*="UTF-8">  <title>Title</title>  <link *rel*="stylesheet" *href*="layui/css/layui.css"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow" > </head> <body> <blockquote *class*="layui-elem-quote layui-text"> 用户全部信息展示 </blockquote> <div *class*="layui-form-item">  <div *class*="layui-inline">  <label *class*="layui-form-label">用户id</label>  <div *class*="layui-input-inline">  <input *type*="tel" *name*="userid" *id*="userid" *autocomplete*="off" *class*="layui-input">  </div>  </div>  <button *class*="layui-btn" *id*="seach" *onclick*="*search*()">搜索</button><br>  <div *style*="padding: 20px; background-color: #F2F2F2;">  <div *class*="layui-row layui-col-space15">  <div *class*="layui-col-md6">  <div *class*="layui-card">  <div *class*="layui-card-header" *id*="user"></div>  <div *class*="layui-card-body" *id*="userbody">  </div>  </div>  </div>  </div>  </div>  <ul *class*="layui-timeline" *id*="timezhou"></ul>  <div *class*="larry-personal-body clearfix">  <table *class*="layui-hide" *id*="usertable" *lay-filter*="usertable"></table>  </div> </div> </body> <script *type*="text/javascript" *src*="layui/layui.js"></script> <script *type*="text/javascript" *src*="js/jquery-3.5.1.min.js"></script> <script *type*="text/html" *id*="barDemo">  <a *class*="layui-btn layui-btn-xs" *lay-event*="edit">编辑</a>  <a *class*="layui-btn layui-btn-danger layui-btn-xs" *lay-event*="del">删除</a> </script> <script *type*="text/html" *id*="toolbarDemo">  <div *class*="layui-btn-container">  <button *class*="layui-btn layui-btn-sm" *lay-event*="getCheckData">右侧进行筛选导出</button>  </div> </script> <script *type*="text/javascript">  *function search*() {*//根据  var* userid = $("#userid").val();  $("#userbody").html('');  $("#timezhou").html('');  $.ajax( {  url:"getuserallbyid",  data:{  'id':userid  },method:'GET',  success:*function* (user) {  $("#user").html('用户id：'+userid);  $("#userbody").append('姓名：'+user['codeNum']+'&nbsp;昵称：'+user['phone']+'&nbsp;年龄：'+user['school']+'&nbsp;性别：'+user['address']  +'&nbsp;身份证号：'+user['codeNum']+'&nbsp;电话：'+user['phone']+'&nbsp;学校：'+user['school']+'&nbsp;住址：'+user['address']);  *//$("#userbody").append('<br>收件人：'+user['codeNum']+'&nbsp;收获人手机：'+user['phone']+'&nbsp;收获人地址：'+user['recevierAddress']);  /\* var logistics=user['logistics'];  console.log(logistics);  for(var i=logistics.length-1;i>=0;i--)  {  console.log(logistics[i]);  $("#timezhou").append(' <li class="layui-timeline-item">\n' +  ' <i class="layui-icon layui-timeline-axis"> </i>\n' +  ' <div class="layui-timeline-content layui-text">\n' +  ' <h3 class="layui-timeline-title">'+'('+logistics[i].operation+')'+logistics[i].operationTime+  ' </h3><p>'+logistics[i].operator+'&nbsp;'+logistics[i].details+'<br>'+logistics[i].adress);  if(logistics[i].phone!=0)  {  $("#timezhou").append('<br>'+logistics[i].phone);  }  $("#timezhou").append(' </p>\n' +  ' </div>\n' +  ' </li>');  }\*/* },  error:*function* (user) {  layer.msg(user)  }  })  }  layui.use('table', *function*(){  *var* table = layui.table;*//高版本建议把括号去掉，有的低版本，需要加()* table.render({  elem: '#usertable'  ,url: 'getalluserall' *//数据接口* ,page: *true //开启分页* ,toolbar: '#toolbarDemo'  ,cols: [[ *//表头* {field: 'id', title: '用户id', sort: *true*, fixed: 'left',width:80}  ,{field: 'name', title: '姓名',width:80}  ,{field: 'nickname', title: '昵称',edit: 'text'}  ,{field: 'age', title: '年龄' ,edit:'text'}  ,{field: 'sex', title: '性别',edit: 'text' }  ,{field: 'codeNum', title: '省份证',edit: 'text' }  ,{field: 'phone', title: '电话',edit: 'text' }  ,{field: 'school', title: '学校',edit: 'text' }  ,{field: 'address', title:'住址', width:150}  ]] });  *//头工具栏事件 //监听单元格编辑* table.on('tool(usertable)', *function*(obj){  *var* data = obj.data;  console.log(obj)  *if*(obj.event === 'del'){  layer.confirm('真的删除行么', *function*(index){  $.ajax({  url:'deletebyuserid',  data: {  'id':data.id,  },method:'GET',  traditional: *true*,  success:*function* (msg) {  layer.msg(msg);  obj.del();  },  error:*function* (msg) {  layer.msg(msg)  }  });  layer.close(index);  });  } *else if*(obj.event === 'edit'){  self.location='edituser.html';  layer.msg(JSON.stringify("您可以直接单击单元格进行编辑"))   }  layer.close(index);  });  }); </script>   1. 添加用户信息页面：     前端代码：  user.html  <!DOCTYPE *html*> <html *lang*="en"> <head>  <meta *charset*="UTF-8">  <title>Title</title>  <link *rel*="stylesheet" *href*="layui/css/layui.css"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow" > </head> <body> <section *class*="layui-larry-box">  <div *class*="larry-personal">  <blockquote *class*="layui-elem-quote layui-text">  <span>添加用户基本信息</span>  </blockquote>  <form *class*="layui-form col-lg-5 " *action*="adduser" *method*="post">  <div *class*="layui-form-item">  <label *class*="layui-form-label">用户id</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="id" *autocomplete*="off" *class*="layui-input" *value*="" >  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">姓名</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="name" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">昵称</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="nickname" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">年龄</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="age" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">性别</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="sex" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <div *class*="layui-input-block">  <button *class*="layui-btn" *lay-submit lay-filter*="formDemo">添加</button>  <button *type*="reset" *class*="layui-btn layui-btn-primary">重置</button>  </div>  </div>  </form>  </div> </section> </body> <script *type*="text/javascript" *src*="layui/layui.js"></script>   1. 用户信息拓展页面：     前端代码：  userex.html  <!DOCTYPE *html*> <html *lang*="en"> <head>  <meta *charset*="UTF-8">  <title>Title</title>  <link *rel*="stylesheet" *href*="layui/css/layui.css"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow" > </head> <body> <section *class*="layui-larry-box">  <div *class*="larry-personal">  <blockquote *class*="layui-elem-quote layui-text">  <span>添加用户信息拓展</span>  </blockquote>  <form *class*="layui-form col-lg-5 " *action*="addexuser" *method*="post">  <div *class*="layui-form-item">  <label *class*="layui-form-label">用户id</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="id" *autocomplete*="off" *class*="layui-input" *value*="" >  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">身份证</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="codeNum" *autocomplete*="off" *class*="layui-input" *value*="" >  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">电话</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="phone" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">学校</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="school" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">住址</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="address" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <div *class*="layui-input-block">  <button *class*="layui-btn" *lay-submit lay-filter*="formDemo">添加</button>  <button *type*="reset" *class*="layui-btn layui-btn-primary">重置</button>  </div>  </div>  </form>  </div> </section> </body> <script *type*="text/javascript" *src*="layui/layui.js"></script>   1. 修改订单信息页面：     前端代码：  editorder.html  <!DOCTYPE *html*> <html *lang*="en"> <head>  <meta *charset*="UTF-8">  <title>Title</title>  <link *rel*="stylesheet" *href*="layui/css/layui.css"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow" > </head> <body> <section *class*="layui-larry-box">  <div *class*="larry-personal">  <blockquote *class*="layui-elem-quote layui-text">  <span>更改订单信息</span>  </blockquote>  <form *class*="layui-form col-lg-5 " *action*="editorder" *method*="post">  <div *class*="layui-form-item">  <label *class*="layui-form-label">订单id</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="id" *autocomplete*="off" *class*="layui-input" *value*="" >  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">收件人姓名</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="recevier" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">收件人地址</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="recevierAddress" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">收件人手机</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="receviePhone" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <div *class*="layui-input-block">  <button *class*="layui-btn" *lay-submit lay-filter*="formDemo" *lay-event*="edit" *onclick*=*check*(*this*)>更改</button>  </div>  </div>  </form>  </div> </section> </body> <script *type*="text/javascript" *src*="layui/layui.js"></script> <script>  *function check*(thisform){  *var* data = obj.data;  *var* id = obj.id;  *var* recevierAddress = obj.recevierAddress;  *var* recevier=obj.recevier;  *var* receviePhone=obj.receviePhone;  console.log(obj)  *if*(obj.event === 'edit'){  layer.confirm("",*function*(index){  $.ajax({  url:'editorder',  data: {  'id':data.id,  'recevier':data.recevier,  'recevierAddress':data.recevierAddress,  'receviePhone':data.receviePhone  },method:'POST',  traditional: *true*,  success:*function* (msg) {  layer.msg(msg);  obj.edit();  },  error:*function* (msg) {  layer.msg(msg)  }  });  layer.close(index);  });  }  } </script>   1. 修改用户基本信息页面：     前端代码：  edituser.html  <!DOCTYPE *html*> <html *lang*="en"> <head>  <meta *charset*="UTF-8">  <title>Title</title>  <link *rel*="stylesheet" *href*="layui/css/layui.css"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow" > </head> <body> <section *class*="layui-larry-box">  <div *class*="larry-personal">  <blockquote *class*="layui-elem-quote layui-text">  <span>更改用户基本信息</span>  </blockquote>  <form *class*="layui-form col-lg-5 " *action*="edituser" *method*="post">  <div *class*="layui-form-item">  <label *class*="layui-form-label">用户id</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="id" *autocomplete*="off" *class*="layui-input" *value*="" >  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">昵称</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="nickname" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">年龄</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="age" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">性别</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="sex" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <div *class*="layui-input-block">  <button *class*="layui-btn" *lay-submit lay-filter*="formDemo" *onclick*=*check*(*this*)>更改</button>  </div>  </div>  </form>  </div> </section> </body> <script *type*="text/javascript" *src*="layui/layui.js"></script> <script>  *function check*(thisform){  *var* data = obj.data;  *var* id = obj.id;  *var* nickname = obj.nickname;  *var* age=obj.age;  *var* sex=obj.sex;  console.log(obj)  *if*(obj.event === 'edit'){  layer.confirm("",*function*(index){  $.ajax({  url:'edituser',  data: {  'id':data.id,  'nickname':data.nickname,  'age':data.age,  'sex':data.sex  },method:'POST',  traditional: *true*,  success:*function* (msg) {  layer.msg(msg);  obj.edit();  },  error:*function* (msg) {  layer.msg(msg)  }  });  layer.close(index);  });  }  } </script>   1. 修改用户拓展信息页面：     前端代码：  editexuser.html  <!DOCTYPE *html*> <html *lang*="en"> <head>  <meta *charset*="UTF-8">  <title>Title</title>  <link *rel*="stylesheet" *href*="layui/css/layui.css"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow"  *rel*="external nofollow" > </head> <body> <section *class*="layui-larry-box">  <div *class*="larry-personal">  <blockquote *class*="layui-elem-quote layui-text">  <span>修改用户拓展信息</span>  </blockquote>  <form *class*="layui-form col-lg-5 " *action*="editexuser" *method*="post">  <div *class*="layui-form-item">  <label *class*="layui-form-label">用户id</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="id" *autocomplete*="off" *class*="layui-input" *value*="" >  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">电话</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="phone" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">学校</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="school" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <label *class*="layui-form-label">住址</label>  <div *class*="layui-input-block">  <input *type*="text" *name*="address" *autocomplete*="off" *class*="layui-input" *value*="">  </div>  </div>  <div *class*="layui-form-item">  <div *class*="layui-input-block">  <button *class*="layui-btn" *lay-submit lay-filter*="formDemo" *onclick*=*check*(*this*)>更改</button>  </div>  </div>  </form>  </div> </section> </body> <script *type*="text/javascript" *src*="layui/layui.js"></script> <script>  *function check*(thisform){  *var* data = obj.data;  *var* id = obj.id;  *var* phone = obj.phone;  *var* school=obj.school;  *var* address=obj.address;  console.log(obj)  *if*(obj.event === 'edit'){  layer.confirm("",*function*(index){  $.ajax({  url:'editexuser',  data: {  'id':data.id,  'phone':data.phone,  'school':data.school,  'address':data.address  },method:'POST',  traditional: *true*,  success:*function* (msg) {  layer.msg(msg);  obj.edit();  },  error:*function* (msg) {  layer.msg(msg)  }  });  layer.close(index);  });  }  } </script>   1. 后端控制层代码   OrderController.java  *package* com.mongodemov3.controller;  *import* com.mongodemov3.config.MongoDBAppender; *import* com.mongodemov3.pojo.\*; *import* com.mongodemov3.service.orderService; *import* com.mongodemov3.utils.CommonUtils; *import* org.slf4j.Logger; *import* org.slf4j.LoggerFactory; *import* org.springframework.beans.factory.annotation.Autowired; *import* org.springframework.web.bind.annotation.GetMapping; *import* org.springframework.web.bind.annotation.PostMapping; *import* org.springframework.web.bind.annotation.RestController;  *import* java.util.Date; *import* java.util.HashMap; *import* java.util.List; *import* java.util.Map;  @RestController *public class* OrderController {  *// 日志信息注入  private static* Logger *logger*= LoggerFactory.getLogger(OrderController.*class*);  *private static* Logger *logs* = LoggerFactory.getLogger(CommonUtils.*class*);  *private static* Logger *log* = LoggerFactory.getLogger(MongoDBAppender.*class*);   @Autowired  orderService **orderService**;   *// 添加订单信息* @PostMapping("addorder")  *public* String addorder(order order, Log log)  {  order.setStatus("发货中");  order.setOrderTime(*new* Date());  order.setShipTime(*new* Date());  **orderService**.addorder(order);  **orderService**.addLog(log);  *return* "添加成功";  }   *// 添加物流信息* @PostMapping("updateorder")  *public* String updateorder(Logistics logistics,Log log)  {  logistics.setOperationTime(*new* Date());  **orderService**.addLogisticsAndUpdateStatus(logistics);  **orderService**.addLog(log);  *return* "添加成功";  }   *// 修改订单信息* @PostMapping("editorder")  *public* String editupdateorder(*int* id,Log log,String recevier, String recevierAddress, *long* receviePhone)  {  **orderService**.editOrder(id,recevier,recevierAddress,receviePhone);  **orderService**.addLog(log);  *return* "修改成功";  }   *//按id查找订单信息* @GetMapping("getorderbyid")  *public* order getOrderById(*int* id,Logistics logistics,LogBean logBean,Log log)  {  order order=**orderService**.getOrderById(id);  **orderService**.addLog(log);  *return* order;  }   *//按id删除订单信息* @GetMapping("deletebyid")  *public* String deleteById(*int* id,Logistics logistics,LogBean logBean,Log log)  {  **orderService**.deleteOrderById(id);  **orderService**.addLog(log);  *return* "成功";  }   *//展示订单信息* @GetMapping("getallorders")  *public* Map<String,Object> getAllOrder()  {  Map<String,Object> map=*new* HashMap<>();  List<order> list=**orderService**.getAllorder();  map.put("code","0");  map.put("count",list.size());  map.put("data",list);  *return* map;  }   *//通过id查找历史订单信息* @GetMapping("gethistoryorderbyid")  *public* historyorder getHistoryOrderById(*int* id, Logistics logistics, LogBean logBean, Log log)  {  historyorder historyorder=**orderService**.getHistoryorderById(id);  **orderService**.addLog(log);  *return* historyorder;  }   *// 展示所有历史订单信息* @GetMapping("getallhistoryorders")  *public* Map<String,Object> getAllHistoryOrder()  {  Map<String,Object> map=*new* HashMap<>();  List<historyorder> list=**orderService**.getAllHistoryorder();  map.put("code","0");  map.put("count",list.size());  map.put("data",list);  *return* map;  } }  UserController.java  *package* com.mongodemov3.controller;  *import* com.mongodemov3.config.MongoDBAppender; *import* com.mongodemov3.pojo.\*; *import* com.mongodemov3.service.orderService; *import* com.mongodemov3.service.userService; *import* com.mongodemov3.utils.CommonUtils; *import* org.slf4j.Logger; *import* org.slf4j.LoggerFactory; *import* org.springframework.beans.factory.annotation.Autowired; *import* org.springframework.web.bind.annotation.GetMapping; *import* org.springframework.web.bind.annotation.PostMapping; *import* org.springframework.web.bind.annotation.RestController;  *import* java.util.Date; *import* java.util.HashMap; *import* java.util.List; *import* java.util.Map;  @RestController *public class* UserController {  *// 日志信息注入  private static* Logger *logger* = LoggerFactory.getLogger(UserController.*class*);  *private static* Logger *logs* = LoggerFactory.getLogger(CommonUtils.*class*);  *private static* Logger *log* = LoggerFactory.getLogger(MongoDBAppender.*class*);    @Autowired  userService **userService**;  *//添加用户基本信息* @PostMapping("adduser")  *public* String addorder(user user, Log log) {  user.setCreateDate(*new* Date());  **userService**.adduser(user);  **userService**.addLog(log);  *return* "添加成功";  }   *//添加用户拓展信息* @PostMapping("addexuser")  *public* String addexuser(user user, UserExtendInfo userExtendInfo, Log log) {  user.setExdateTime(*new* Date());  **userService**.addexuser(userExtendInfo);  **userService**.addLog(log);  *return* "添加成功";  }   *//按id查找用户* @GetMapping("getuserbyid")  *public* user getOrderById(*int* id, LogBean logBean, Log log) {  user user = **userService**.getUserById(id);  **userService**.addLogs(logBean);  **userService**.addLog(log);  *return* user;  } *//按id删除用户* @GetMapping("deletebyuserid")  *public* String deleteById(*int* id, Log log) {  **userService**.deleteUserById(id);  **userService**.addLog(log);  *return* "成功";  }   *//展示用户基本信息* @GetMapping("getalluser")  *public* Map<String, Object> getAllUser() {  Log log = *new* Log();  Map<String, Object> map = *new* HashMap<>();  List<user> list = **userService**.getAlluser();  map.put("code", "0");  map.put("count", list.size());  map.put("data", list);  **userService**.addLog(log);  *return* map;  }   *//展示用户拓展信息* @GetMapping("getallexuser")  *public* Map<String, Object> getAllexUser() {  {  Log log = *new* Log();  Map<String, Object> map = *new* HashMap<>();  List<UserExtendInfo> list = **userService**.getAllexuser();  map.put("code", "0");  map.put("count", list.size());  map.put("data", list);  **userService**.addLog(log);  *return* map;  }  }   *//展示用户所有2信息* @GetMapping("getalluserall")  *public* Map<String, Object> getAllUserall() {  Log log = *new* Log();  Map<String, Object> map = *new* HashMap<>();  List<userall> list = **userService**.getAlluserall();  map.put("code", "0");  map.put("count", list.size());  map.put("data", list);  **userService**.addLog(log);  *return* map;  }   *//修改用户基本信息* @PostMapping("edituser")  *public* String editupdateuser(*int* id,Log log,String nickname, *int* age, String sex)  {  **userService**.editUser(id,nickname,age,sex);  **userService**.addLog(log);  *return* "修改成功";  }   *//修改用户拓展信息* @PostMapping("editexuser")  *public* String editupdateexuser(*int* id,Log log,String phone, String school, String address)  {  **userService**.editexUser(id,phone,school,address);  **userService**.addLog(log);  *return* "修改成功";  } }   1. 后端业务层代码   orderService.java  *package* com.mongodemov3.service;  *import* com.mongodemov3.pojo.\*;  *import* java.util.List;  *public interface* orderService {  *//添加订单  void* addorder(order order);  *//添加物流日志信息  void* addLogisticsAndUpdateStatus(Logistics logistics);  *//添加日志信息：方法、请求名称、IP信息等  void* addLogs(LogBean logBean);  *//添加日志信息：使用logback(框架日志信息系统实现)  void* addLog(Log log);  *//按照ID查找订单* order getOrderById(*int* id);  *//按照ID删除订单  boolean* deleteOrderById(*int* id);  *//展示所有订单信息* List<order> getAllorder();  *//历史订单* List<historyorder> getAllHistoryorder();  *//按ID查找历史订单信息* historyorder getHistoryorderById(*int* id);  *//修改订单信息  void* editOrder(*int* id,String recevier,String recevierAddress,*long* receviePhone); }  userService.java  *package* com.mongodemov3.service;  *import* com.mongodemov3.pojo.\*;  *import* java.util.List;  *public interface* userService {  *//添加用户基本信息  void* adduser(user user);  *//添加用户拓展信息  void* addexuser(UserExtendInfo userExtendInfo);  *//添加日志信息：方法、请求名称、IP信息等  void* addLogs(LogBean logBean);  *//添加日志信息：使用logback(框架日志信息系统实现)  void* addLog(Log log);  *//按ID查找用户（实现有问题）* user getUserById(*int* id);  *//按ID删除用户信息  boolean* deleteUserById(*int* id);  *//展示所有用户基本信息* List<user> getAlluser();  *//展示用户拓展信息* List<UserExtendInfo> getAllexuser();  *//展示用户所有信息* List<userall> getAlluserall();  *//修改用户基本信息  void* editUser(*int* id,String nickname,*int* age,String sex);  *//修改用户拓展信息  void* editexUser(*int* id,String phone,String school,String address); }  orderServiceImpl.java  *package* com.mongodemov3.service.Impl;  *import* com.mongodemov3.config.MongoDBAppender; *import* com.mongodemov3.dao.orderDao; *import* com.mongodemov3.pojo.\*; *import* com.mongodemov3.service.orderService; *import* com.mongodemov3.utils.CommonUtils; *import* org.slf4j.Logger; *import* org.slf4j.LoggerFactory; *import* org.springframework.beans.factory.annotation.Autowired; *import* org.springframework.data.mongodb.core.MongoTemplate; *import* org.springframework.data.mongodb.core.query.Criteria; *import* org.springframework.data.mongodb.core.query.Query; *import* org.springframework.data.mongodb.core.query.Update; *import* org.springframework.stereotype.Service;  *import* java.util.List;  @Service *public class* orderServiceImpl *implements* orderService {  *private static* Logger *logger*= LoggerFactory.getLogger(orderServiceImpl.*class*);  *private static* Logger *logs* = LoggerFactory.getLogger(CommonUtils.*class*);  *private static* Logger *log* = LoggerFactory.getLogger(MongoDBAppender.*class*);  @Autowired  MongoTemplate **mongoTemplate**;   @Autowired  orderDao **orderDao**;   *//创建订单，传来order对象  public void* addorder(order order)  {  **orderDao**.addorder(order);  }     *//更新物流  public void* addLogisticsAndUpdateStatus(Logistics logistics)  {  **orderDao**.addLogisticsAndUpdateStatus(logistics);  }   @Override  *public void* addLogs(LogBean logBean) {  **orderDao**.addLogs(logBean);  }   @Override  *public void* addLog(Log log) {  **orderDao**.addLog(log);  }   *//通过id查询物流  public* order getOrderById(*int* id)  {  *return* **orderDao**.getOrderById(id);  }  *//根据id删除记录  public boolean* deleteOrderById(*int* id)  {  *return* **orderDao**.deleteOrderById(id);  }  *//查询所有订单  public* List<order> getAllorder()  {  *return* **orderDao**.getAllorder();  }   @Override  *public* List<historyorder> getAllHistoryorder() {  *return* **orderDao**.getAllHistoryorder();  }    @Override  *public* historyorder getHistoryorderById(*int* id) {  *return* **orderDao**.getHistoryorderById(id);  }   @Override  *public void* editOrder(*int* id,String recevier, String recevierAddress, *long* receviePhone) {  **orderDao**.editOrder(id,recevier,recevierAddress,receviePhone);  }  }  userServiceImpl.java  *package* com.mongodemov3.service.Impl;  *import* com.mongodemov3.config.MongoDBAppender; *import* com.mongodemov3.dao.userDao; *import* com.mongodemov3.pojo.\*; *import* com.mongodemov3.service.userService; *import* com.mongodemov3.utils.CommonUtils; *import* org.slf4j.Logger; *import* org.slf4j.LoggerFactory; *import* org.springframework.beans.factory.annotation.Autowired; *import* org.springframework.data.mongodb.core.MongoTemplate; *import* org.springframework.data.mongodb.core.query.Criteria; *import* org.springframework.data.mongodb.core.query.Query; *import* org.springframework.data.mongodb.core.query.Update; *import* org.springframework.stereotype.Service;  *import* java.util.List;  @Service *public class* userServiceImpl *implements* userService {  *private static* Logger *logger*= LoggerFactory.getLogger(userServiceImpl.*class*);  *private static* Logger *logs* = LoggerFactory.getLogger(CommonUtils.*class*);  *private static* Logger *log* = LoggerFactory.getLogger(MongoDBAppender.*class*);  @Autowired  MongoTemplate **mongoTemplate**;   @Autowired  userDao **userDao**;   @Override  *public void* adduser(user user) {   **userDao**.adduser(user);  }   @Override  *public void* addexuser(UserExtendInfo userExtendInfo) {  **userDao**.addexuser(userExtendInfo);  }     @Override  *public void* addLogs(LogBean logBean) {  **userDao**.addLogs(logBean);  }   @Override  *public void* addLog(Log log) {  **userDao**.addLog(log);  }   @Override  *public* user getUserById(*int* id) {  *return* **userDao**.getUserById(id);  }   @Override  *public boolean* deleteUserById(*int* id) {  *return* **userDao**.deleteUserById(id);  }   @Override  *public* List<user> getAlluser() {  *return* **userDao**.getAlluser();  }   @Override  *public* List<UserExtendInfo> getAllexuser() {  *return* **userDao**.getAllexuser();  }   @Override  *public* List<userall> getAlluserall() {  *return* **userDao**.getAlluserall();  }   @Override  *public void* editUser(*int* id, String nickname, *int* age, String sex) {  **userDao**.editUser(id,nickname,age,sex);  }   @Override  *public void* editexUser(*int* id, String phone, String school, String address) {  **userDao**.editexUser(id ,phone,school,address);  } }   1. 后端dao层代码   orderDao.java  *package* com.mongodemov3.dao;  *import* com.mongodemov3.pojo.\*;  *import* java.util.List;  *public interface* orderDao {  *//添加订单  void* addorder(order order);  *//添加物流日志信息  void* addLogisticsAndUpdateStatus(Logistics logistics);  *//添加日志信息：方法、请求名称、IP信息等  void* addLogs(LogBean logBean);  *//添加日志信息：使用logback(框架日志信息系统实现)  void* addLog(Log log);  *//按照ID查找订单* order getOrderById(*int* id);  *//按照ID删除订单  boolean* deleteOrderById(*int* id);  *//展示所有订单信息* List<order> getAllorder();  *//历史订单* List<historyorder> getAllHistoryorder();  *//按ID查找历史订单信息* historyorder getHistoryorderById(*int* id);  *//修改订单信息  void* editOrder(*int* id,String recevier,String recevierAddress,*long* receviePhone); }  userDao.java  *package* com.mongodemov3.dao;  *import* com.mongodemov3.pojo.\*;  *import* java.util.List;  *public interface* userDao {  *//添加用户基本信息  void* adduser(user user);  *//添加用户拓展信息  void* addexuser(UserExtendInfo userExtendInfo);  *//添加日志信息：方法、请求名称、IP信息等  void* addLogs(LogBean logBean);  *//添加日志信息：使用logback(框架日志信息系统实现)  void* addLog(Log log);  *//按ID查找用户（实现有问题）* user getUserById(*int* id);  *//按ID删除用户信息  boolean* deleteUserById(*int* id);  *//展示所有用户基本信息* List<user> getAlluser();  *//展示用户拓展信息* List<UserExtendInfo> getAllexuser();  *//展示用户所有信息* List<userall> getAlluserall();  *//修改用户基本信息  void* editUser(*int* id,String nickname,*int* age,String sex);  *//修改用户拓展信息  void* editexUser(*int* id,String phone,String school,String address); }  orderDaoImpl.java  *package* com.mongodemov3.dao.Impl;  *import* com.mongodemov3.config.MongoDBAppender; *import* com.mongodemov3.dao.orderDao; *import* com.mongodemov3.pojo.\*; *import* com.mongodemov3.service.Impl.orderServiceImpl; *import* com.mongodemov3.utils.CommonUtils; *import* org.slf4j.Logger; *import* org.slf4j.LoggerFactory; *import* org.springframework.beans.factory.annotation.Autowired; *import* org.springframework.data.mongodb.core.MongoTemplate; *import* org.springframework.data.mongodb.core.query.Criteria; *import* org.springframework.data.mongodb.core.query.Query; *import* org.springframework.data.mongodb.core.query.Update; *import* org.springframework.stereotype.Repository;  *import* java.util.List; @Repository *public class* orderDaoImpl *implements* orderDao {  *private static* Logger *logger*= LoggerFactory.getLogger(orderServiceImpl.*class*);  *private static* Logger *logs* = LoggerFactory.getLogger(CommonUtils.*class*);  *private static* Logger *log* = LoggerFactory.getLogger(MongoDBAppender.*class*);  @Autowired  MongoTemplate **mongoTemplate**;  @Override  *public void* addorder(order order) {  **mongoTemplate**.insert(order,"order");  **mongoTemplate**.insert(order,"historyorder");  }   @Override  *public void* addLogisticsAndUpdateStatus(Logistics logistics) {  String status=logistics.getOperation();  Query query = *new* Query(Criteria.where("\_id").is(logistics.getOrderId()));  Update update = *new* Update();  update.set("status", status);*//更新状态* update.push("Logistics",logistics);  **mongoTemplate**.upsert(query, update, order.*class*);  **mongoTemplate**.upsert(query, update, historyorder.*class*);  *//mongoTemplate.upsert(query, update, user.class);* }   @Override  *public void* addLogs(LogBean logBean) {  Query query = *new* Query(Criteria.where("\_id").is(logBean.getId()));  Update update = *new* Update();  update.push("(LogBean",logBean);  **mongoTemplate**.upsert(query, update, order.*class*);  *//mongoTemplate.upsert(query, update, user.class);* }   @Override  *public void* addLog(Log log) {  Query query = *new* Query(Criteria.where("createdTime").is(log.getCreatedTime()));  Update update = *new* Update();  update.push("Log",log);  **mongoTemplate**.upsert(query, update, order.*class*);  *//mongoTemplate.upsert(query, update, user.class);* }   @Override  *public* order getOrderById(*int* id) {  Query query = *new* Query(Criteria.where("\_id").is(id));  order order=**mongoTemplate**.findOne(query, order.*class*);  *return* order;  }   @Override  *public boolean* deleteOrderById(*int* id) {  Query query = *new* Query(Criteria.where("\_id").is(id));  **mongoTemplate**.remove(query, order.*class*,"order");  *return true*;  }   @Override  *public* List<order> getAllorder() {  List<order>list=**mongoTemplate**.findAll(order.*class*,"order");  *return* list;  }   @Override  *public* List<historyorder> getAllHistoryorder() {  List<historyorder>list=**mongoTemplate**.findAll(historyorder.*class*,"historyorder");  *return* list;  }   @Override  *public* historyorder getHistoryorderById(*int* id) {  Query query = *new* Query(Criteria.where("\_id").is(id));  historyorder historyorder=**mongoTemplate**.findOne(query, historyorder.*class*);  *return* historyorder;  }   @Override  *public void* editOrder(*int* id, String recevier, String recevierAddress, *long* receviePhone) {  order order = *new* order();   Query query = *new* Query(Criteria.where("\_id").is(id));  *//System.out.println(query);* Update update = *new* Update();  update.set("recevier",recevier);  update.set("recevierAddress",recevierAddress);  update.set("receviePhone", receviePhone);  *//System.out.println(update);* **mongoTemplate**.upsert(query, update, "order");  *//mongoTemplate.findAndModify(query,update, order.class);  //mongoTemplate.save(update,"order");* } }  userDaoImpl.java  *package* com.mongodemov3.dao.Impl;  *import* com.mongodemov3.config.MongoDBAppender; *import* com.mongodemov3.dao.userDao; *import* com.mongodemov3.pojo.\*; *import* com.mongodemov3.service.Impl.userServiceImpl; *import* com.mongodemov3.utils.CommonUtils; *import* org.slf4j.Logger; *import* org.slf4j.LoggerFactory; *import* org.springframework.beans.factory.annotation.Autowired; *import* org.springframework.data.mongodb.core.MongoTemplate; *import* org.springframework.data.mongodb.core.query.Criteria; *import* org.springframework.data.mongodb.core.query.Query; *import* org.springframework.data.mongodb.core.query.Update; *import* org.springframework.stereotype.Repository;  *import* java.util.List; @Repository *public class* userDaoImpl *implements* userDao {   *private static* Logger *logger*= LoggerFactory.getLogger(userServiceImpl.*class*);  *private static* Logger *logs* = LoggerFactory.getLogger(CommonUtils.*class*);  *private static* Logger *log* = LoggerFactory.getLogger(MongoDBAppender.*class*);  @Autowired  MongoTemplate **mongoTemplate**;  @Override  *public void* adduser(user user) {  **mongoTemplate**.insert(user,"user");  }   @Override  *public void* addexuser(UserExtendInfo userExtendInfo) {  user user =*new* user();  Query query = *new* Query(Criteria.where("\_id").is(userExtendInfo.getId()));  Update update = *new* Update();  update.set("codeNum", userExtendInfo.getCodeNum());  update.set("phone", userExtendInfo.getPhone());  update.set("school", userExtendInfo.getSchool());  update.set("address", userExtendInfo.getAddress());  update.push("user",user);  *//mongoTemplate.insert(userExtendInfo,"user");* **mongoTemplate**.upsert(query,update, user.*class*);  }   @Override  *public void* addLogs(LogBean logBean) {  Query query = *new* Query(Criteria.where("\_id").is(logBean.getId()));  Update update = *new* Update();  update.push("(LogBean",logBean);  *//mongoTemplate.upsert(query, update, order.class);* **mongoTemplate**.upsert(query, update, user.*class*);  }   @Override  *public void* addLog(Log log) {  Query query = *new* Query(Criteria.where("createdTime").is(log.getCreatedTime()));  Update update = *new* Update();  update.push("Log",log);  *//mongoTemplate.upsert(query, update, order.class);* **mongoTemplate**.upsert(query, update, user.*class*);  }   @Override  *public* user getUserById(*int* id) {  Query query = *new* Query(Criteria.where("\_id").is(id));  System.***out***.println(id);  System.***out***.println(query);  user user= **mongoTemplate**.findOne(query, user.*class*);  System.***out***.println(user);  *return* user;  }   @Override  *public boolean* deleteUserById(*int* id) {  Query query = *new* Query(Criteria.where("\_id").is(id));  **mongoTemplate**.remove(query, userall.*class*,"user");  *return true*;  }   @Override  *public* List<user> getAlluser() {  List<user>list=**mongoTemplate**.findAll(user.*class*,"user");  *return* list;  }   @Override  *public* List<UserExtendInfo> getAllexuser() {  List<UserExtendInfo>list=**mongoTemplate**.findAll(UserExtendInfo.*class*,"user");  *return* list;  }   @Override  *public* List<userall> getAlluserall() {  List<userall>list=**mongoTemplate**.findAll(userall.*class*,"user");  *return* list;  }   @Override  *public void* editUser(*int* id, String nickname, *int* age, String sex) {  user user = *new* user();  Query query = *new* Query(Criteria.where("\_id").is(id));  System.***out***.println(query);  Update update = *new* Update();  update.set("nickname",nickname);  update.set("age",age);  update.set("sex", sex);  System.***out***.println(update);  **mongoTemplate**.upsert(query, update, "user");  }   @Override  *public void* editexUser(*int* id, String phone, String school, String address) {  UserExtendInfo userExtendInfo = *new* UserExtendInfo();  Query query = *new* Query(Criteria.where("\_id").is(id));  System.***out***.println(query);  Update update = *new* Update();  update.set("phone",phone);  update.set("school",school);  update.set("address", address);  System.***out***.println(update);  **mongoTemplate**.upsert(query, update, "user");  } }   1. 日志配置文件logback-spring.xml   *<?xml version*="1.0" *encoding*="UTF-8"*?> <!-- This configuration added to prevent logback default of DEBUG output -->* <configuration>  *<!--定义日志文件的存储地址 勿在 LogBack 的配置中使用相对路径-->* <property *name*="LOG\_HOME" *value*="./log"/>  *<!-- 设置变量。定义变量后，可以使“${}”来使用变量。 -->* <property *name*="CONSOLE\_LOG\_PATTERN"  *value*="%red(%date{yyyy-MM-dd HH:mm:ss}) %highlight(%-5level) %red([%thread]) %boldMagenta(%logger{50}) %cyan(%msg%n)"/>  <appender *name*="console" *class*="ch.qos.logback.core.ConsoleAppender">  <encoder>  <pattern>%red(%4d{yyyy-MM-dd HH:mm:ss.fff}) [%thread] %-5p - %c - %m%n</pattern>  <pattern>${CONSOLE\_LOG\_PATTERN}</pattern>  </encoder>  </appender>  *<!-- ch.qos.logback.core.rolling.RollingFileAppender 文件日志输出 -->* <appender *name*="file" *class*="ch.qos.logback.core.rolling.RollingFileAppender">  <File>${LOG\_HOME}/room\_1.log</File>  <rollingPolicy *class*="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">  <FileNamePattern>${LOG\_HOME}/room\_1-%d{yyyy-MM-dd}-%i.log</FileNamePattern>  *<!--指定日志保存天数-->* <MaxHistory>20</MaxHistory>  <TimeBasedFileNamingAndTriggeringPolicy *class*="ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP">  *<!--指定每个日志文件大小-->* <MaxFileSize>5MB</MaxFileSize>  </TimeBasedFileNamingAndTriggeringPolicy>  </rollingPolicy>  <encoder>  <pattern>%4d{yyyy-MM-dd HH:mm:ss} [%thread] %-5p - %c - %m%n</pattern>  </encoder>  </appender>   *<!--指定包内日志打印级别-->* <logger *name*="org.gars" *level*="debug"/>  <logger *name*="org.apache.commons.httpclient" *level*="error"/>   <root *level*="info">  <appender-ref *ref*="console"/>  <appender-ref *ref*="file"/>  </root>  <appender *name*="mongo" *class*="com.mongodemov3.config.MongoDBAppender">  </appender>   <root *level*="info">  <appender-ref *ref*="console"/>  <appender-ref *ref*="file"/>  <appender-ref *ref*="mongo"/>  </root> </configuration> | | | |
| 6.实验结论及心得  通过本次实验实现了一个小型电商系统的一些基础功能，深入运用了mongo与Java代码的连接，并使用了springboot框架作为后端代码实现框架，前端也使用了一些新技术实现。深入熟悉了mongo的功能及存储模式。收获颇丰。 | | | |