《校园二手市场》代码清单

目录：

1. 项目描述

1．1标识

1．2系统概述

1.3引用文件

1. 目录结构

2.1后端目录结构说明

2.2前端目录结构说明

**1.项目描述**

1.1标识

1.2系统概述

校园二手市场软件项目旨在为浙大城市学院师生提供二手物品交易的平台，用户可以在平台上上传生活中的二手物品并进行交易。

1.3引用文件

**2.目录结构**

2.1后端目录结构说明

用户管理：@RestController

@RequestMapping("user")

@Api(tags = "用户表")

public class UserController {

    /\*\*

     \* 对象

     \*/

    @Resource

    private UserDao userService;

    /\*\*

     \* 通过主键查询单条数据

     \*

     \* @param id 主键

     \* @return 单条数据

     \*/

    @ApiOperation(value = "查询单条数据")

    @PostMapping("getone")

    public AjaxResult getone(Integer id) {

        return AjaxResult.success(userService.queryById(id));

    }

    /\*\*

     \* 查询列表数据

     \*

     \* @return 列表数据

     \*/

    @ApiOperation(value = "查询列表数据")

    @RequestMapping(value = "getlist", method = RequestMethod.POST)

    @ResponseBody

    public AjaxResult getlist(@RequestBody User user) {

        PageHelper.startPage(user.getOffset(), user.getLimit());

        List<Map<String, Object>> list = userService.queryAll(user);

        PageInfo<Map<String, Object>> pageInfo = new PageInfo<>(list);

        return AjaxResult.success(pageInfo);

    }

    /\*\*

     \* 删除

     \*

     \* @return 是否成功

     \*/

    @ApiOperation(value = "删除数据")

    @PostMapping("del")

    public AjaxResult del(Integer id) {

        Integer row = userService.deleteById(id);

        return row > 0 ? AjaxResult.success() : AjaxResult.fail();

    }

    /\*\*

     \* 保存

     \*

     \* @return 是否成功

     \*/

    @ApiOperation(value = "保存数据")

    @RequestMapping(value = "save", method = RequestMethod.POST)

    @ResponseBody

    public AjaxResult save(@RequestBody User user) {

        Integer row = 0;

        Date d = new Date();

        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd kk:mm:ss ");

        user.setTime(sdf.format(d));

        if (user.getId() == null) {

            row = userService.insert(user);

        } else {

            row = userService.update(user);

        }

        return row > 0 ? AjaxResult.success() : AjaxResult.fail();

    }

}

订单管理：：@RestController

@RequestMapping("orders")

@Api(tags = "订单表")

public class OrdersController {

    /\*\*

     \* 对象

     \*/

    @Resource

    private OrdersDao ordersService;

    /\*\*

     \* 通过主键查询单条数据

     \*

     \* @param id 主键

     \* @return 单条数据

     \*/

    @ApiOperation(value = "查询单条数据")

    @PostMapping("getone")

    public AjaxResult getone(Integer id) {

        return AjaxResult.success(ordersService.queryById(id));

    }

    /\*\*

     \* 查询列表数据

     \*

     \* @return 列表数据

     \*/

    @ApiOperation(value = "查询列表数据")

    @RequestMapping(value = "getlist", method = RequestMethod.POST)

    @ResponseBody

    public AjaxResult getlist(@RequestBody Orders orders) {

        PageHelper.startPage(orders.getOffset(), orders.getLimit());

        List<Map<String, Object>> list = ordersService.queryAll(orders);

        PageInfo<Map<String, Object>> pageInfo = new PageInfo<>(list);

        return AjaxResult.success(pageInfo);

    }

    /\*\*

     \* 删除

     \*

     \* @return 是否成功

     \*/

    @ApiOperation(value = "删除数据")

    @PostMapping("del")

    public AjaxResult del(String code) {

        Integer row = ordersService.deleteById(code);

        return row > 0 ? AjaxResult.success() : AjaxResult.fail();

    }

    /\*\*

     \* 保存

     \*

     \* @return 是否成功

     \*/

    @ApiOperation(value = "保存数据")

    @RequestMapping(value = "save", method = RequestMethod.POST)

    @ResponseBody

    public AjaxResult save(@RequestBody Orders orders) {

        Integer row = 0;

        Date d = new Date();

        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd kk:mm:ss ");

        orders.setTime(sdf.format(d));

        if (orders.getId() == null) {

            row = ordersService.insert(orders);

        } else {

            row = ordersService.update(orders);

        }

        return row > 0 ? AjaxResult.success() : AjaxResult.fail();

    }

}

物品管理：@RestController

@RequestMapping("goods")

@Api(tags = "商品表")

public class GoodsController {

    /\*\*

     \* 对象

     \*/

    @Resource

    private GoodsDao goodsService;

    /\*\*

     \* 通过主键查询单条数据

     \*

     \* @param id 主键

     \* @return 单条数据

     \*/

    @ApiOperation(value = "查询单条数据")

    @PostMapping("getone")

    public AjaxResult getone(Integer id) {

        return AjaxResult.success(goodsService.queryById(id));

    }

    /\*\*

     \* 查询列表数据

     \*

     \* @return 列表数据

     \*/

    @ApiOperation(value = "查询列表数据")

    @RequestMapping(value = "getlist", method = RequestMethod.POST)

    @ResponseBody

    public AjaxResult getlist(@RequestBody Goods goods) {

        PageHelper.startPage(goods.getOffset(), goods.getLimit());

        List<Map<String, Object>> list = goodsService.queryAll(goods);

        PageInfo<Map<String, Object>> pageInfo = new PageInfo<>(list);

        return AjaxResult.success(pageInfo);

    }

    /\*\*

     \* 删除

     \*

     \* @return 是否成功

     \*/

    @ApiOperation(value = "删除数据")

    @PostMapping("del")

    public AjaxResult del(Integer id) {

        Integer row = goodsService.deleteById(id);

        return row > 0 ? AjaxResult.success() : AjaxResult.fail();

    }

    /\*\*

     \* 保存

     \*

     \* @return 是否成功

     \*/

    @ApiOperation(value = "保存数据")

    @RequestMapping(value = "save", method = RequestMethod.POST)

    @ResponseBody

    public AjaxResult save(@RequestBody Goods goods) {

        Integer row = 0;

        Date d = new Date();

        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd kk:mm:ss ");

        goods.setTime(sdf.format(d));

        if (goods.getId() == null) {

            row = goodsService.insert(goods);

        } else {

            row = goodsService.update(goods);

        }

        return row > 0 ? AjaxResult.success() : AjaxResult.fail();

    }

}

评论管理：

2.2前端目录结构说明

首页：<template>

  <div class="home p-r bg">

    <Head @search="getData" :showSearch="true" />

    <div class="home-content">

      <div class="logo mt20 mb20 f-r f-a-c f-j-s center">

        <div class="font-default font30"> </div>

        <el-input placeholder="请输入搜索内容" style="width:400px" v-model="search" class="search">

          <el-button slot="append" icon="el-icon-search" @click="getData"></el-button>

        </el-input>

      </div>

      <el-carousel height="400px" arrow="never" class="center">

        <el-carousel-item v-for="item in slideList" :key="item">

          <el-image style="width: 100%;height:100%;" :src="item" fit="cover"></el-image>

        </el-carousel-item>

      </el-carousel>

      <div class="row-father center pt20">

        <div class="rows">

          <div v-for="(goods,index) in goodslist" :key="index" class="row zs" @click="$router.push('/goods?id='+goods.id)">

            <div class="row-content pb10 p10">

              <img style="height:200px;width:100%" :src="goods.img" lazy="loaded">

              <div class="font14 t-c mt20 font-hidden">

                {{goods.name}}

              </div>

              <p class="font14 font-black t-c">{{goods.price}}元</span>

              </p>

            </div>

          </div>

        </div>

      </div>

    </div>

    <Footer />

  </div>

</template>

<script>

import Head from "@/components/Head";

import Footer from "@/components/Footer";

import \* as api from "@/api/api";

export default {

  components: { Head, Footer },

  data() {

    return {

      offset: 1, //当前页

      limit: 10, //每页大小

      total: 0, //总记录数

      hidden: false, //分页是否隐藏b

      slideList: [require("@/assets/bg1.jpg"), require("@/assets/bg2.jpg")],

      goodslist: [],

      search: "",

    };

  },

  created() {

    this.getData();

  },

  mounted() {},

  methods: {

    //获取商品

    getData() {

      api

        .goodslist({

          name: this.search,

          offset: this.offset,

          limit: this.limit,

          status: "启用",

        })

        .then((res) => {

          if (res.code == 200) {

            this.goodslist = res.data.list;

          }

        });

    },

  },

  computed: {},

};

</script>

<style lang="scss" scoped>

.center {

  width: 1200px;

  left: 50%;

  margin-left: -600px;

  position: relative;

}

.home {

  background-color: #c41a21;

  .home-content {

    overflow: hidden;

    ::v-deep .el-carousel\_\_button {

      width: 10px;

      height: 10px;

      border-radius: 50%;

    }

    .logo {

      .search {

        input {

          border-radius: 0px;

        }

        ::v-deep .el-input\_\_inner {

          border-top-left-radius: 20px;

          border-bottom-left-radius: 20px;

          background: #f0f0f0;

          border: none;

        }

        ::v-deep .el-input-group\_\_append {

          background: #f0f0f0;

          border-radius: 0px;

          border-top-right-radius: 20px;

          border-bottom-right-radius: 20px;

          border: none;

          button {

            border-radius: 0px;

          }

        }

      }

    }

  }

  .rows {

    overflow: hidden;

    margin-left: -20px;

    .row {

      img {

        object-fit: cover;

      }

      .row-content {

        background: #fff;

      }

      width: 20%;

      float: left;

      padding-left: 20px;

      margin-bottom: 20px;

    }

  }

}

.bor {

  border-bottom: 2px solid #f23038;

}

.types {

  width: 100%;

  overflow-x: scroll;

  &::-webkit-scrollbar {

    width: 7px;

    height: 0px;

  }

}

.type-row {

  height: 50px;

  line-height: 50px;

  min-width: 120px;

}

.type-default {

  color: #ffe8cd;

  &:hover {

    color: #fff;

  }

}

</style>

购物车：<template>

  <div class="home bg">

    <Head :showSearch="false" />

    <Middle :showBread="false" :showSearch="false">

      <div slot class="content m20" v-show="tableData.length">

        <el-table :data="tableData" style="width: 100%" header-align="center" height="850px" @selection-change="handleSelectionChange">

          <el-table-column type="selection" width="55" align="center" />

          <el-table-column prop="date" label="商品" width="180" align="center">

            <template slot-scope="scope">

              <img :src="scope.row.img" style="width:150px;height:100px;object-fit:contain" align="center">

            </template>

          </el-table-column>

          <el-table-column prop="name" label="名称" width="180" align="center" />

          <el-table-column prop="type" label="分类" width="180" align="center" />

          <el-table-column prop="price" label="单价" align="center" />

          <el-table-column prop="date" label="数量" width="190" align="center">

            <template slot-scope="scope">

              <el-input-number v-model="scope.row.buynum" @change="handleChange(scope.row.buynum,scope.$index)" :min="1" :max="scope.row.num" label="描述文字"></el-input-number>

            </template>

          </el-table-column>

          <el-table-column prop="price" label="操作" align="center">

            <template slot-scope="scope" class="t-c" align="center">

              <span class="font-default zs" @click="del(scope.$index)">删除</span>

            </template>

          </el-table-column>

        </el-table>

        <div class="f-r f-a-c f-j-s">

          <div class="font-default pl20 zs" @click="delall">

            删除

          </div>

          <div class="f-r f-a-c f-j-r mt20">

            <div class="font14">总计金额：</div>

            <div class="font20 font-red m20">¥ {{sumPrice}}</div>

            <el-button type="danger" size="medium" class="btn" style="width:150px" @click="pay">结算</el-button>

          </div>

        </div>

      </div>

      <div slot class="content m20 f-c f-a-c f-j-c" v-show="!tableData.length" style="height:800px">

        <el-empty description="购物车为空"></el-empty>

        <el-button type="default" @click="$router.push('/')">返回首页</el-button>

      </div>

    </Middle>

    <Footer />

  </div>

</template>

<script>

import Head from "@/components/Head";

import Footer from "@/components/Footer";

import Middle from "@/components/Middle";

import \* as api from "@/api/api";

import { mapGetters, mapMutations } from "vuex";

import { Loading } from "element-ui";

export default {

  components: { Head, Middle, Footer },

  data() {

    return {

      tableData: [],

      show: false,

      multipleSelection: [],

    };

  },

  created() {},

  mounted() {

    if (!this.goods || this.goods.length == 0) {

      this.tableData = [];

    } else {

      this.tableData = this.goods;

    }

  },

  methods: {

    ...mapMutations(["setGoods"]),

    // 全选

    handleSelectionChange(val) {

      this.multipleSelection = val;

    },

    // 删除

    delall() {

      if (!this.multipleSelection.length) {

        this.$message.error("没有选中商品!");

        return;

      }

      let result = [];

      for (let i = 0; i < this.tableData.length; i++) {

        let obj = this.tableData[i];

        let id = obj.id;

        let isExist = false;

        for (let j = 0; j < this.multipleSelection.length; j++) {

          let aj = this.multipleSelection[j];

          let n = aj.id;

          if (n == id) {

            isExist = true;

            break;

          }

        }

        if (!isExist) {

          result.push(obj);

        }

      }

      this.tableData = result;

      this.setGoods(result);

    },

    //改变购物车数量

    handleChange(num, index) {

      this.tableData[index].buynum = num;

      this.setGoods(this.tableData);

    },

    //删除

    del(index) {

      this.tableData.splice(index, 1);

      this.$message.success("删除成功!");

      this.setGoods(this.tableData);

    },

    // 点击结算

    settlement() {

      this.show = true;

    },

    //下单

    async pay() {

      if (this.goods.length < 0) {

        return;

      }

      if (this.multipleSelection.length == 0) {

        this.$message.error("请选择结算物品!");

        return;

      }

      this.show = false;

      let loading = Loading.service({

        body: true,

        text: "提交订单中",

      });

      let code = this.$tool.random(15487456321, 98745621589);

      let promiseAll = [];

      for (const key in this.multipleSelection) {

        let params = {

          code: code,

          num: this.multipleSelection[key]["buynum"],

          gid: this.multipleSelection[key]["id"],

          uid: this.userinfo.id,

          status: "已下单",

        };

        promiseAll.push(await api.orderssave(params));

      }

      Promise.all(promiseAll).then((res) => {

        let result = [];

        for (let i = 0; i < this.tableData.length; i++) {

          let obj = this.tableData[i];

          let id = obj.id;

          let isExist = false;

          for (let j = 0; j < this.multipleSelection.length; j++) {

            let aj = this.multipleSelection[j];

            let n = aj.id;

            if (n == id) {

              isExist = true;

              break;

            }

          }

          if (!isExist) {

            result.push(obj);

          }

        }

        setTimeout(() => {

          loading.close();

          this.$message.success("下单成功!");

          this.tableData = result;

          this.setGoods(result);

        }, 1500);

      });

    },

  },

  computed: {

    ...mapGetters(["goods", "goodsNum", "userinfo"]),

    sumPrice() {

      let sum = 0;

      if (this.multipleSelection.length == 0) {

        return 0;

      }

      this.multipleSelection.map((item) => {

        sum += item.buynum \* item.price;

      });

      return sum;

    },

  },

};

</script>

<style lang="scss" scoped>

.home {

  background-color: #f6f6f6;

  height: 100%;

}

.bg {

  background-size: 100% 100%;

}

.content {

  padding: 20px;

  width: 1200px;

  position: relative;

  left: 50%;

  margin-left: -600px;

  background-color: #fff;

  // box-shadow: 0 2px 12px 0 rgba(0, 0, 0, 0.1);

  // height: 950px;

  overflow-y: scroll;

}

.btn {

  background-color: #c41a21;

  background: linear-gradient(90deg, #f23038 0%, #c41a21 100%);

  // height: 50px;

  // width: 200px;

}

.addr-row {

  width: 270px;

  height: 180px;

  border: 1px solid #e0e0e0;

}

.addr-active {

  border: 1px solid #c41a21;

}

.default {

  color: #b0b0b0;

}

</style>

我的订单：<template>

  <div class="home">

    <Head />

    <div class="f-r h100 mt20 my-menu center">

      <div class="my-left h100 mr20 pt40 pl40">

        <div class="my-left-row mb20 zs font15" @click="tabActive=item.component" :class="tabActive==item.component?'font-default':'font-grey'" v-for="item in tabbars" :key="item.name">{{item.name}}</div>

      </div>

      <Middle :showBread="false" :showSearch="false" class="my-right w100 h100">

        <div slot class="w100 h100">

          <component :is="tabActive" :name="item.component" v-for="item in tabbars" :key="item.component" v-if="tabActive==item.component" />

        </div>

      </Middle>

    </div>

    <Footer />

  </div>

</template>

<script>

import Head from "@/components/Head";

import Footer from "@/components/Footer";

import Middle from "@/components/Middle";

import Userorders from "./components/Userorders";

import \* as api from "@/api/api";

export default {

  components: { Head, Middle, Footer, Userorders },

  data() {

    return {

      tabActive: "Userorders",

      tabbars: [

        {

          component: "Userorders",

          name: "我的订单",

        },

      ],

    };

  },

  created() {

    if (this.$route.query.component) {

      this.tabActive = this.$route.query.component;

    }

  },

  watch: {

    "$route.query.component": {

      immediate: true,

      handler(newval, oldval) {

        if (newval != oldval) {

          this.tabActive = newval;

        }

      },

    },

  },

  mounted() {},

  methods: {},

  computed: {},

};

</script>

<style lang="scss" scoped>

.home {

  background-color: #f6f6f6;

  min-height: 800px;

  background-size: 100% 100%;

}

.my-menu {

  .my-left {

    width: 300px;

    height: 500px;

    background: #fff;

  }

  .my-right {

    padding: 20px;

    min-height: 700px;

    background-color: #fff;

  }

}

.center {

  width: 1200px;

  position: relative;

  left: 50%;

  margin-left: -600px;

}

</style>

物品界面：<template>

  <div class="home p-r bg">

    <Head @search="getData" :showSearch="true" />

    <div class="search-row">

      <div class="logo mt20 mb20 f-r f-a-c f-j-s">

        <div class="font-default font30"> </div>

      </div>

    </div>

    <div class="goods-content center">

      <el-breadcrumb separator-class="el-icon-arrow-right">

        <el-breadcrumb-item :to="{ path: '/' }">首页</el-breadcrumb-item>

        <el-breadcrumb-item>商品详情</el-breadcrumb-item>

        <el-breadcrumb-item>{{info.type}}</el-breadcrumb-item>

        <el-breadcrumb-item>{{info.name}}</el-breadcrumb-item>

      </el-breadcrumb>

      <div class="row1 mt20 f-r f-a-l ">

        <img :src="info.img" style="width:460px;height:500px;border: 1px #c7c7c7 solid;object-fit: cover">

        <div class="f-c ml30 w100 p-r" style="height:500px">

          <div class="font-w600 font22">{{info.name}}</div>

          <p>{{info.tip}}</p>

          <div class="price font14 p10 w100">

            售价

            <span class="font30 ml40">¥{{info.price}}</span>

          </div>

          <div class="type font14 p10 w100 font-grey">

            库存

            <span class="ml40 font-black">{{info.num}}</span>

          </div>

          <div class="p-a btn">

            <el-button type="danger" round class="car" @click="addcar(info)">加入购物车</el-button>

          </div>

        </div>

      </div>

      <div class="row2 mt20">

        <el-tabs v-model="activeName">

          <el-tab-pane label="商品详情" name="details" v-html="info.content" class="details-content"></el-tab-pane>

        </el-tabs>

      </div>

    </div>

    <Footer />

  </div>

</template>

<script>

import Head from "@/components/Head";

import Footer from "@/components/Footer";

import \* as api from "@/api/api";

import { mapGetters, mapMutations } from "vuex";

export default {

  components: { Head, Footer },

  data() {

    return {

      id: "",

      activeName: "details",

      rate: 5,

      info: {},

    };

  },

  created() {

    this.id = this.$route.query.id;

    this.getData();

  },

  computed: {

    ...mapGetters(["userinfo"]),

  },

  mounted() {},

  methods: {

    ...mapMutations(["add"]),

    //获取商品

    getData() {

      api

        .goodsone({

          id: this.id,

        })

        .then((res) => {

          if (res.code == 200) {

            this.info = res.data;

          }

        });

    },

    //加入购物车

    addcar(item) {

      if (Object.keys(this.userinfo).length == 0) {

        this.$message.error("请先登录!");

        return;

      } else {

        this.add(item);

      }

    },

  },

};

</script>

<style lang="scss" scoped>

.home {

  background-color: #fff;

  .search-row {

    width: 1200px;

    left: 50%;

    margin-left: -600px;

    position: relative;

    .logo {

      .search {

        input {

          border-radius: 0px;

        }

        ::v-deep .el-input\_\_inner {

          border-radius: 0px;

        }

        ::v-deep .el-input-group\_\_append {

          background-color: #fff;

          border-radius: 0px;

          button {

            border-radius: 0px;

          }

        }

      }

    }

  }

  .goods-content {

    min-height: 70vh;

    ::v-deep .el-tabs\_\_item {

      font-size: 16px;

    }

    ::v-deep .el-rate\_\_icon {

      font-size: 22px;

    }

    .row1 {

      .price {

        color: #fff;

        background-color: #c41a21;

        background: linear-gradient(90deg, #f23038 0%, #c41a21 100%);

        border-radius: 8px 8px 0 0;

      }

      .type {

        background-color: #f7f9fa;

      }

      .btn {

        bottom: 0;

        width: 300px;

        left: 50%;

        margin-left: -150px;

        button {

          border-radius: 30px;

        }

        .car {

          background-color: #c41a21;

          background: linear-gradient(90deg, #f23038 0%, #c41a21 100%);

          height: 50px;

          width: 200px;

        }

      }

    }

  }

}

.details-content {

  ::v-deep img {

    width: 100%;

  }

}

.center {

  width: 1200px;

  position: relative;

  left: 50%;

  margin-left: -600px;

}

</style>