**基于java的绿碳交易系统（核心代码）**

**登陆模块：**

<!-- 前端核心代码 -->

export default {

data() {

return {

form: {

// 输入账号

account: '',

// 输入密码

account\_pwd: '',

login\_from: 'lpzhaoshang\_h5',

code:'',

platform:''

}

}

},

onLoad() {

},

onShow() {

if(uni.getStorageSync('userAdmin')){

uni.switchTab({

url: '/pages/index/money'

});

}

},

methods: {

login(){

this.$u.api.login(this.form).then(res => {

// 判断是否正确 如果正确 跳转到收付款页面

if(res.code == 1){

this.$u.toast(res.message);

uni.setStorageSync('phone',res.data.phone)

uni.setStorageSync('userAdmin',res.data)

setTimeout(()=>{

uni.switchTab({

url: '/pages/index/money'

});

},500)

}else{

this.$u.toast(res.message);

}

})

}

}

}

// 接口核心代码

public function index(){

return view('login/index');

}

public function login(){

$password = $this->request->param('account\_pwd');

$username = $this->request->param('account');

$where=[

'phone'=>$username,

'password'=>$password

];

$users = new user();

if( !empty($username) || !empty($password) ){

$userdata = $users->where($where)->find();

if(!$userdata){

return show(2,'账号未注册或密码错误');

}else{

Session::set('username', $userdata);

return show(1,'登录成功',$userdata,200,1);

}

}else{

return show(2,'账号或者密码未填写');

}

**注册模块：**

<!-- 前端核心代码 -->

export default {

components: {

},

data() {

return {

checked: false,

form: {

// 输入手机号

mobile: '',

// 输入密码

account\_pwd: '',

// 输入验证码

code: '',

auth\_code: '',

},

codeTips: '',

}

},

onLoad() {},

onShow() {

},

methods: {

codeChange(text) {

this.codeTips = text;

},

getCode() {

// 判断手机号格式是否正确

if (this.$refs.uCode.canGetCode) {

if (!this.$u.test.mobile(this.form.mobile)) {

this.$u.toast('手机格式错误');

return;

}

this.$refs.uCode.start();

} else {

this.$u.toast('倒计时结束后再发送');

}

},

tojumps() {

uni.navigateTo({

url: '/pages/index/index'

})

},

submit() {

let that = this;

if (!that.$u.test.mobile(that.form.mobile)) {

that.$u.toast('手机格式错误');

return;

}

// 判断输入的验证码格式是否正确

if (!that.$u.test.code(that.form.code, 6)) {

that.$u.toast('请输入六位数字的验证码');

return;

}

this.$u.api.register(this.form).then(res => {

if (res.code == 1) {

this.$u.toast(res.message);

this.$u.vuex('vuex\_user\_token', '1111');

setTimeout(() => {

uni.navigateTo({

url: '/pages/index/login'

})

}, 500)

} else {

this.$u.toast(res.message);

}

})

},

}

}

// 接口核心代码

class Register extends BaseController

{

public function index(){

return view('login/reg');

}

public function save(){

$phone = $this->request->param('mobile');

$password = $this->request->param('account\_pwd');

// 验证密码是否为空

if(empty($password)){

return show(2,'密码不能为空');

}

$usermodel = new user();

$isusername = $usermodel->where(['phone'=>$phone])->find();

// 判断手机号是否已经存在

if($isusername != null){

return show(2,'当前手机号已经被注册',[],200,2);

}

else{

$data =[

'password'=>$password,

'phone'=>$phone

];

$message = $usermodel->save($data);

if ($message){

return show(1,'注册成功',[],200,);

}else{

return show(2,'注册失败',[],200,2);

}

}

}

}

**收付款模块：**

import tkiQrcode from "@/components/tki-qrcode/tki-qrcode.vue"

export default{

components: {tkiQrcode},

data(){

return{

checked: false,

cid: 'tki-qrcode-canvas', // 二维码唯一ID

val: 'https://blog.csdn.net/Dark\_programmer', // 要生成的二维码值

size: 400, // 二维码大小

unit: 'upx', // 单位

background: '#000000', // 背景色

foreground: '#ffffff', // 前景色

pdground: '#ffffff', // 角标色

icon: '', // 二维码图标 二维码中心的icon图标

iconsize: 60, // 二维码中心icon图标大小

lv: 3, // 二维码容错级别 ， 一般不用设置，默认就行

onval: true, // val值变化时是否自动重新生成二维码

loadMake: true, // 组件加载完成后是否自动生成二维码

src: '', // 二维码生成后的图片地址或base64

usingComponents: false, // 是否使用了自定义组件模式

showLoading: true, // 是否显示loading

loadingText: '二维码生成中', // loading展示文案

}

},

methods:{

change(status) {

// console.log(status);

},

},

onLoad() {

this.val = '付款码' + '\n' + '用户名:'+ uni.getStorageSync('userAdmin')['name'] + '\n' +'用户id:'+uni.getStorageSync('userAdmin')['id']+ '\n' +'用户手机号:'+uni.getStorageSync('userAdmin')['phone']

}

}

**交易记录模块：**

<!-- 前端核心代码 -->

<view class="card" v-for="item in datalist" :key="item.id">

<view class="biaotiimage"><image :src="'http://weixin.com/'+item.images" mode=""></image></view>

<view class="neirong">

<view class="biaoti1"><text>商品消费{{item.price}}积分</text></view>

<view class="biaoti3"><text>{{item.created\_time}}</text></view>

</view>

<view class="biaoti4"><text>-{{item.price}}.00</text></view>

</view>

<!-- 前端核心代码 -->

export default{

data(){

return{

sum:"",

datalist:[]

}

},

onShow(){

this.getdata()

},

methods: {

getdata(){

this.$u.api.orderall().then(res => {

this.datalist = res.data

this.sum = this.datalist.reduce((x,y)=>x+parseInt(y.price),0);

})

},

}

}

// 接口核心代码

public function orderall($id=""){

$promodel = new Promoneymodel();

$shopmodel = new \app\web\model\shop();

$userdata = Session::get('username');

if(!$userdata){

return show(3,'登录过期，请重新登录');

}

$order = Db::name('order')->where('uid',$userdata['id'])->select();

$ordernumber = [];

foreach($order as $k=>&$vo){

$ordernumber[] = $vo['orderid'];

}

$str = implode(',',$ordernumber);

$shopmdata = Db::name('shop')->select();

$data=[];

$address = Db::name('address')->select();

$list =[];

$addresslist = [];

if($id !=""){

$data = $promodel->where('id',$id)->where(array(array('ordername','in',$str)))->select();

}else{

$data = $promodel->where(array(array('ordername','in',$str)))->select();

}

foreach($shopmdata as $k=>&$vo){

$list[$vo['id']] = $vo;

}

foreach($address as $k=>&$vo){

$addresslist[$vo['id']] = $vo;

}

foreach ($data as $k=>&$vo){

$vo['images'] = $list[$vo['shopid']]['images'];

$vo['address'] = $addresslist[$vo['addressid']]['address'];

$vo['mobile'] = $addresslist[$vo['addressid']]['mobile'];

}

return show(1,'200',$data);

}

**个人信息模块:**

<!-- 前端核心代码 -->

export default{

data(){

return{

userdata:{},

mylet:{

"color":"#000",

"borderBottom":"#f5f5f5"

}

}

},

onLoad() {

// 根据注册时间，顺序信息生成独特的用户id

let id = uni.getStorageSync('userAdmin')['id'];

this.$u.api.userselect({id:id}).then(res => {

let datas = res.data

this.userdata = datas

})

},

methods:{

jumpTo(url){

uni.navigateTo({

url

})

},

submit(){

console.log(this.userdata.name)

this.$u.api.saveuser(this.userdata).then(res => {

setTimeout(()=>{

uni.navigateBack()

},1500)

uni.showToast({

title:"信息保存成功"

})

})

},

// 上传头像

selectimage(){

uni.chooseImage({

success: (res) => {

let tempFilePaths = res.tempFilePaths;

tempFilePaths.forEach((item) => {

uni.uploadFile({

url: 'http://weixin.com/admin/uploads/upload', //服务器地址

fileType: "image", //ZFB必填,不然报错

filePath: item, //这个就是我们上面拍照返回或者先中照片返回的数组

name: 'file',

success: (uploadFileRes) => {

this.userdata['images'] = JSON.parse(uploadFileRes.data)['url']

console.log(this.userdata['images'])

}

});

})

}

});

}

}

}

// 接口核心代码

public function saveuser(){

$data = $this->request->param();

$usermodels = new usermodel();

if( $usermodels->update($data)){

return show(1,'信息保存成功');

}

}

**个人钱包模块:**

<!-- 前端核心代码 -->

export default{

data(){

return{

cartList:[

{id:1,name:'现金',lable:'',style:'#18b566',icon:'icf\_wallet\_open'},

{id:2,name:'积分',lable:'',style:'#bb4a96',icon:'jifenguanli'}

]

}

},

onLoad() {

let id = uni.getStorageSync('userAdmin')['id'];

this.$u.api.userselect({id:id}).then(res => {

let datas = res.data

this.cartList[0].lable = datas['moneys']

this.cartList[1].lable = datas['integral']

})

},

}

// 接口部分

// 获取用户信息

public function userselect($id=''){

$usermodel = new \app\web\model\user();

$data = [];

if($id !=''){

$data= $usermodel->where('id',$id)->find();

}else{

$data = $usermodel->select();

}

return show(1,200,$data);

}

**个人地址模块:**

<!-- 前端核心代码 地址列表页面 -->

export default {

data() {

return {

source: 0,

addressList: []

}

},

onShow(option){

let userid = uni.getStorageSync('userAdmin')['id'];

this.$u.api.getuseraddress({id:userid}).then(res => {

this.dataData = res.data

if(res.data.length>0){

this.addressList = res.data

console.log(useraddress)

}

})

},

onLoad(option){

console.log(option.source);

this.source = option.source;

},

methods: {

//选择地址

checkAddress(item){

if(this.source == 1){

uni.setStorageSync('address',item);

uni.navigateBack();

}

},

addAddress(type, item){

uni.navigateTo({

url: `/pages/index/addressMessage?type=${type}&data=${JSON.stringify(item)}`

})

},

//添加或修改成功之后回调

refreshList(data, type){

//添加或修改后事件，这里直接在最前面添加了一条数据，实际应用中直接刷新地址列表即可

this.addressList.unshift(data);

console.log(data, type);

}

}

}

// 获取用户地址信息

public function getuseraddress($id){

$addressmodels = new Address();

$data = $addressmodels->where('pid',$id)->select();

return show(1,200,$data);

}

<!-- 前端核心代码 新增地址页面 -->

export default {

data() {

return {

type:1,

addressData: {

name: '',

mobile: '',

address: '',

}

}

},

onLoad(option){

let title = '新增收货地址';

if(option.type==='edit'){

title = '编辑收货地址'

this.type = 2,

this.addressData = JSON.parse(option.data)

}

this.manageType = option.type;

uni.setNavigationBarTitle({

title

})

},

methods: {

switchChange(e){

this.addressData.default = e.detail;

},

//地图选择地址

chooseLocation(){

uni.chooseLocation({

success: (data)=> {

this.addressData.addressName = data.name;

this.addressData.address = data.name;

}

})

},

deleteconfirm(id){

if(uni.getStorageSync('address')){

if(id == uni.getStorageSync('address')['id']){

uni.setStorageSync('address',null)

}

}

this.$u.api.deteleAddress({id,id}).then(res => {

uni.showModal({

title:"确定要删除该地址吗？",

success() {

uni.showToast({

icon:"success",

title:res.message

})

setTimeout(()=>{

let pages = getCurrentPages();

let beforePage = pages[pages.length - 2];

beforePage.$vm.reFresh = Math.random()

uni.navigateBack()

}, 800)

}

})

})

},

//提交

confirm(){

let data = this.addressData;

if(!data.name){

uni.showToast({

icon:"error",

title:"请填写收货人姓名"

})

return;

}

if(!/(^1[3|4|5|7|8][0-9]{9}$)/.test(data.mobile)){

uni.showToast({

icon:"error",

title:"请输入正确的手机号码"

})

return;

}

if(!data.address){

uni.showToast({

icon:"error",

title:"请填写地址信息信息"

})

}

this.addressData['pid'] = uni.getStorageSync('userAdmin')['id'];

console.log(this.addressData)

this.$u.api.useraddress(this.addressData).then(res => {

uni.showToast({

icon:"success",

title:res.message

})

setTimeout(()=>{

uni.navigateBack()

}, 800)

})

//this.$api.prePage()获取上一页实例，可直接调用上页所有数据和方法，在App.vue定义

},

}

}

/\*\*

\* 地址增加

\*/

class Address extends BaseController

{

public function index(){

}

// 地址增加或修改

public function useraddress($id=""){

$data = $this->request->param();

$addressmodel1 = new addressmodel();

if($id == ''){

if($addressmodel1->save($data)){

return show(1,'保存成功');

}

}else{

if($addressmodel1->update($data)){

return show(1,'更新成功');

}

}

}

public function deteleAddress($id){

$addressmodel1 = new addressmodel();

if($addressmodel1->where('id',$id)->delete()){

return show(1,'删除成功');

}

}

}

**商品信息模块:**

<!-- 前端核心代码 -->

export default{

data(){

return{

current:'',

tabactive:'',

list:[],

tabs: []

}

},

methods:{

tabChange(index,pid) {

this.tabactive = pid

this.current = index;

this.getData(1)

},

getData(params){

let wheres = {}

if(params == 0){

wheres ={

'pid': ''

}

}else{

wheres ={

'pid': this.tabactive

}

}

this.$u.api.shoplist(wheres).then(res => {

this.tabs =res.data.tablist

this.tabs.unshift({

name:'全部',

id:'0',

})

this.list = res.data.shoplist

})

},

},

onShow() {

this.getData()

}

}

// 获取商品列表数据

public function shoplist($pid=""){

$where=[];

if($pid !='' && $pid !=0 ){

$where=[

"pid"=> $pid

];

}

$colummodel = new colum();

$shopmodel = new modelShop();

$tabData = $colummodel->select();

$datalist = $shopmodel->where($where)->select();

$shopdata=[

'tablist'=> $tabData, 'shoplist'=> $datalist

];

return show(1,200,$shopdata);

}

/\*\*

\* 商品详情

\*/

public function shopdetails()

{

$id = $this->request->param('id');

$where=[];

if($id){

$where = [

'id'=>$id

];

}

$shopmodel = new modelShop();

$datalist = $shopmodel->where($where)->find();

return show(1,200,$datalist);

}

**积分消费模块:**

export default {

data() {

return {

useraddress: {},

dataData: {},

maskState: 0, //优惠券面板显示状态

desc: '', //备注

payType: 1, //1微信 2支付宝

}

},

onLoad(options) {

let shopid = options.id;

this.$u.api.shopdetails({

id: shopid

}).then(res => {

this.dataData = res.data

})

},

onShow() {

this.getaddress()

},

methods: {

goAddress(){

uni.navigateTo({

url:`/pages/index/address?source=1`

})

},

getaddress() {

let userid = uni.getStorageSync('userAdmin')['id'];

let addresslist = uni.getStorageSync('address')

if (addresslist) {

this.useraddress = addresslist

} else {

this.$u.api.getuseraddress({

id: userid

}).then(res => {

if (res.data.length > 0) {

this.useraddress = res.data[0]

}

})

}

},

//显示优惠券面板

toggleMask(type) {

let timer = type === 'show' ? 10 : 300;

let state = type === 'show' ? 1 : 0;

this.maskState = 2;

setTimeout(() => {

this.maskState = state;

}, timer)

},

numberChange(data) {

this.number = data.number;

},

changePayType(type) {

this.payType = type;

},

submit() {

if(JSON.stringify(this.useraddress)=='{}'){

uni.showToast({

title:'请填写收货地址'

})

}

let moneys = uni.getStorageSync('userAdmin')['integral']

if ((moneys-this.dataData.newprice)>0) {

let form={

addressid:this.useraddress['id'],

addressname:this.useraddress['name'],

shopname:this.dataData['name'],

shopid:this.dataData['id'],

created\_id:new Date(),

price:this.dataData['newprice']

}

this.$u.api.orderbuy(form).then(res => {

if(res.status ==1 ){

setTimeout(()=>{

uni.navigateBack()

},1500)

uni.showToast({

title:res.message

})

}

})

}else{

uni.showToast({

icon: "error",

title: "该用户积分不足，不足以支付"

})

}

},

stopPrevent() {}

}

}

// 订单付款金钱

public function orderbuy(){

$data = $this->request->param();

$userdata = Session::get('username');

if(!$userdata){

return show(3,'登录过期，请重新登录');

}

// 用户积分

$user\_integral = $userdata['integral'];

//用户id

$user\_id = $userdata['id'];

$userprice = $user\_integral-$data['price'];

if($userprice<0){

return show(2,'该用户积分不够');

}

$ordernumber = time().rand(1000,99999);

$order = [

'uid' => $user\_id,

'orderid' => $ordernumber,

'creat\_time'=> date("Y-m-d",time())

];

// 保存订单编号

$ordermodel = new order();

if($ordermodel->save($order)){

// 更新用户金钱积分信息

$usermodel = new user();

$userfrom =[

'id'=>$user\_id,

'integral'=>$userprice

];

if($usermodel->update($userfrom)){

// 保存订单信息

$Promoney = new Promoney();

$data['ordername'] = $ordernumber;

$data['created\_time'] = date("Y-m-d",time());

if($Promoney->save($data)){

return show(1,'购买成功');

}

}

return show(2,'保存用户信息失败');

}

}

**订单查询模块:**

export default {

data() {

return {

// 这里修改图标

// logoimg: '/static/icon-logo.png',

dname:"绿碳交易系统",

current: 0,

scrollLeft: 0,

isClickChange: false,

currentTab: 0,

menuTabs: [

{

name: '待发货'

}, {

name: '已发货'

}, {

name: '已完成'

},

],

orderList:[],

menuLists: [

[

{

"dname":"自营Apple产品专营店",

"zt":"已取消",

"img": '../../static/pd-001.png',

"name":"Apple 苹果 iPhone8 Plus 4G手机 深空灰 移动联通版64G裸机Apple 苹果 iPhone8 Plus 4G手机 深空灰 移动联通版64G裸机",

"sum":"1",

"pri":"6899.00",

// 0=没有,1=有,2=标红

"but\_rebuy":2,

"but\_pingjia":1,

"but\_fapiao":1,

"but\_zhifu":0,

"but\_wuliu":0,

"but\_ddshouhuo":0,

},

],

]

}

},

computed:{

labelLength() {

// 筛选数组下标name内的内容，只返回传进来的值相同的数组

return this.orderList.filter(value => {

if(this.current == value.status ){

return value

}

})

}

},

onLoad() {

this.getdata()

},

methods: {

getdata(){

this.$u.api.orderall().then(res => {

this.orderList = res.data

})

},

comfirmSave(shopid){

let that = this

uni.showModal({

title:"确定收货吗？",

success() {

that.$u.api.saveStatus({id:shopid}).then(res => {

setTimeout(()=>{

that.getdata()

},1000)

uni.showToast({

title:res.message

})

})

}

})

},

gotodetails(userid){

uni.navigateTo({

url:`/pages/index/orderdetails?id=${userid}`

})

},

change(index) {

this.current = index;

},

getWidth: function(id) { //得到元素的宽高

return new Promise((res, rej) => {

uni.createSelectorQuery().select("#" + id).fields({

size: true,

scrollOffset: true

}, (data) => {

res(data);

}).exec();

})

},

getDateList: function(tabIndex) {

// var entity = this.menuTabs[tabIndex].name;

// this.menuLists[tabIndex].push(entity);

},

}

}

// 查找所有订单

public function orderall($id=""){

$promodel = new Promoneymodel();

$shopmodel = new \app\web\model\shop();

$userdata = Session::get('username');

if(!$userdata){

return show(3,'登录过期，请重新登录');

}

$order = Db::name('order')->where('uid',$userdata['id'])->select();

$ordernumber = [];

foreach($order as $k=>&$vo){

$ordernumber[] = $vo['orderid'];

}

$str = implode(',',$ordernumber);

$shopmdata = Db::name('shop')->select();

$data=[];

$address = Db::name('address')->select();

$list =[];

$addresslist = [];

if($id !=""){

$data = $promodel->where('id',$id)->where(array(array('ordername','in',$str)))->select();

}else{

$data = $promodel->where(array(array('ordername','in',$str)))->select();

}

foreach($shopmdata as $k=>&$vo){

$list[$vo['id']] = $vo;

}

foreach($address as $k=>&$vo){

$addresslist[$vo['id']] = $vo;

}

foreach ($data as $k=>&$vo){

$vo['images'] = $list[$vo['shopid']]['images'];

$vo['address'] = $addresslist[$vo['addressid']]['address'];

$vo['mobile'] = $addresslist[$vo['addressid']]['mobile'];

}

return show(1,'200',$data);

}

// 确定收货

/\*\*

\* @return bool

\*/

public function saveStatus($id)

{

$promodel = new Promoneymodel();

$data=[

'id'=>$id,

'status'=>2

];

if($promodel->where('id',$id)->update($data)) {

return show(1,'收货成功');

}

}

}