```
    <!-- 注册页面 -->
    <!--personal/pages/phoneLogin/phoneLogin.wxml-->
    <view class="container ">
    <image class="logo" src="/image/common/icon_login@2x.png"></image>
    <text class="text" wx:if="{{pageFrom.from=='no-username'}}">由于系统更新，请
先完善账号</text>
    <text class="text" wx:else>卧涛商城</text>
    <view class="form">
    <view class="phone {{accountError?'error':''}}">
    <text class="label-width">账号名</text>
    <input class="input height-48" bindblur="checkAccount" value="{{account}}"
bindinput="getAccount" placeholder="4~16位纯字母(或字母+数字组合)"
placeholder-class="holder-style"/>
    </view>
    <view wx:if="{{accountError}}" class="phoneError">{{error[2]}}</view>
    <view class="phone {{phoneError?'error':''}}">
    <text class="label-width">手机号码</text>
    <input type="number" class="input height-48" disabled="{{ pageFrom.mobile }}"
value="{{pageFrom.mobile ? pageFrom.mobile : phoneNumber}}" bindblur="checkPhone"
bindinput="getPhone" placeholder="请输入绑定的手机号码"
placeholder-class="holder-style"/>
    </view>
    <view wx:if="{{phoneError}}" class="phoneError">{{error[0]}}</view>
    <view class="phone check {{codeError?'error':''}}">
    <text class="label-width">验证码</text>
    <input bindinput="getCode" type="number" class="input height-48 short-width"
bindblur="checkCode"  placeholder="请输入验证码"
placeholder-class="holder-style"/>
    <button type="primary" plain="true" bindtap="sendMsg"
class="{{btnText.length>6?'long-width':''}}"
disabled="{{btnDisabled&&!pageFrom.mobile}}">{{btnText}}</button>
    </view>
    <view wx:if="{{codeError}}" class="phoneError">{{error[1]}}</view>
    <view class="phone {{passwordError?'error':''}}">
    <text class="label-width">密码</text>
    <input type="{{eyesOn ? 'text' : 'password'}}" class="input height-48"
bindblur="checkPassword" value="{{password}}" bindinput="getword"
placeholder="6~16位字母+数字或符号的组合密码"
placeholder-class="holder-style"/>
    <image src="{{eyesOn ? '/image/index/show_eye.png':
'/image/index/hide_eye.png'}}" class="img" bindtap="eyeStatus"></image>
    </view>
    <view wx:if="{{wordError}}" class="phoneError">{{error[3]}}</view>
    <view class="phone {{passwordError?'error':''}}">
    <text class="label-width">确认密码</text>
    <input type="{{eyesOnAgain ? 'text' : 'password'}}" class="input height-48"
bindblur="checkPasswordAgain" value="{{passwordAgain}}" bindinput="getwordAgain"
placeholder="6~16位字母+数字或符号的组合密码"
placeholder-class="holder-style"/>
    <image src="{{eyesOnAgain ? '/image/index/show_eye.png':
'/image/index/hide_eye.png'}}" class="img" bindtap="eyeStatusAgain"></image>
    </view>
    <view wx:if="{{wordErrorAgain}}" class="phoneError">{{error[4]}}</view>
    <view class="agreement">
    <checkbox-group bindchange="checkboxChange">
    <checkbox value="agree"  class="checkbox" color="#fff" />注册后自动登录
    </checkbox-group>
    </view>
    </view>
    <button  bindtap="phoneRegist" class="loginBtn
{{loginDisabled?'errorLogin':''}}">{{pageFrom.from=='no-username'? '确认' : '注
册'}}</button>
    </view>
    <!-- 认证提示弹窗 -->
    <Tips bind:cancel='cancel' wx:if="{{tipStatus}}"></Tips>
    js 代码：
    const app = getApp();
    const phoneLogin = require('../../../utils/api.js').phoneLogin;
    const testCode = require('../../../utils/api.js').testCode;
    import {
    getUserInfos
    } from "../../../utils/common/userDeal";
    import {
    hexMD5
    } from "../../../utils/md5.js";
    import {
    userRegister,
    completeOldUserInfo
    } from "../../../utils/api/user"
    Page({
    data: {
    linkId: "",
    tipStatus: false,
    phoneNumber: "",
    account: "",
    error: [
    "请输入正确的手机号码",
    "请输入验证码",
    "请输入 4~16 位纯字母（或字母+数字组合）",
    '请输入 6~16 字母加数字或特殊符号组合的密码',
    '密码输入错误，请重新输入！'
    ],
    phoneError: false,
    codeError: false,
    wordError: false,
    wordErrorAgain: false,
    accountError: false,
    codeTime: "",
    btnDisabled: true,
    btnText: "获取验证码",
    codeTime: 0,
    loginDisabled: true,
    code: "",
    embedToken: "",
    password: '',
    passwordAgain: '',
    pageFrom: {},
    eyesOn: false,
    eyesOnAgain: false,
    // from: "launch-order",
    // mobile: "18019935577"
    },
    openAgreement() {
    wx.navigateTo({
    url: '/personal/pages/agreement/agreement',
    })
    },
    openSecrecy() {
    wx.navigateTo({
    url: '/personal/pages/secrecy/secrecy',
    })
    },
    //获取用户输入框数据
    getPhone(e) {
    this.setData({
    phoneNumber: e.detail.value.trim()
    })
    let reg = /^1[3-9]\d{9}$/;
    if (reg.test(this.data.phoneNumber) && this.data.btnText == "获取验证码") {
    this.setData({
    btnDisabled: false
    })
    } else {
    this.setData({
    btnDisabled: true
    })
    }
    },
    getCode(e) {
    this.setData({
    code: e.detail.value.trim()
    })
    },
    getAccount(e) {
    this.setData({
    account: e.detail.value.trim()
    })
    },
    getword(e) {
    this.setData({
    password: e.detail.value.trim()
    })
    },
    eyeStatus() {
    this.setData({
    eyesOn: !this.data.eyesOn
    })
    },
    eyeStatusAgain() {
    this.setData({
    eyesOnAgain: !this.data.eyesOnAgain
```

```
    })
    },
    //协议勾选
    checkboxChange(e) {
    console.log(e.detail.value)
    if (e.detail.value.length == 0) {
    this.setData({
    loginDisabled: true
    })
    } else {
    this.setData({
    loginDisabled: false
    })
    }
    },
    //手机号校验
    checkPhone() {
    let reg = /^1[3-9]\d{9}$/;
    if (reg.test(this.data.phoneNumber)) {
    this.setData({
    phoneError: false,
    })
    } else {
    this.setData({
    phoneError: true,
    })
    }
    },
    //校验账号
    checkAccount() {
    // let reg = /^(?![0-9]+$)(?![a-zA-Z]+$)[a-zA-Z][0-9A-Za-z_-]{5,15}$/;
    let reg = /^(?![0-9]+$)[a-zA-Z0-9]{4,16}$/;
    if (reg.test(this.data.account)) {
    this.setData({
    accountError: false,
    })
    } else {
    this.setData({
    accountError: true,
    })
    }
    },
    //校验密码
    checkPassword() {
    let reg =
/^(?![\d]+$)(?![a-zA-Z]+$)(?![_/#\\@*&%]+$)[\da-zA-Z_/#\\@*&%]{6,16}$/;
    if (reg.test(this.data.password)) {
    this.setData({
    wordError: false,
    })
    } else {
```

```
    this.setData({
    wordError: true,
    })
    }
    },
    getwordAgain(e) {
    this.setData({
    passwordAgain: e.detail.value.trim()
    })
    },
    checkPasswordAgain() {
    let password = this.data.password
    let passwordAgain = this.data.passwordAgain
    if (password == passwordAgain) {
    this.setData({
    wordErrorAgain: false,
    })
    } else {
    this.setData({
    wordErrorAgain: true,
    })
    }
    },
    //验证码非空校验
    checkCode() {
    this.setData({
    codeError: this.data.code ? false : true,
    })
    },
    //获取验证码
    sendMsg() {
    let that = this;
    let mobile = that.data.pageFrom.mobile ? that.data.pageFrom.mobile :
that.data.phoneNumber
    testCode(mobile).then(res => {
    if (res.code == 0) {
    wx.showToast({
    title: "验证码已发送",
    icon: 'success',
    duration: 2000
    })
    that.setData({
    codeTime: 60,
    btnDisabled: true
    })
    that.timer();
    } else if (res.code == 1) {
    wx.showToast({
    title: res.data.cnMsg,
    icon: 'none',
    duration: 2000
```

```
})
}
})
},
timer() {
if (this.data.codeTime > 0) {
this.setData({
codeTime: this.data.codeTime - 1,
btnText: this.data.codeTime + "s后重新获取",
})
setTimeout(this.timer, 1000);
} else {
this.setData({
btnDisabled: false,
btnText: "获取验证码"
})
}
},
//登录
phoneRegist() {
if (this.data.account == "") return app.toastBox('请输入账号名！', 'none', 1500);
if (!this.data.pageFrom.mobile && this.data.phoneNumber == "") return app.toastBox('请输入手机号码！', 'none', 1500);
if (this.data.code == "") return app.toastBox('请输入验证码！', 'none', 1500);
if (this.data.password == "") return app.toastBox('请输入密码！', 'none', 1500);
if (this.data.passwordAgain == "" || this.data.passwordAgain != this.data.password) return app.toastBox('请确认密码！', 'none', 1500);
let data = {
mobile: this.data.pageFrom.mobile ? this.data.pageFrom.mobile : this.data.phoneNumber,
authCode: this.data.code,
password: hexMD5(this.data.password),
registerClient: 2,
username: this.data.account,
};
if (this.data.linkId !== "") {
data.inviteCode = this.data.linkId
}
let url = this.data.pageFrom.from == 'no-username' ? completeOldUserInfo : userRegister
let _this = this
url(data).then(res => {
if (res.code == 0) {
wx.showToast({
title: this.data.pageFrom.from == 'no-username' ? '修改成功！' : '注册成功！',
icon: 'success',
duration: 1000
});
if (!this.data.loginDisabled) {
if (url == userRegister) {
```

```
wx.setStorageSync(’token’, res.data)
}
getUserInfos(’openId’);
setTimeout(() => {
if (_this.data.pageFrom.from == ’launch-order’) return app.toGoalUrl(’/order/pages/launch/launch?from=regLog’);
_this.checkCodeStatus()
}, 1000)
} else {
setTimeout(() => {
return app.toGoalUrl(’/personal/pages/wxLogin/wxLogin’);
}, 1000)
}
} else {
wx.showToast({
title: res.msg,
icon: ’none’,
duration: 1500
})
}
})
},
// 扫码返回判断
checkCodeStatus() {
if (wx.getStorageSync(’codeStatus’)) {
wx.navigateBack({
delta: 2,
})
wx.removeStorageSync(’codeStatus’)
} else {
app.toTabUrl(’/pages/me/me’)
}
},
// 取消弹窗
cancel() {
this.setData({
tipStatus: false
})
wx.navigateBack({
delta: 2
})
},
// 检查用户是否认证，且是否存在待认证的订单
checkCertify(userInfo) {
let params = {
accountId: userInfo.accountId
}
getUncertifiedOrder(params).then(res => {
if ((res.code == 1) && (userInfo.certificationStatus !== ’1’)) {
this.setData({
tipStatus: true
```

```
    })
    } else {
    this.cancel()
    }
    })
    },
    onLoad: function (options) {
    if (options.data) {
    this.setData({
    pageFrom: JSON.parse(options.data),
    })
    if(this.data.pageFrom.from=='no-username'){
    wx.setNavigationBarTitle({
    title: "完善信息"
    })
    }
    }
    if (wx.getStorageSync('linkId')) {
    this.setData({
    linkId: wx.getStorageSync('linkId')
    })
    console.log(this.data.linkId + "-linkId")
    }
    },
    onShow() {
    // this.lauchOrder()
    }
    })
    <!-- 登录页面 -->
    wxml 代码：
    <!--personal/pages/phoneLogin/phoneLogin.wxml-->
    <view class="container ">
    <image class="logo" src="/image/common/icon_login@2x.png"></image>
    <text class="text">卧涛商城</text>
    <view class="tab-sty">
    <view class="word-sty {{currentTab==0?'select-sty':''}}"  data-current="0"
bindtap="switchTab">手机号码登录</view>
    <view wx:if="{{!pageFrom.mobile}}" class="word-sty
{{currentTab==1?'select-sty':''}}" data-current="1" bindtap="switchTab">账号密码
登录</view>
    </view>
    <!-- 手机号码登录 -->
    <view wx:if="{{currentTab==0}}">
    <view class="form">
    <view class="phone {{phoneError?'error':''}}">
    <text class="label-width">手机号码</text>
    <input type="number" class="input" disabled="{{pageFrom.mobile}}"
bindblur="checkPhone" bindinput="getPhone" value="{{pageFrom.mobile}}"
placeholder-class="holder-style" placeholder="请输入手机号码"/>
    </view>
    <view wx:if="{{phoneError}}" class="phoneError">{{error[0]}}</view>
    <view class="phone check {{codeError?'error':''}}">
    <text class="label-width">验证码</text>
    <input bindinput="getCode" type="number" class="input check-input"
bindblur="checkCode"  placeholder-class="holder-style" placeholder="请输入验证
码"/>
    <button type="primary" plain="true" bindtap="sendMsg"
disabled="{{btnDisabled&&!pageFrom.mobile}}">{{btnText}}</button>
    </view>
    <view wx:if="{{codeError}}" class="phoneError">{{error[1]}}</view>
    </view>
    <button  bindtap="phoneLogin" class="loginBtn
{{loginDisabled?'errorLogin':''}}">登录</button>
    <view class="popup" hidden="{{mask}}">
    <view class="mask" catchtouchmove="preventTouchMove"
catchtap="closeMask"></view>
    <view class="popup_main">
    <view class="relative-fix">
    <view class="pop-title">
    <text>选择一个关联账号登录</text>
    <image catchtap="closeMask" class="close-ico"
src="{{imageUrl+'icon/head/b0b594b92ed64062b013ab47da74b923.png'}}"></image>
    </view>
    <view class="scroll-item">
    <view class="user-item" wx:for="{{accountList}}" wx:key="index"
catchtap="chooseOne" data-index="{{index}}">
    <image wx:if="{{userIndex == index&&item.isDisabled!=1}}"
src="/personal/image/center/radio_check.png"></image>
    <image wx:else src="/personal/image/center/radio_gray.png"></image>
    <text class="{{item.isDisabled==1?'gray-txt':''}}">{{item.username ?
item.username : '暂无用户名(登录后请尽快完善)' }}</text>
    <text class="warn-text" wx:if="{{item.isDisabled==1}}">该账号已禁用</text>
    </view>
    </view>
    <view class="note-text" wx:if="{{userIndex == -1}}">该手机号码绑定的所有账号
都已被禁用，如有需要请重新注册或联系客服处理</view>
    <!-- <view class="submit-btn {{userIndex == -1?'gray-btn':''}}"
catchtap="submitLogin">确 定</view> -->
    <view class="fixed-btn"><view class="submit-btn {{userIndex ==
-1?'gray-btn':''}}" catchtap="submitLogin">确 定</view></view>
    </view>
    </view>
    <!-- <view class="fixed-btn"><view class="submit-btn {{userIndex ==
-1?'gray-btn':''}}" catchtap="submitLogin">确 定</view></view> -->
    </view>
    </view>
    <!-- 账号密码登录 -->
    <view wx:if="{{currentTab==1}}">
    <accountLogin loginDisabled="{{loginDisabled}}"></accountLogin>
    </view>
    </view>
    <view class="agreement">
    <checkbox-group bindchange="checkboxChange">
    <checkbox value="agree" class="checkbox" color="#fff" /><text class="color3">
登录即表示同意 </text>
    <text class="agreement-link" bindtap="openAgreement">《卧涛用户协议》</text>
```

```
和<text class="agreement-link" bindtap="openSecrecy">《隐私保护声明》</text>
    </checkbox-group>
    </view>
    <!-- 认证提示弹窗 -->
    <Tips bind:cancel='cancel’ wx:if="{{tipStatus}}"></Tips>
```

js 代码：

```
    const app = getApp();
    const testCode = require(’../../../utils/api.js’).testCode;
    import {
    getUserInfos
    } from "../../../utils/common/userDeal"
    import {
    loginBySMS,
    chooseAccount
    } from "../../../utils/api/user"
    import {
    getUncertifiedOrder
    } from "../../../utils/api/order.js"
    Page({
    options: {
    addGlobalClass: true,  //采用全局样式
    },
    data: {
    linkId: "",
    tipStatus: false,
    phoneNumber: "",
    currentTab: 0,
    error: [
    "请输入正确的手机号码",
    "请输入验证码",
    ],
    phoneError: false,
    codeError: false,
    btnDisabled: false,
    btnText: "获取验证码",
    codeTime: 0,
    loginDisabled: true,
    code: "",
    embedToken: "",
    mask: true,
    imageUrl: app.globalData.imageURL,
    accountList: [],
    userIndex: null,
    loginTemp: '',
    pageFrom: {}
    },
    preventTouchMove(e) {},
    closeMask() {
    this.setData({
    mask: true,
    })
    },
    chooseOne(e) {
    let index = e.currentTarget.dataset.index;
    if(this.data.accountList[index].isDisabled==1) return app.toastBox('该账号已禁用','error',1500)
    this.setData({
    userIndex: index
    })
    },
    openAgreement() {
    wx.navigateTo({
    url: '/personal/pages/agreement/agreement',
    })
    },
    openSecrecy() {
    wx.navigateTo({
    url: '/personal/pages/secrecy/secrecy',
    })
    },
    //获取用户输入框数据
    getPhone(e) {
    this.setData({
    phoneNumber: e.detail.value.trim()
    })
    let reg = /^1[3-9]\d{9}$/;
    if (reg.test(this.data.phoneNumber) && this.data.btnText == "获取验证码") {
    this.setData({
    btnDisabled: false
    })
    } else {
    this.setData({
    btnDisabled: true
    })
    }
    },
    getCode(e) {
    this.setData({
    code: e.detail.value.trim()
    })
    },
    //协议勾选
    checkboxChange(e) {
    console.log(e);
    if (e.detail.value.length == 0) {
    this.setData({
    loginDisabled: true
    })
    } else {
    this.setData({
    loginDisabled: false
    })
    }
    },
    //手机号校验
    checkPhone() {
    let reg = /^1[3-9]\d{9}$/;
    if (reg.test(this.data.phoneNumber)) {
    this.setData({
    phoneError: false,
    })
    } else {
    this.setData({
    phoneError: true,
    })
    }
    },
    //验证码非空校验
    checkCode() {
    this.setData({
    codeError: this.data.code ? false : true,
    })
    },
    let that = this;
    let mobile = that.data.pageFrom.mobile ? that.data.pageFrom.mobile :
that.data.phoneNumber
    testCode(mobile).then(res => {
    if (res.code == 0) {
    wx.showToast({
    title: "验证码已发送",
    icon: 'success',
    duration: 1000
    })
    that.setData({
    codeTime: 60,
    btnDisabled: true
    })
    that.timer();
    } else if (res.code == 1) {
    wx.showToast({
    title: res.data.cnMsg,
    icon: 'none',
    duration: 1000
    })
    }
    })
    },
    timer() {
    if (this.data.codeTime > 0) {
    this.setData({
    codeTime: this.data.codeTime - 1,
    btnText: this.data.codeTime + "s 后重新获取",
    })
    setTimeout(this.timer, 1000);
    } else {
    this.setData({
    btnDisabled: false,
    btnText: "获取验证码"
    })
    }
    },
    //登录
    phoneLogin() {
    var that = this;
    if (this.data.loginDisabled) return app.toastBox('请勾选协议！', 'none', 1500);
    let linkId = that.data.linkId
    if (!this.data.codeError && !this.data.phoneError) {
    wx.login({
    success: function (result) {
    let data = {
    mobile: that.data.pageFrom.mobile ? that.data.pageFrom.mobile : that.data.phoneNumber,
    authcode: that.data.code,
    // lastLoginWayType: 1 //标记上次登录的方式用于初始化回显：0-微信登录，1-手机号登录
    };
    if (linkId !== "") {
    data.inviteCode = linkId
    }
    loginBySMS(data).then(res => {
    if (res.code == 0) {
    let list = res.data.list;
    let ableIndex = list.findIndex(item => item.isDisabled==0);
    that.setData({
    userIndex: ableIndex,
    accountList: list,
    loginTemp: res.data.loginTemp
    })
    if (list.length > 1 && !that.data.pageFrom.accountId) {
    if (that.data.pageFrom.mobile) return that.filterAccount(list);
    that.setData({
    mask: false,
    })
    } else {
    that.submitLogin();
    }
    } else {
    wx.showToast({
    title: res.msg,
    icon: 'none',
    duration: 3000
    })
    }
    })
    }
    })
    }
    },
    filterAccount(val) {
    let that = this;
    let listFilter = val.filter((item) => {
    return item.authStatus == '0’
    });
    if (listFilter.length == 0) return wx.showModal({
    title: '操作提示',
    content: '当前手机号无关联未认证账号，请再去注册一个吧~',
    confirmText: '现在就去',
    showCancel: false,
    success(res) {
    if (res.confirm) {
    let para = {
    from: 'launch-order',
    mobile: that.data.pageFrom.mobile,
    }
    that.toReg(para)
    } else if (res.cancel) {
    that.triggerEvent('close-dialog’, true)
    }
    }
    });
    this.setData({
    accountList: listFilter,
    mask: false,
    })
    },
    toReg(para) {
    let data = JSON.stringify(para)
    app.toGoalUrl(`/personal/pages/regist/regist?data=${data}`)
    },
    // 选择账号登录
    submitLogin() {
    console.log(this.data.userIndex);
    if(this.data.userIndex == -1) return app.modalBox(null,'账号异常，检查账号信
息或联系客服处理！',false);
    let that = this;
    let list = this.data.accountList;
    let index = list.length == 1 ? 0 : this.data.userIndex;
    let para = {
    accountId: this.data.pageFrom.accountId ? this.data.pageFrom.accountId :
list[index].accountId,
    loginTemp: this.data.loginTemp,
    };
    chooseAccount(para).then(res => {
    if (res.code == 0) {
    wx.setStorageSync('token’, res.data);
    this.setData({
```

```
    mask: true,
    });
    getUserInfos('openId', function (res) {
    console.log(res.username);
    if (res.username) {
    if (that.data.pageFrom.mobile &&
wx.getStorageSync('launchUser').mobileStatus == '授权中') return
app.toGoalUrl('/personal/pages/authentication/authentication')
    if (that.data.pageFrom.mobile) return
app.toGoalUrl('/order/pages/launch/launch?from=regLog')
    setTimeout(() => {
    that.checkCodeStatus()
    }, 500)
    } else {
    console.log(2);
    let para = {
    from: 'no-username',
    mobile: that.data.phoneNumber,
    }
    that.toReg(para)
    }
    })
    }
    })
    },
    // 扫码返回判断
    checkCodeStatus() {
    if (wx.getStorageSync('codeStatus')) {
    wx.navigateBack({
    delta: 2,
    })
    wx.removeStorageSync('codeStatus')
    } else {
    app.toTabUrl('/pages/me/me')
    }
    },
    // 取消弹窗
    cancel() {
    this.setData({
    tipStatus: false
    })
    wx.navigateBack({
    delta: 2
    })
    },
    // 检查用户是否认证，且是否存在待认证的订单
    checkCertify(userInfo) {
    let params = {
    accountId: userInfo.accountId
    }
    getUncertifiedOrder(params).then(res => {
    if ((res.code == 1) && (userInfo.certificationStatus !== '1')) {
```

```
this.setData({
tipStatus: true
})
} else {
this.cancel()
}
})
},
// 切换标签页
switchTab(e) {
console.log(e)
this.setData({
currentTab: e.currentTarget.dataset.current
})
},
/**
* 生命周期函数--监听页面加载
*/
onLoad: function (options) {
console.log(options);
if (options.data) {
this.setData({
pageFrom: JSON.parse(options.data),
})
};
if (wx.getStorageSync('linkId')) {
this.setData({
linkId: wx.getStorageSync('linkId')
})
console.log(this.data.linkId + "-linkId")
}
},
})
<!-- 产品分类列表 -->
```

wxml 代码:

```
<wxs src="/utils/common/filters.wxs" module="handles"></wxs>
<!-- 遮罩样式 -->
<view wx:if="{{openFilter}}" class="global-tip" bindtap="closeTip"></view>
<view class="classify-content">
<!-- 回到顶部 -->
<view wx:if="{{showTop}}">
<image class="toTop" bindtap="toTop" src="/image/common/icon_back@2x.png" />
</view>
<!-- 搜索页面 -->
<view class="top-fixed">
<view class="search-contain">
<view class="search-contain-left" bindtap="skipToArea">
<text>{{handles.handleStr(province)}}</text>
<image src="../../image/index/area_icon.png"></image>
</view>
<text class="lang-sty"></text>
```

```
    <view class="search-contain-right" bindtap="skipToGlobal">
    <text>搜索想要的内容</text>
    </view>
    </view>
    <text bindtap="skipToGlobal" class="fixed-search">搜索</text>
    </view>
    <!-- 内容区域 -->
    <view class="home-content">
    <!-- 左侧固定导航区域 -->
    <view class="content-left" style="min-height:{{contentHeight}}">
    <view wx:for="{{serviceList}}" wx:key="index"
class="{{newIndex==index?'choosed':''}}">
    <view catchtap="bindChange" data-index="{{index}}" data-item="{{item}}">
    <text wx:if="{{newIndex==index}}" class="choosed-img"></text>
    <text
class="{{newIndex==index?'choosed-text':''}}">{{item.categoryName}}</text>
    </view>
    </view>
    </view>
    <view class="content-right">
    <!-- 新产品 -->
    <view class="project-content">
    <!-- 筛选条件选择 -->
    <view class="fixed-content" wx:if="{{classFilter.length>0}}">
    <scroll-view scroll-x="true" class="scroll-contain" enhanced="{{true}}"
show-scrollbar="{{false}}">
    <view class="new-scroll">
    <view class="scroll-item" wx:for="{{classFilter}}" wx:key="index"
bindtap="changeNew" data-index="{{index}}" data-item="{{item}}">
    <text class="{{((cFilterId == index)&&openFilter||item.selectChildId!='')?
'active-scroll-sty':''}}">{{item.attributeName}}</text>
    <text class="{{(cFilterId == index)&&openFilter?
'active-angle-sty':''}}"></text>
    </view>
    </view>
    </scroll-view>
    <!-- 选择 option 区域 -->
    <view class="option-list
{{classFilter[cFilterId].id==2?'pad1':classFilter[cFilterId].id==3?'pad2':''}}"
wx:if="{{openFilter}}">
    <text class="option-item {{classFilter[cFilterId].selectChildId ==
'' ?'active-option':''}}" bindtap="changeOption" data-item="{{null}}">
    <text wx:if="{{classFilter[cFilterId].id==1}}">{{province}}</text>
    <text wx:else>全部</text>
    </text>
    <text class="option-item {{classFilter[cFilterId].selectChildId ==
item.id ?'active-option':''}}" bindtap="changeOption" data-item="{{item}}"
wx:for="{{optionList}}" wx:key="index">{{item.categoryName}}</text>
    </view>
    </view>
    <!-- 产品列表 -->
    <view class="items-list {{classFilter.length>0?'':'active-items-list'}}"
wx:if="{{newTotal>0}}">
```

```
<view class="project-item" data-number="{{item.number}}" wx:for="{{newProductList}}" wx:key="index">
<!-- 其他产品 item -->
<otherItem projectInfo="{{item}}" index="{{cProId}}"></otherItem>
</view>
</view>
<view class="items-list {{classFilter.length>0?'':'active-items-list'}}" wx:else>
<view class="no-data">
暂无匹配数据
</view>
</view>
</view>
</view>
</view>
</view>
<!-- 省市区弹窗  -->
<areaItem bind:getCityName="getCityName" areaDialog="{{areaDialog}}" region="{{region}}" areaValue="{{areaValue}}"></areaItem>
```

js 代码：

```
const app = getApp();
import {
getNewClassify,
getNewProduct,
getProjectAttribute
} from "../../utils/api/product"
import {
checkUrl
} from "../../utils/common/utils"
import {
getDistrict
} from "../../utils/api/search"
Page({
data: {
areaList: [],
levelList: [{
id: 1,
categoryName: '国家级'
},{
id: 2,
categoryName: '省级'
},{
id: 3,
categoryName: '市级'
},{
id: 4,
categoryName: '区级'
}],
// 选择的产品 id（如选择的是二级分类的筛选项，显示对应二级分类 id 值）
cProId: "",
reachStatus: false, // 是否触底加载更多数据
openFilter: false, // 是否显示遮罩 options
```

```
    cFilterId: 0,
    showTop: false,
    newTotal: 0, // 用于上拉加载显示判断
    page: 1,
    pageSize: 20,
    newProductList: [],
    newIndex: 0,
    province: wx.getStorageSync('province'),
    provinceId: wx.getStorageSync('provinceId'),
    contentHeight: (wx.getStorageSync('windowHeight') -
wx.getStorageSync('statusBarHeight') - wx.getStorageSync('navigationBarHeight')
- 5) + 'px', //5:顶部高度的padding-bottom:5px
    serviceList: [], // 左侧产品栏
    classFilter: [], // 对应产品筛选类（包含二级分类+属性）
    optionList: [], // 筛选数属性显示option集合
    list: "",
    searchCode: "0", //0-项目申报（某些地方0-知识产权，真是头疼）
    projectList: [],
    propertyList: [],
    copyrightList: "",
    patentList: "",
    trademarkList: "",
    otherList: "",
    district: "",
    areaDialog: false,
    areaValue: [],
    region: []
    },
    closeTip() {
    this.setData({
    openFilter: false
    })
    },
    changeNew(e) {
    let item = e.currentTarget.dataset.item
    let index = e.currentTarget.dataset.index
    this.setData({
    cFilterId: index,
    openFilter: true,
    page: 1,
    optionList: item.child
    })
    // 点击属性获取option数据 isAttr：true-是属性 false-是二级分类
    if (item.isAttr) {
    this.getProAttrData(false, item.id)
    }
    },
    // （分类+属性）option筛选赋值
    changeOption(e) {
    let item = e.currentTarget.dataset.item
    let cFilterId = this.data.cFilterId
    let newIndex = this.data.newIndex
    let serviceList = this.data.serviceList
    let classFilter = this.data.classFilter
    let classFilterItem = classFilter[cFilterId]
    let classId = classFilter[cFilterId].id
    let cId = serviceList[newIndex].id
    if (item == null) {
    classFilter[cFilterId].selectChildId = ''
    classFilter[cFilterId].attributeName = classId == 1 ? this.data.province :
(classFilterItem.isAttr == true ? classFilterItem.defaultName : '分类')
    this.setData({
    classFilter: classFilter,
    cProId: cId,
    })
    // 获取产品类表数据
    this.getProList()
    } else {
    classFilter[cFilterId].selectChildId = item.id
    classFilter[cFilterId].attributeName = item.categoryName
    this.setData({
    classFilter: classFilter,
    cProId: classFilterItem.isAttr == true ? cId : item.id,
    })
    // 获取产品类表数据
    this.getProList()
    }
    this.closeTip()
    },
    // 回到页面顶部
    toTop() {
    app.toTop()
    },
    // 跳转到项目申报列表
    toProject() {
    wx.navigateTo({
    url: '/search/pages/classifySearch/classifySearch',
    })
    },
    this.setData({
    areaDialog: true
    })
    },
    // 跳转到知产详情
    toPropertyDetail(e) {
    if (wx.getStorageSync('userInfo')) {
    let id = e.currentTarget.dataset.myid;
    wx.navigateTo({
    url: "/search/pages/propertyDetail/propertyDetail?propertyId=" + id
    })
    } else {
    wx.showToast({
    title: '您还未登录,请先登录吧~',
    icon: 'none',
    duration: 1000
    })
    setTimeout(function () {
    wx.navigateTo({
    url: "/personal/pages/wxLogin/wxLogin"
    })
    }, 1000)
    }
    },
    // 跳转到项目详情
    toProjectDetail(e) {
    if (wx.getStorageSync('userInfo')) {
    let number = e.currentTarget.dataset.number;
    wx.navigateTo({
    url: "/search/pages/projectDetail/projectDetail?number=" + number
    })
    } else {
    wx.showToast({
    title: '您还未登录,请先登录吧~',
    icon: 'none',
    duration: 1000
    })
    setTimeout(function () {
    wx.navigateTo({
    url: "/personal/pages/wxLogin/wxLogin"
    })
    }, 1000)
    }
    },
    // 跳转到全局搜索页面
    skipToGlobal() {
    if (wx.getStorageSync('userInfo')) {
    wx.navigateTo({
    url: '/search/pages/global/global',
    })
    } else {
    wx.showToast({
    title: '您还未登录,请先登录吧~',
    icon: 'none',
    duration: 1000
    })
    setTimeout(function () {
    wx.navigateTo({
    url: "/personal/pages/wxLogin/wxLogin"
    })
    }, 1000)
    }
    },
    // 跳转到 classifySearch 页面
    toClassifySearch(e) {
```

```
let id = e.currentTarget.dataset.id
wx.navigateTo({
url: `/search/pages/classifySearch/classifySearch?optionId=${id}`
})
},
search(e) {
wx.navigateTo({
url: '/search/pages/global/global?type=' + e.currentTarget.dataset.type,
});
},
bindChange(e) {
// 回到顶部
wx.pageScrollTo({
scrollTop: 0,
duration: 0
})
let index = e.currentTarget.dataset.index;
let item = e.currentTarget.dataset.item;
this.setData({
cProId: item.id,
newIndex: index,
cFilterId: 0,
openFilter: false,
classFilter: item.child.length > 0 ? [{
attributeName: "分类",
isAttr: false,
selectChildId: "", // 选择的option项Id
child: item.child
}] : [],
newProductList: [],
page: 1
})
this.getProList()
this.getAttributeData([item.id])
},
onShareAppMessage(res) {
return {
title: app.globalData.shareTitle, //默认小程序标题
path: app.globalData.sharePath
}
},
onShareTimeline() {
return {
title: app.globalData.shareTitle, //默认小程序标题
path: app.globalData.sharePath
}
},
// 获取选择的地区
getSelectArea() {
let provinceId = this.data.provinceId
let province = this.data.province
```

```
if (provinceId) {
if (wx.getStorageSync('provinceId') != provinceId) {
this.setData({
provinceId: wx.getStorageSync('provinceId')
})
// 仅地区切换时重置地区筛选条件
this.resetAreaFilter()
this.getLeftClassify()
} else {
this.setData({
provinceId: provinceId
})
}
} else {
this.setData({
provinceId: provinceId
})
}
if (province!=wx.getStorageSync('province')) {
this.setData({
province: wx.getStorageSync('province')
})
} else {
this.setData({
province: province
})
}
wx.removeStorageSync('areaFilter')
},
// 获取左边分类栏数据
getLeftClassify() {
let classFilter = this.data.classFilter
let newIndex = this.data.newIndex
let params = {
parentId: 0,
}
getNewClassify(params).then(res => {
if (res.code == 0 && res.data.length > 0) {
classFilter = res.data[newIndex].child.length > 0 ? [{
defaultName: "分类",
attributeName: "分类",
isAttr: false,
selectChildId: "", // 选择的 option 项 Id
child: res.data[newIndex].child
}] : []
let cId = res.data[newIndex].id
this.setData({
serviceList: res.data,
classFilter: classFilter,
cProId: cId
})
```

```
// 获取产品列表数据
this.getProList()
// 获取产品筛选属性数据
this.getAttributeData([cId])
}
})
},
// 获取筛选属性数据
getAttributeData(categoryIds) {
let classFilter = this.data.classFilter
if(categoryIds==1){
let attribute = [{
id: 1,
attributeName: "地区",
categoryId: 1
},{
id: 2,
attributeName: "级别",
categoryId: 1
},{
id: 3,
attributeName: "归口",
categoryId: 1
},]
attribute.forEach(item => {
item.selectChildId = ""
item.isAttr = true
item.defaultName = item.attributeName
})
this.setData({
classFilter: classFilter.concat(attribute)
})
}
},
// 获取项目申报筛选属性数据
getProAttrData(status, id) {
getProjectAttribute().then(res => {
if (res.code == 0) {
let provinceId = wx.getStorageSync('provinceId')
let data = wx.getStorageSync('district')
let area = []
let levelList = this.data.levelList
if (status) {
this.setData({
areaList: data
})
// 获取重构产品列表数据总入口
this.getLeftClassify()
} else {
// 获取地区数据
data.forEach(item => {
    if(item.children.length>0){
    item.children.map(item => {
    if(item.no == provinceId){
    area = item
    }
    })
    }
    })
    // 获取其他筛选项 option 数据
    let selection = res.data.selection.filter(item => {
    return item.categoryId == 2
    })
    if (selection.length > 0) {
    selection[0].options.forEach(item => {
    item.categoryName = item.name
    })
    }
    if (area.children.length > 0) {
    area.children.forEach(item => {
    item.categoryName = item.name
    item.id = item.no
    })
    this.setData({
    optionList: id == 1 ? area.children : (id == 2 ? levelList :
selection[0].options)
    })
    } else {
    this.setData({
    optionList: id == 1 ? [] : (id == 2 ? levelList : selection[0].options)
    })
    }
    }
    }
    })
    },
    // 筛选地区
    filterArea(val) {
    let areaList = this.data.areaList
    let filterList = []
    let r = areaList.filter(item => {
    return item.value == val
    })
    if (r.length > 0) {
    r[0].children.forEach(item => {
    filterList.push(item.value)
    })
    }
    return filterList
    },
    // 获取产品列表数据
    getProList() {
```

```
    let id = this.data.cProId
    let classFilter = this.data.classFilter
    var attribute = []
    classFilter.forEach(item => {
    if ((item.id && (item.id == 1)) || (item.id && (item.id == 3) &&
item.selectChildId != '')) {
    attribute.push({
    attributeId: item.id,
    attributeValueList: item.selectChildId == '' ?
[wx.getStorageSync(’provinceId’)] : [item.selectChildId]
    })
    }
    if(item.id == 2){
    attribute.push({
    attributeId: item.id,
    attributeValueList: [item.selectChildId]
    })
    }
    })
    let arr = []
    if (this.data.newIndex == 0) {
    arr = attribute.length > 0 ? attribute : [{
    attributeId: 1,
    attributeValueList: [this.data.provinceId]
    }]
    }
    let params = {
    pageNo: this.data.page,
    pageSize: this.data.pageSize,
    categoryIdList: id == 2 ? [3,4,5,6] : [id],
    sort: 'lattest',
    spuArea: '',
    spuKinds: '',
    spuLevel: ''
    }
    arr.forEach(item=>{
    if(item.attributeId == 1){
    params.spuArea = item.attributeValueList.toString()
    }else if(item.attributeId == 2){
    params.spuLevel = item.attributeValueList.toString() == '' ? '全部’ :
Number(item.attributeValueList.toString())
    }else{
    params.spuKinds = item.attributeValueList.toString()
    }
    })
    if(params.spuLevel==''||params.spuLevel==null){
    params.spuLevel = '全部’
    }
    getNewProduct(params).then(res => {
    if (res.code == 0) {
    let reachStatus = this.data.reachStatus
    let newProductList = this.data.newProductList
    let result = res.data.records
    result.forEach(item => {
    item.urlInfo = checkUrl(item.imageUrl)
    })
    this.setData({
    newProductList: reachStatus == true ? newProductList.concat(result) : result,
    reachStatus: false,
    newTotal: res.data.total,
    })
    }
    })
    },
    // 重置地区筛选数据
    resetAreaFilter() {
    let classFilter = this.data.classFilter
    classFilter.forEach(item => {
    if (item.id == 1) {
    item.selectChildId = ""
    item.attributeName = "地区"
    }
    })
    this.setData({
    classFilter: classFilter
    })
    },
    // 获取子组件传值
    getCityName(e){
    if(e.detail != ''){
    this.setData({
    province: e.detail,
    areaDialog: false
    })
    this.getProList()
    this.resetAreaFilter()
    }else{
    this.setData({
    areaDialog: false
    })
    }
    },
    // 获取地区
    getData(){
    let data = this.data.region
    let cityId = wx.getStorageSync('provinceId')
    let index1 = 0
    let index2 = 0
    let parentno = ''
    data.forEach((item) => {
    item.children.map(function(citem, cindex){
    if(citem.no == cityId){
    parentno = citem.parentNo
```

```
index2 = cindex
}
})
})
data.map(function(item,index){
if(item.no ==  parentno){
index1 = index
}
})
this.setData({
areaValue: [index1, index2]
})
},
/**
* 页面上拉触底事件的处理函数
*/
onReachBottom: function () {
let newProductList = this.data.newProductList
let newTotal = this.data.newTotal
if (newProductList.length < newTotal) {
this.setData({
reachStatus: true,
page: this.data.page + 1
})
this.getProList()
}
},
//监听页面滑动
onPageScroll(e) {
var that = this;
if (e.scrollTop > 0) {
that.setData({
showTop: true
})
} else {
that.setData({
showTop: false
})
}
},
// 校验登录
checkLoginStatus() {
if (!wx.getStorageSync('userInfo')) {
wx.showToast({
title: '您还未登录,请先登录吧~',
icon: 'none',
duration: 1000
})
setTimeout(function () {
wx.navigateTo({
url: "/personal/pages/wxLogin/wxLogin"
```

```
})
}, 1000)
} else {
this.data.areaList.length > 0 ? '' : this.getProAttrData(true)
if (wx.getStorageSync('areaFilter')) {
this.setData({
page: 1,
newProductList: []
})
// 获取当前定位地区数据
this.getSelectArea()
} else {
this.getSelectArea()
}
}
},
// 修改 tabbar index
changeTabbar() {
let islogin = wx.getStorageSync('userInfo') ? true : false
if (typeof this.getTabBar === 'function' && this.getTabBar()) {
this.getTabBar().setData({
selected: 1,
isLogin: islogin
})
}
},
onLoad(options) {
if (options.type) {
this.setData({
newIndex: options.type,
})
}
this.setData({
district: wx.getStorageSync('provinceId'),
province: wx.getStorageSync('province'),
provinceId: wx.getStorageSync('provinceId'),
})
if(wx.getStorageSync('district')){
this.setData({
region: wx.getStorageSync('district')
})
}else{
getDistrict().then(res=>{
if(res.data.code == 0){
let data = res.data.data
wx.setStorageSync('district', data)
}
})
this.setData({
region: wx.getStorageSync('district')
})
```

```
    }
    },
    onShow: function () {
    this.checkLoginStatus()
    this.changeTabbar()
    this.getData()
    })
    <!-- 产品详情页 -->
```

wxml代码：

```
    <!--search/pages/productdetails/productdetails.wxml-->
    <import src="../wxParse/wxParse.wxml" />
    <wxs src="/utils/productdetails/productdetails.wxs" module="tools"></wxs>
    <view class="global-tip" wx:if="{{showDialog==true}}" bindtap="handleDialog"></view>
    <view>
    <!-- 图片区域 -->
    <view class="product-img">
    <swiper class="bg-img" indicator-dots="{{coverUrls.length>1}}" indicator-color="#EAEBEF" indicator-active-color="#61BAFF">
    <swiper-item wx:for="{{coverUrls}}" wx:key="idx">
    <video custom-cache="{{false}}" controls="{{videoControl}}" wx:if="{{item.urlType=='mp4'||item.urlType=='avi'||item.urlType=='mov'}}" class="bg-img" src="{{imageUrl+item.imageUrl}}"></video>
    <image wx:else class="bg-img" src="{{imageUrl+item.imageUrl}}" />
    </swiper-item>
    </swiper>
    </view>
    <!-- 服务费 -->
    <view class="service-price">
    <block wx:if="{{noSetPrice}}">
    <view class="service-item item-name">该组合产品尚未设置价格</view>
    </block>
    <block wx:if="{{!baseInfo.isOnline && !noSetPrice}}">
    <view class="service-item justify-left">
    <view class="item-name">服务费</view>
    <view class="item-price">面议</view>
    </view>
    </block>
    <block wx:if="{{baseInfo.isOnline && !noSetPrice}}">
    <!-- 一口价 -->
    <block wx:if="{{choiceAttribute.dueTypeOfficialFee || choiceAttribute.dueTypeAgencyFee || choiceAttribute.dueTypeThirdPartyFee}}">
    <view class="service-item justify-left">
    <view class="item-name">服务费</view>
    <view class="item-price"><text class="markyuan">￥</text>{{tools.filterViewprice(tools.filterOnePrice(choiceAttribute))}}</view>
    </view>
    </block>
    <block wx:elif="{{choiceAttribute.noAttritube==true}}">
    <view class="service-item justify-left">
    <view class="item-name">服务费</view>
    <view class="item-price">待定</view>
    </view>
    </block>
    <!-- 定金+尾款 -->
    <block wx:else>
    <view class="service-item">
    <view class="flex">
    <view class="item-name">定金</view>
    <view class="item-price"><text class="markyuan">
￥</text>{{tools.filterViewprice(tools.filterdepositPrice(choiceAttribute))}}</
view>
    </view>
    <view class="flex">
    <view class="item-name">尾款</view>
    <view class="item-price"><text class="markyuan">
￥</text>{{tools.filterViewprice(tools.filterfinalPrice(choiceAttribute))}}</vi
ew>
    </view>
    </view>
    </block>
    </block>
    <view class="area" wx:if="{{baseInfo.categoryId==1}}">
    <view>
    <view class="mb">地区</view>
    <view class="attribute-txt">{{areas}}</view>
    </view>
    <view>
    <view class="mb">归口</view>
    <view class="attribute-txt">{{belong}}</view>
    </view>
    </view>
    </view>
    <!-- 标题 -->
    <view class="details">
    <view class="productservice">
    <view class="service-box">
    <view class="product-title line-2-ellipsis">{{baseInfo.spuTitle}}</view>
    <view class="product-sologan line-1-ellipsis">{{baseInfo.spuSlogan}}</view>
    <view class="advantage">
    <view wx:for="{{iconList}}" wx:key="index">
    <image class="advantage-img" src="{{item.url}}" />
    <text class="advantage-text">{{item.name}}</text>
    </view>
    </view>
    </view>
    </view>
    <view class="product-fee">
    <view class="fee-detail" wx:if="{{choiceAttribute.depositTypeOfficialFee ||
choiceAttribute.dueTypeOfficialFee}}" bindtap="handleDialog">
    <view>
    <text class="guanfee">官费</text>
    <text class="guanprice"
wx:if="{{choiceAttribute.depositTypeOfficialFee}}">{{tools.filterViewprice(choi
ceAttribute.depositTypeOfficialFee)}}</text>
```

```
    <text class="guanprice"
wx:if="{{choiceAttribute.dueTypeOfficialFee}}">{{tools.filterViewprice(choiceAt
tribute.dueTypeOfficialFee)}}</text>
    </view>
    <image class="more" src="../../image/productdetails/gengduo_icon@2x.png" />
    </view>
    <view class="fee-detail" bindtap="handleDialog">
    <view>
    <text class="guanfee">规格</text>
    <text class="guanprice dark">{{attributeText}}</text>
    </view>
    <image class="more" src="../../image/productdetails/gengduo_icon@2x.png" />
    </view>
    </view>
    </view>
    <!-- 固定导航头 -->
    <view class="nav-bar" wx:if="{{showTabStatus}}">
    <view class="nav-bar-item" wx:for="{{articleTitleArray}}" wx:key="index"
bindtap="changeNav" data-index="{{index}}">
    <text
class="{{navBarIndex==index?'active-nav-sty':''}}">{{item.title}}</text>
    <text wx:if="{{navBarIndex==index}}"></text>
    </view>
    </view>
    <!-- 内容区域 -->
    <view class="product-content" id="productBox">
    <view class="service-pro">
    <view class="service-pro-title">服务流程</view>
    <image class="pro-img"
src="https://static.wotao.com/icon/head/1d4f42cc2ba943acacee636066678247.png"><
/image>
    </view>
    <view class="pc-item" id="{{'pc-item'+index}}" wx:for="{{articleArray}}"
wx:key="index">
    <!-- 标签标题区域 -->
    <view class="pc-item-title">
    <text></text>
    <text>{{articleTitleArray[index].title}}</text>
    </view>
    <!-- 标签内容区域 -->
    <view class="pc-item-content">
    <view wx:if="{{item.tag == 'video'}}">
    <template is="wxParseVideo" data="{{wxParseData:item}}" />
    </view>
    <view wx:else>
    <template is="wxParse" data="{{wxParseData:item}}" />
    </view>
    </view>
    </view>
    </view>
    <!-- 产品列表 -->
    <View class="hot-contain" wx:if="{{hotList.length>0}}">
    <view class="hot-title">
    <text></text>
    <text>热门商品</text>
    </view>
    <view class="new-product-list" style="min-height:480px">
    <view class="new-product-item" wx:for="{{hotList}}" wx:key="index">
    <!-- 其他五类新产品 item -->
    <otherItem item="{{item}}" index="{{index}}"
categoryId="{{baseInfo.categoryId}}" hot="{{true}}"></otherItem>
    </view>
    </view>
    </View>
    <!-- 底部按钮区域 -->
    <view class="foot">
    <footBar></footBar>
    <view class="foot-btn">
    <view bindtap="handleCollect" class="left-btn
{{collectStatus==true?'collected':'not-collected'}}">
    <text wx:if="{{collectStatus==true}}">已收藏</text>
    <text wx:if="{{collectStatus== false}}">加入收藏</text>
    </view>
    <block wx:if="{{baseInfo.isOnline == true}}">
    <button class="loginBtn {{noSetPrice?'disable-btn':''}}"
bindtap="showPriceDialog">立即下单</button>
    </block>
    <block wx:else>
    <button open-type="contact" class="loginBtn">立即咨询</button>
    </block>
    </view>
    </view>
    <!-- 弹框 -->
    <view class="dialog" wx:if="{{showDialog==true}}">
    <view class="attribute">
    <view wx:for="{{dialogImgList}}" style="display:flex" wx:key="index">
    <video class="popImg" custom-cache="{{false}}" controls="{{videoControl}}"
wx:if="{{item.urlType=='mp4'||item.urlType=='avi'||item.urlType=='mov'}}"
src="{{imageUrl + item.imageUrl}}" controls="controls"></video>
    <image class="popImg" wx:else src="{{imageUrl + item.imageUrl}}" />
    <view class="popfee" wx:if="{{!noSetPrice}}">
    <view class="first">
    <block wx:if="{{baseInfo.isOnline == false}}">
    <text class="a">服务费</text>
    <text class="b" style="font-size: 26rpx;">面议</text>
    </block>
    <block wx:if="{{baseInfo.isOnline == true}}">
    <block wx:if="{{choiceAttribute.dueTypeOfficialFee ||
choiceAttribute.dueTypeAgencyFee || choiceAttribute.dueTypeThirdPartyFee}}">
    <text class="a">服务费</text><text class="b"><text style="font-size:22rpx">
￥</text>{{tools.filterViewprice(tools.filterOnePrice(choiceAttribute))}}</text
>
    </block>
    <block wx:elif="{{choiceAttribute.noAttritube==true}}">
    <text class="a">服务费</text>
    </block>
    <block wx:else>
    <view>
    <text class="a">定金</text><text class="b"><text style="font-size:22rpx">
￥</text>{{tools.filterViewprice(tools.filterdepositPrice(choiceAttribute))}}</
text>
    </view>
    <view>
    <text class="a">尾款</text><text class="b"><text style="font-size:22rpx">
￥</text>{{tools.filterViewprice(tools.filterfinalPrice(choiceAttribute))}}</te
xt>
    </view>
    </block>
    </block>
    </view>
    <view class="second" wx:if="{{ choiceAttribute.depositTypeOfficialFee ||
choiceAttribute.dueTypeOfficialFee}}">
    <text class="a">官<text decode="{{true}}">&nbsp;&nbsp;&nbsp;</text>费
</text><text class="b"><text style="font-size:22rpx">￥</text>
    <block
wx:if="{{choiceAttribute.depositTypeOfficialFee}}">{{tools.filterViewprice(choi
ceAttribute.depositTypeOfficialFee)}}</block>
    <block
wx:if="{{choiceAttribute.dueTypeOfficialFee}}">{{tools.filterViewprice(choiceAt
tribute.dueTypeOfficialFee)}}</block>
    </text>
    </view>
    </view>
    <view wx:else><text class="a">该组合产品尚未设置价格</text></view>
    </view>
    <block wx:if="{{transactionList.length>0}}">
    <view wx:for="{{attributeList}}" wx:for-index="parentIndex"
wx:for-item="parentItem">
    <view class="attrleft">{{parentItem.key}}</view>
    <view class="attrright">
    <view wx:for="{{parentItem.value}}" wx:for-index="childIndex"
wx:for-item="childItem" wx:key="childIndex">
    <view wx:if="{{tools.filterBelong(childItem,noPriceIndex)}}"
data-parentIndex="{{parentIndex}}" data-childIndex="{{childIndex}}"
bindtap="handleClick" class="attributeItem {{parentIndex==0 &&
tools.filterSlice(activeKey,0,1)==childIndex?'activeTip':''}} {{parentIndex==1
&&
tools.filterSlice(activeKey,2,3)==childIndex?'activeTip':''}}">{{childItem}}</v
iew>
    </view>
    </view>
    </view>
    <view class="flex">
    <view class="att mt">数量选择</view>
    <view>
    <text class="min" data-index="0" bindtap="checkNum">-</text>
    <text class="buy">{{productNum}}</text>
    <text class="min" data-index="1" bindtap="checkNum">+</text>
    </view>
    </view>
    </block>
```

```
</view>
</view>
</view>
```

js代码：

```
const app = getApp();
var WxParse = require('../wxParse/wxParse.js');
import {
addPageViews,
pageWeightSpu,
pageInfo,
getNewProduct
} from "../../../utils/api/product.js"
import {
allFactor
} from "../../../utils/api"
import {
filterArea
} from "../../../utils/common/cacheDatas"
import {
getProductInfo,
isCollected,
addcollect,
cancelcollect
} from "../../../utils/api/productdetails.js"
Page({
/**
* 页面的初始数据
*/
data: {
hotList: [],
videoControl: false,
spuId: '',
belong: null,
goodsIdgoodsId: '',
proType: '',
navBarIndex: 0, // 导航头下标
showTabStatus: false, // 控制导航头显示
imageUrl: app.globalData.imageURL,
articleTitleArray: [],
articleArray: [], //富文本
coverUrls: [], //轮播图
iconList: [{
name: "全程监督",
url: "/search/image/productdetails/quanchengjiandu_icon@2x.png"
}, {
name: "保障权益",
url: "/search/image/productdetails/baozhangquanyi_icon@2x.png"
}, {
name: "十年行业经验",
url: "/search/image/productdetails/shinianjingyan_icon@2x.png"
}],
baseInfo: {},
attributeList: [], //属性规格
transactionList: [], //后台配置可交易选项
activeKey: null, // 默认选中规格
flag: null, //是否为单层属性
productNum: 1, //数量
collectStatus: false,
dialogImgList: [], //弹框里面内容
choiceAttribute: {}, // 选中的规格（动态）
attributeText: null, // 选中的文案（动态）
showDialog: false,
area: '',
noPriceIndex: [],
skuIndexArr: [],
noSetPrice: false,
},
// 导航头选择并滑动到指定区域
changeNav(e) {
let index = e.currentTarget.dataset.index
this.setData({
navBarIndex: index
})
var query = wx.createSelectorQuery() // 创建节点查询器
query.selectViewport().scrollOffset(); // 节点的竖直滚动位置
query.select(`#pc-item${index}`).boundingClientRect() // 动态获取 id
query.exec(function (res) { // 执行请求
wx.pageScrollTo({
scrollTop: res[0].scrollTop + res[1].top - 46, // 滚动到页面节点的上边界坐标
duration: 0 // 滚动动画的时长
});
})
},
// 获取产品详情
getDetail(spuId) {
console.log(this.data.spuId, 'getDetail');
wx.showLoading({
title: '加载中',
})
getProductInfo({spuId}).then(res => {
if (res.code == 0) {
wx.hideLoading({
success: (res) => {}
})
let data = res.data
//轮播图
if (data.baseInfo.imageList && data.baseInfo.imageList.length) {
let urlList = []
data.baseInfo.imageList.forEach((item, index) => {
urlList[index] = Object.assign(item, {
urlType: item.imageUrl.substr(item.imageUrl.lastIndexOf('.') + 1)
})
```

```
    })
    this.setData({
    coverUrls: urlList
    })
    }
    if (data.baseInfo.categoryId == 1) {
    let areas = [];
    let aItem = data.baseInfo.generateAttributeList.find(item =>
item.attributeName == ’地区’);
    let belongs = data.baseInfo.generateAttributeList.find(item =>
item.attributeId == 2);
    areas[0] = Number(aItem.attributeValueDesc)
    areas[1] = Number(aItem.attributeValue)
    this.setData({
    areas: aItem.attributeValueDesc,//filterArea(areas),
    belong: belongs.attributeValueDesc
    })
    }
    //富文本
    let articleArray = [];
    data.detailInfoList.forEach(item => {
    articleArray.push({
    title: item.tagName,
    content: item.text
    })
    })
    this.setData({
    articleTitleArray: articleArray
    })
    let that = this
    for (let i = 0; i < articleArray.length; i++) {
    WxParse.wxParse(’article’ + i, ’html’, articleArray[i].content, that);
    if (i === articleArray.length - 1) {
    WxParse.wxParseTemArray(”articleArray”, ’article’, articleArray.length,
that);
    }
    }
    console.log(this.data.articleArray,’articleArray’)
    // 弹框内容
    let dialogImgList = data.baseInfo.imageList.filter(item => item.isDefault ==
true)
    let imgList = []
    dialogImgList.forEach((item, index) => {
    imgList[index] = Object.assign(item, {
    urlType: item.imageUrl.substr(item.imageUrl.lastIndexOf(’.’) + 1)
    })
    })
    //是否为单层属性
    let flag = data.tradeInfo.skuList.some(item => item.skuIndex.indexOf(’_’) >
-1)
    this.setData({
    baseInfo: data.baseInfo,
    dialogImgList: imgList,
    transactionList: data.tradeInfo.skuList,
    activeKey: data.tradeInfo.skuList[0].skuIndex,
    flag
    });
    let skuIndexs = this.data.transactionList.map(x => {
    return x.skuIndex
    });
    this.setData({
    skuIndexArr: skuIndexs
    });
    //去除没有搭配的属性
    let filterattributeList = data.tradeInfo.dynamicAttributeList.map(item => ({
    key: Object.keys(item)[0],
    value: Object.values(item)[0]
    }))
    let recordArr = []
    let doublerecordArr = []
    data.tradeInfo.skuList.forEach(element => {
    if (element.skuIndex.length == 1) {
    recordArr.push(Number(element.skuIndex))
    } else {
    recordArr.push(Number(element.skuIndex.substr(0, 1)))
    doublerecordArr.push(Number(element.skuIndex.substr(2)))
    }
    })
    // Array.from(filterattributeList)[0].value =
Array.from(filterattributeList)[0].value.filter((item,index)=>Array.from(new
Set(recordArr)).includes(index))
    let difference = Array.from(filterattributeList)[0].value.filter((item, index)
=> !new Set(recordArr).has(index));
    this.setData({
    noPriceIndex: difference
    })
    // if(flag){
    //     Array.from(filterattributeList)[1].value =
Array.from(filterattributeList)[1].value.filter((item,index)=>Array.from(new
Set(doublerecordArr)).includes(index))
    // }
    this.setData({
    attributeList: filterattributeList
    })
    this.handlechoiceAttribute()
    this.handleattributeText()
    if(data.baseInfo.categoryId == 1){
    let para = this.fixFiler()
    this.getHotList(data.baseInfo.categoryId,para,spuId)
    } else {
    this.getHotList(data.baseInfo.categoryId,[],spuId)
    }
    }
    })
```

```
},
// 传参修改
fixFiler(){
let list = []
this.data.baseInfo.generateAttributeList.forEach(item=>{
list.push({
attributeId: item.attributeId,
attributeValueList: [item.attributeValue]
})
})
return list
},
// 总价计算
totalFix() {
let optionList = this.data.projectInfo.policyGovfeeTypeList
let count = this.data.productCount // 选择的产品数量
let earnestMoney = this.data.projectInfo.earnestMoney // earnestMoney 定金
if (optionList.length == 0) {
return count * (Number(earnestMoney))
} else {
let index = this.data.agencyFeeType // 官费代缴选择下标
let governmentFee = optionList[index].typeFee // governmentFee 此处仅表示官费代缴费用
return count * (Number(earnestMoney) + Number(governmentFee))
}
},
showPriceDialog() {
if (!this.data.showDialog) return this.setData({
showDialog: true
})
this.toSubmit()
},
toSubmit() {
if (this.data.noSetPrice) return;
let choiceAttribute = this.data.choiceAttribute;
let params = {
pageFrom: 1, // 1 代表从产品页面进
contractScope: this.data.baseInfo.contractScope,
id: this.data.spuId, // 产品 id
attributeText: this.data.attributeText,
goodsId: this.data.goodsId, //商品 id
categoryId: this.data.baseInfo.categoryId, //产品类型
type: this.data.proType,
number: this.data.baseInfo.spuCode, // 产品 number
choiceAttribute: choiceAttribute,
coverUrlInfo: this.data.dialogImgList[0] ?
this.data.dialogImgList[0].imageUrl : null, // 封面对象
productName: this.data.baseInfo.spuTitle, // 产品名称
productCount: this.data.productNum, // 选择的产品数量
tailstatusPayFlag: false // 是否支付尾款
}
console.log(params, '---------');
```

```
let data = JSON.stringify(params)
wx.navigateTo({
url: `/order/pages/submitOrder/projectOrder/projectOrder?data=${data}`,
})
},
//切换规格
handleClick(event) {
let parentIndex = event.currentTarget.dataset.parentindex
let childIndex = event.currentTarget.dataset.childindex
if (parentIndex == 0) {
if (this.data.flag) {
let activeKey = childIndex + '_' + this.data.activeKey.substr(2)
this.setData({
activeKey
})
} else {
this.setData({
activeKey: String(childIndex)
})
}
} else {
let activeKey = this.data.activeKey.substr(0, 1) + '_' + childIndex
this.setData({
activeKey
})
}
this.handlechoiceAttribute()
this.handleattributeText()
},
//数量选择
checkNum(event) {
let type = event.currentTarget.dataset.index
if (type == '1') {
if (this.data.productNum == 99) {
return
}
this.setData({
productNum: ++this.data.productNum
})
} else {
if (this.data.productNum == 1) {
return
}
this.setData({
productNum: --this.data.productNum
})
}
},
handlechoiceAttribute() {
if (!this.data.skuIndexArr.includes(this.data.activeKey)) {
this.setData({
noSetPrice: true,
})
return app.toastBox('该类型尚未设置价格', 'error', 1000)
}
if (this.data.activeKey && this.data.transactionList.length > 0) {
if (this.data.transactionList.filter(item => item.skuIndex == this.data.activeKey).length) {
let choiceAttribute = this.data.transactionList.filter(item => item.skuIndex == this.data.activeKey)[0].skuPayTypeList
let goodsId = this.data.transactionList.filter(item => item.skuIndex == this.data.activeKey)[0].id
this.setData({
choiceAttribute: choiceAttribute,
goodsId: goodsId
})
} else {
this.setData({
choiceAttribute: {
noAttritube: true
}
})
}
} else {
this.setData({
choiceAttribute: {}
})
}
},
handleattributeText() {
if (this.data.activeKey) {
if (this.data.activeKey.length == 1) {
let attributeText = this.data.attributeList[0].value[Number(this.data.activeKey)]
this.setData({
attributeText
})
} else {
let attributeText = this.data.attributeList[0].value[Number(this.data.activeKey.substr(0, 1))] + ',' + this.data.attributeList[1].value[Number(this.data.activeKey.substr(2, 3))]
this.setData({
attributeText
}))
<!-- 提交订单，确认订单页面 -->
wxml代码：
<!--order/pages/submitOrder/projectOrder/projectOrder.wxml-->
<wxs src="/utils/order/order.wxs" module="tools"></wxs>
<!-- 公共导航栏组件 -->
<navBar type="{{1}}" bind:close-dialog="closeDialog" cType="{{orderInfo.pageFrom}}" navTitle="{{orderInfo.pageFrom == 0?'确认支付':'确认订单'}}"></navBar>
<view class="submit-order" style="margin-top: {{height}}rpx;">
<!-- 项目申报产品信息 -->
<projectInfo bind:setCount="setCount" orderInfo="{{orderInfo}}" serviceFee="{{serviceFee}}" totalPriceList="{{totalPriceList}}"></projectInfo>
<!-- 收件地址 -->
<view class="address" wx:if="{{orderInfo.pageFrom == 0}}">
<text>收件信息</text>
<addressItem addressInfo="{{defaultAddress}}"></addressItem>
</view>
<!-- 支付方式 -->
<view class="pay-way {{totalPriceList[2].value>6000?'active-pay-way':''}}" wx:if="{{orderInfo.pageFrom == 0}}">
<text>支付方式</text>
<view wx:for="{{payList}}" wx:key="index">
<view class="pay-desc" wx:if="{{index<2||(index==2&&totalPriceList[2].value<=6000)}}">
<view class="pay-img">
<image src="{{item.url}}"></image>
<text>{{item.title}}</text>
</view>
<view class="check-sty" bindtap="changePay" data-index="{{item.index}}">
<image wx:if="{{checkedIndex == item.index}}" src="/order/image/orders/checkd.png"></image>
<image wx:else src="/order/image/orders/uncheck.png"></image>
</view>
</view>
</view>
</view>
<!-- 提交订单按钮 -->
<view class="submit-btn">
<view class="submit-left">
<!-- 总价显示区域 -->
<view class="total-fix">
<text>共{{orderInfo.productCount}}件，</text>
<view class="total-price">
<text>小计：</text>
<text>￥</text>
<text wx:if="{{orderInfo.pageFrom!=0||((orderInfo.pageFrom==0)&&!orderInfo.tailstatusPayFlag)}}">{{orderInfo.totalPrice}}</text>
<text wx:if="{{(orderInfo.pageFrom==0)&&orderInfo.tailstatusPayFlag}}">{{orderInfo.feeInfo.tailShould}}</text>
</view>
</view>
<view>
<!-- 交易协议 -->
<view class="pay-agreement" wx:if="{{orderInfo.pageFrom == 0}}">
<!-- wx:if="{{!orderInfo.tailstatusPayFlag}}" -->
<image bindtap="changeAgreement" data-status="{{false}}" wx:if="{{agreementStatus}}" src="/order/image/orders/checkd.png"></image>
<image bindtap="changeAgreement" data-status="{{true}}" wx:else src="/order/image/orders/uncheck.png"></image>
<text>请阅读并同意</text>
<text bindtap="toAgreement">《卧涛网交易服务协议》</text>
```

```
</view>
</view>
</view>
<view class="confirm-btn">
<text wx:if="{{orderInfo.pageFrom == 1 || orderInfo.pageFrom == 2}}" bindtap="checkPower" class="long-txt-sign">立即签约下单</text>
<!-- <text wx:if="{{orderInfo.pageFrom == 1 || orderInfo.pageFrom == 2}}" bindtap="toPay">提交订单</text> -->
<text wx:else bindtap="toOrderPay">立即支付</text>
</view>
</view>
<view class="popup" hidden="{{mask}}">
<view class="mask" catchtouchmove="preventTouchMove" catchtap="closeMask"></view>
<view class="popup_main">
<view class="pop-title">
<text>选择签约方式</text>
<image catchtap="closeMask" class="close-ico" src="{{imageUrl+'icon/head/b0b594b92ed64062b013ab47da74b923.png'}}"></image>
</view>
<view class="user-item" wx:for="{{signList}}" wx:key="index" catchtap="chooseOne" data-item="{{item.value}}">
<image wx:if="{{signIndex == item.value}}" src="{{proImageUrl+'icon/head/b968ca9ea15a4b65a7a3933d2e247494.png'}}"></image>
<image wx:else src="{{proImageUrl+'icon/head/f87e7fe190d843ca9b8c25ade5d95abe.png'}}"></image>
<text>{{item.title}}</text>
<text class="sign-desc">{{item.desc}}</text>
</view>
<view class="pop-btn" catchtap="changeContract">确 定</view>
</view>
</view>
</view>
```

js代码：

```
// order/pages/submitOrder/projectOrder/projectOrder.js
const app = getApp();
var common = require('../../../../utils/common/userDeal')
import { getOpenId, layout } from "../../../../utils/api"
import { getEssAuthStatus } from "../../../../utils/api/user"
import { payForApplet } from "../../../../utils/api/orders"
import { submitOrder, submitToPay } from "../../../../utils/api/order"
import { getAddressList } from "../../../../utils/api/address"
import { updateAccountInfo } from "../../../../utils/api/user"
import { throttle } from "../../../../utils/common/utils"
Page({
/**
* 页面的初始数据
*/
data: {
// 微信支付参数
signList:[{
value:1,
title:'电子合同',
desc: '线上签约，安全高效省心'
},{
value:2,
title:'纸质合同',
desc: '面签，扫描，邮寄等流程'
}],
signIndex:2,
wxInfo: {},
orderItemInfo: {},
baseURL: app.globalData.baseURL,
priceType: '',
// 尾款总价
tailAmount: 0,
thirdFee: 0,
orderListNo: '',
// 显示价格总价
totalPriceList: [{
title: "定金总计",
value: ""
},
{
title: "官费总计",
value: ""
},
{
title: "本次应支付金额",
value: ""
},
],
mask:true,
wxPayMax: false,
finalIndex: null, //防止数据结构有问题，尾款未设置情况
// status：0-待付款 1-未开始 2-已付款
// payProcess：控制样式显示 true 红色 false 灰色
serviceFee: [{
title: "定金",
value: "",
status: 0,
payProcess: true
}, {
title: "尾款",
value: "",
status: 1,
payProcess: false
}, {
title: "官费",
value: "",
status: 0,
payProcess: true
}],
dialogStatus: true,
orderNo: "",
// 页面内容距最顶部的距离
height: app.globalData.navHeight * 2 + 10,
// 跳转到对公转账支付页要带的信息
transferInfo: {},
defaultAddress: {},
userInfo: {},
orderInfo: {},
imageUrl: app.globalData.imageURL,
proImageUrl:'https://static.wotao.com/',
agreementStatus: true,
checkedIndex: 1,
payList: [{
url: "/order/image/orders/transfar_pay.png",
title: "线下对公转账支付",
index: 1,
},
{
url: "/order/image/orders/bao_pay.png",
title: "支付宝支付",
index: 0,
},
{
url: "/order/image/orders/weixin_icon.png",
title: "微信支付",
index: 2,
},
],
saveOrderInfo:null,
},
// 检查授权情况
checkPower(){
let that = this;
let data = this.data.orderInfo;
let contractType = data.contractScope == 3 ? 2 : data.contractScope; //1:电子，2:纸质，contractScope：3合同可选择
let certifiedType = wx.getStorageSync('userInfo').certifiedType; //0：未认证；1：初级认证；2：高级认证
let hignAuth = certifiedType == 1 ? '高级' : '';
if(certifiedType == 0 || contractType == 1 && certifiedType == 1 || !certifiedType) return wx.showModal({
title: '操作提示',
content:'由于您的身份未'+hignAuth+'认证，暂时无法进行签约操作，为了给您提供更好的服务，请尽快完成认证!',
showCancel: true,
confirmText:'去认证',
cancelText:'放弃认证',
success (res) {
if (res.confirm) {
let userInfo = wx.getStorageSync('userInfo');
setTimeout(function () {
wx.navigateTo({
    url: '/personal/pages/authentication/authentication?accountType=' +
userInfo.accountType,
    })
    }, 1000)
    }
    }
    });
    console.log(contractType,certifiedType);
    if(contractType == 2 && (certifiedType == 1 || certifiedType == 2)) return
this.toSignOrder();
    getEssAuthStatus().then(res=>{
    if(res.code == 0){
    if(res.data.status=='2'|| res.data.status=='5' || res.data.status=='6')
return this.toSignOrder();
    // if(res.data.status == '0') return
app.toGoalUrl(`/personal/pages/authentication/authentication`) 0 未激活
    let content = res.data.status=='3'?'签约授权中，点击查看授权结果' : '由于您
首次使用签约功能，需要完成签约授权，请先去授权再来签约吧~'
    if(res.data.status!='2') return
app.modalBox('',content,'',function(response){
    if(response){
    that.toImpower(res.data.status,res.data.authUrl)
    }
    })
    }
    });
    },
    // 去授权查看授权结果
    toImpower(status,url){
    let para = {
    from: 'old-sign',
    authorizer: 'sq'
    }
    let data = JSON.stringify(para)
    let authUrl =
wx.getStorageSync('userInfo').userType=='2'?`/personal/pages/orgAuthentication/
orgAuthentication?data=${data}`:
`/personal/pages/perAuthentication/operatePage/operatePage?data=${data}`
    console.log(authUrl);
    if(status==3){
    wx.setStorageSync('reloadUrl', url);
    app.toGoalUrl(`/pagesFace/pages/webview/webview?root=user`)
    } else {
    app.toGoalUrl(authUrl)
    }
    },
    toSignOrder(){
    let data = this.data.orderInfo;
    if(data.contractScope == 3) return this.setData({
    mask: false,
    });
    this.submitOrderData();
    },
    changeContract() {
    let that = this;
    let certifiedType = wx.getStorageSync('userInfo').certifiedType; //0：未认证；
1：初级认证；2：高级认证
    // 选择电子签约身份只有初级认证
    if(this.data.signIndex == 1 && certifiedType == 1) return wx.showModal({
    title: '操作提示',
    content:'由于您的身份未高级认证，暂时无法进行电子签约操作，为了给您提供更好
的服务，请尽快完成高级认证，或者更换签约方式！',
    showCancel: true,
    confirmText:'去认证',
    cancelText:'放弃认证',
    success (res) {
    if (res.confirm) {
    that.closeMask();
    let userInfo = wx.getStorageSync('userInfo');
    setTimeout(function () {
    wx.navigateTo({
    url: '/personal/pages/authentication/authentication?accountType=' +
userInfo.accountType,
    })
    }, 300)
    }
    }
    });
    that.closeMask();
    this.submitOrderData();
    },
    closeMask(){
    this.setData({
    mask: true,
    })
    },
    //弹框选择签约方式
    chooseOne(e){
    let index = e.currentTarget.dataset.item;
    console.log(index);
    this.setData({
    signIndex: index
    })
    },
    // 提交订单
    submitOrderData() {
    let orderInfo = this.data.orderInfo
    let params = {
    buyNum: orderInfo.productCount,
    goodsSkuId: orderInfo.goodsId,
    // addressId: this.data.defaultAddress.id,
    orderPort: 'APPLET', //下单端口(PC，APP，H5,APPLET)
    orderSource: 1, //订单来源(1-自主下单；2-后台发起)
    totalAmount: orderInfo.pageFrom == 2 ? orderInfo.totalAmount :
this.finalFixTotal(),
    }
    if (wx.getStorageSync(’spuLinkId’) && wx.getStorageSync(’spuInfoArr’)) {
    let spuInfoArr = wx.getStorageSync(’spuInfoArr’)
    let result = spuInfoArr.filter(item => {
    return item.spuId == orderInfo.id
    })
    if (result.length > 0) {
    params.inviteCode = wx.getStorageSync(’spuLinkId’)
    }
    }
    // 从订单页面过来采用支付宝支付传订单编号 orderNo
    if (this.data.orderInfo.pageFrom == 0) {
    params.orderNo = orderInfo.orderNo
    params.id = orderInfo.orderId
    }
    console.log(’产品参数’, params)
    if (this.data.orderInfo.pageFrom == 1 || this.data.orderInfo.pageFrom == 2) {
    // 首次提交订单
    this.firstToPayOroder(params)
    } else {
    // 待付款去支付
    this.submitToPayOrder(params)
    }
    },
    // 首次提交订单
    firstToPayOroder(params) {
    let that = this;
    let page2 = {
    id: wx.getStorageSync(’orderId’).orderId,
    addressId: this.data.defaultAddress.id,
    orderPort: 'APPLET’,
    }
    let contractType = that.data.orderInfo.contractScope == 3 ?
that.data.signIndex : that.data.orderInfo.contractScope;
    let data = Object.assign({contractType: contractType}, params)
    let urls = that.data.orderInfo.pageFrom == 2 ? '/tmOrder/confirmOrder' :
'/tmOrder/saveOrder';
    wx.request({
    url: that.data.baseURL + urls,
    method: 'post’,
    data: that.data.orderInfo.pageFrom == 2 ? page2 : data,
    header: {
    'content-type’: 'application/json’,
    'Accept’: 'application/json’,
    'Authorization’: wx.getStorageSync(’token’)
    },
    success(res) {
    if (res.statusCode == 401) {
    wx.showToast({
    title: '您还未认证,请先认证吧~’,
    icon: 'none’,
    duration: 1000
    })
    let userInfo = wx.getStorageSync('userInfo');
    setTimeout(function () {
    wx.navigateTo({
    url: '/personal/pages/authentication/authentication?accountType=' +
userInfo.accountType,
    })
    }, 1000)
    } else {
    let resp = res.data;
    if (resp.code == 0) {
    console.log(that.data.signIndex,'00000');
    let datas = resp.data;
    let data = that.data.orderInfo;
    let contractType = data.contractScope == 3 ? that.data.signIndex :
data.contractScope; //1:电子，2:纸质，contractScope：3 合同可选择
    if(contractType == 1) return common.eleContract({
    id: datas.id,
    contractType: 1,
    orderNo: datas.orderNo
    })
    wx.showModal({
    title: '操作提示',
    content: '订单提交成功，请待业务经理上传纸质合同~',
    showCancel:false,
    confirmText:'我知道了',
    success (res) {
    if (res.confirm) {
    wx.navigateTo({
    url: '/personal/pages/orderIndex/order?type=1',
    })
    }
    }
    })
    } else {
    wx.showToast({
    title: resp.msg,
    icon: 'none',
    })
    }
    }
    },
    fail(err) {
    console.log(err);
    console.log(urls);
    wx.showModal({
    title: '提示',
    content: '网络请求失败',
    showCancel: false
    })
    }
    })
```

```
},
toOrderListSign(){
},
// 总价计算
totalFix(count) {
let goverment = this.data.totalPriceList[1].value // 官费价格
let deposit = this.data.totalPriceList[0].value // 定金
return count * (Number(deposit) + Number(goverment))
},
// 订单总价判断返回
finalFixTotal() {
let cOrderInfo = this.data.orderInfo;
let previous = this.totalFix(cOrderInfo.productCount);
let tail = cOrderInfo.productCount * (Number(this.data.tailAmount))
if (this.data.tailAmount == 0) {
return previous
} else {
return previous + tail
}
},
// 修改产品数量
setCount(val) {
let status = val.detail
let cOrderInfo = this.data.orderInfo
if (status) {
if (cOrderInfo.productCount < 99) {
cOrderInfo.productCount += 1
cOrderInfo.totalPrice = this.totalFix(cOrderInfo.productCount)
console.log(cOrderInfo.totalPrice);
this.setData({
orderInfo: cOrderInfo
})
}
} else {
if (cOrderInfo.productCount > 1) {
cOrderInfo.productCount -= 1
cOrderInfo.totalPrice = this.totalFix(cOrderInfo.productCount)
this.setData({
orderInfo: cOrderInfo
})
}
}
if (cOrderInfo.totalPrice && this.data.checkedIndex == 2) {
this.setData({
checkedIndex: 0,
wxPayMax: true,
})
}
// this.fixTotalPay()
},
againPay() {
```

```
if (this.data.checkedIndex == 0) {
// 选择支付宝支付
wx.redirectTo({
url: `/order/pages/orderPay/baoPay/baoPay?orderNo=${this.data.orderListNo}&addressId=${this.data.defaultAddress.id}`,
});
} else {
// 选择对公转账支付或者微信支付
this.skipToPay()
}
},
toConfrim() {
let that = this
wx.showModal({
title: "操作提示",
content:'由于您的身份未认证，暂时无法进行提交订单操作，为了给您提供更好的服务，请尽快完成认证~',
cancelColor: "#999999",
cancelText: '放弃认证',
confirmColor: "#36A6FE",
confirmText: '去认证',
success(res) {
if (res.confirm) {
if (that.data.orderInfo.pageFrom == 2) {
let para = {
from: 'launch-order',
}
let data = JSON.stringify(para)
let url = wx.getStorageSync('launchUser').userType == '2' ? `/personal/pages/orgAuthentication/orgAuthentication?data=${data}` : `/personal/pages/perAuthentication/operatePage/operatePage?data=${data}`
app.toGoalUrl(url)
} else {
let url = '/personal/pages/authentication/authentication'
app.toGoalUrl(url)
}
} else if (res.cancel) {
}
}
})
},
// 跳转对公转账支付页
toPay: throttle(function () {
let status = ['2','5','6'];
let certified = [1,2];
if(!status.includes(wx.getStorageSync('userInfo').status)&&!certified.includes(wx.getStorageSync('userInfo').certifiedType)) return this.toConfrim();
// if (wx.getStorageSync('userInfo').authStatus == '0')
if (this.data.defaultAddress.id) {
if (this.data.agreementStatus) {
if (this.data.orderInfo.pageFrom == 0) {
```

```
// 从订单页面过来分两种情况判断
this.skipToPay();
} else {
// 从产品页面过来先提交订单再跳转支付页面
this.checkPayWay()
}
} else {
wx.showToast({
icon: "none",
title: '请勾选《卧涛网交易服务协议》！',
})
}
} else {
wx.showToast({
icon: "none",
title: '请添加收件地址！',
})
}
}, 3000),
// 跳转对公转账支付页
toOrderPay: throttle(function () {
if (!this.data.agreementStatus) return wx.showToast({
icon: "none",
title: '请勾选《卧涛网交易服务协议》！',
})
if (!this.data.defaultAddress.id) return wx.showToast({
icon: "none",
title: '请添加收件地址！',
})
if (this.data.orderInfo.pageFrom == 0) {
// 从订单页面过来分两种情况判断
this.skipToPay();
} else {
// 从产品页面过来先提交订单再跳转支付页面
this.checkPayWay()
}
}, 3000),
// 改变支付方式 0：支付宝 1：线下对公转账支付
changePay(e) {
let index = e.currentTarget.dataset.index
if (this.data.orderInfo.totalPrice > 6000 && index == 2) return app.modalBox('操作提示', '金额大于 6000 请使用其他付款方式！', false);
this.setData({
checkedIndex: index
})
},
// 是否同意《卧涛网交易服务协议》 true：同意 false：不同意
changeAgreement(e) {
let status = e.currentTarget.dataset.status
console.log(status)
this.setData({
    agreementStatus: status
    })
    },
    // 计算价格总价
    fixTotalPay() {
    let info = this.data.orderInfo
    let totalPriceList = this.data.totalPriceList
    let priceType = 'orderInfo.priceType';
    let choiceAttribute = info.choiceAttribute;
    if (choiceAttribute.dueTypeAgencyFee || choiceAttribute.dueTypeOfficialFee ||
choiceAttribute.dueTypeThirdPartyFee) {
    console.log(choiceAttribute, 'due-price');
    totalPriceList[0].title = '服务费总计'
    totalPriceList[0].value = Number(choiceAttribute.dueTypeAgencyFee || 0) +
Number(choiceAttribute.dueTypeThirdPartyFee || 0)
    totalPriceList[1].value = Number(choiceAttribute.dueTypeOfficialFee) || 0
    totalPriceList[2].value = totalPriceList[0].value + totalPriceList[1].value
    this.setData({
    [priceType]: 'due-price', //一口价
    totalPriceList: totalPriceList,
    thirdFee: choiceAttribute.dueTypeOfficialFee
    })
    console.log(this.data.totalPriceList, 'totalPriceList');
    } else if (choiceAttribute.finalTypePendingFee == 0) {
    console.log(choiceAttribute, 'wait-tail');
    totalPriceList[0].value = Number(choiceAttribute.depositTypeAgencyFee || 0) +
Number(choiceAttribute.depositTypeThirdPartyFee || 0)
    totalPriceList[1].value = Number(choiceAttribute.depositTypeOfficialFee || 0)
    totalPriceList[2].value = totalPriceList[0].value + totalPriceList[1].value
    this.setData({
    [priceType]: 'wait-tail', //尾款待定
    totalPriceList: totalPriceList,
    thirdFee: choiceAttribute.depositTypeOfficialFee
    })
    } else {
    let finalIndex = String(Object.keys(choiceAttribute)).indexOf("final"); //
防止数据结构有问题，尾款未设置情况
    console.log(choiceAttribute, 'tail-price');
    totalPriceList[0].value = Number(choiceAttribute.depositTypeAgencyFee || 0) +
Number(choiceAttribute.depositTypeThirdPartyFee || 0)
    totalPriceList[1].value = Number(choiceAttribute.depositTypeOfficialFee || 0)
    totalPriceList[2].value = totalPriceList[0].value + totalPriceList[1].value
    this.setData({
    [priceType]: 'tail-price', //有尾款,
    totalPriceList: totalPriceList,
    finalIndex: finalIndex,
    thirdFee: choiceAttribute.depositTypeOfficialFee,
    tailAmount: Number(choiceAttribute.finalTypeAgencyFee || 0) +
Number(choiceAttribute.finalTypeThirdPartyFee || 0)
    })
    }
    console.log(this.data.totalPriceList, 'totalPriceList');
    let price = 'orderInfo.totalPrice';
    this.setData({
    [price]: this.totalFix(this.data.orderInfo.productCount)
    });
    if (this.data.orderInfo.totalPrice && this.data.checkedIndex == 2) {
    this.setData({
    checkedIndex: 0
    })
    }
    this.getServiceFeeSty();
    },
    // 赋值价格样式数据
    getServiceFeeSty() {
    let info = this.data.orderInfo
    let serviceFee = this.data.serviceFee
    if (info.pageFrom == 1) {
    // 从产品页面进入此页面或者从订单列表和详情（前提 tailstatusPayFlag 必须为 false）进入此页面
    if (info.priceType == 'due-price') {
    serviceFee[0].title = '服务费'
    }
    serviceFee[0].value = this.data.totalPriceList[0].value;
    serviceFee[0].status = info.tailstatusPayFlag == true ? 2 : 0
    serviceFee[0].payProcess = true
    serviceFee[1].value = this.data.finalIndex == -1 ? '待定' : this.data.tailAmount
    serviceFee[1].status = info.tailstatusPayFlag == true ? 0 : 1
    serviceFee[1].payProcess = false
    serviceFee[2].value = this.data.thirdFee ? this.data.thirdFee : 0
    serviceFee[2].status = info.tailstatusPayFlag == true ? 2 : 0
    serviceFee[2].payProcess = true
    console.log(this.data.orderInfo.priceType, '价格类型');
    this.setData({
    serviceFee: serviceFee.filter((item, i) => {
    return item.value != 0 && item.value != '' && item.value != undefined || item.value == 0 && i == 1 && this.data.orderInfo.priceType == 'wait-tail'
    }),
    });
    console.log(this.data.serviceFee, '流程费用 list');
    }
    },
    // 从产品详情页（订单页）进入本页面时对 orderInfo 赋值
    setOrderInfo(data) {
    this.setData({
    orderInfo: data
    });
    console.log(data, '产品详情信息')
    this.fixTotalPay();
    },
    //过滤支付方式
    filterPayways(data) {
    if (data.type == 3 || data.categoryId == 3) {
```

```
let data = this.data.payList
this.setData({
payList: data.splice(0, data.length - 2)
});
}
},
// 地址赋值
fitAddressVal(list) {
let result = list.filter((item) => {
return item.isDefault == 1
})
if (result.length > 0) {
this.setData({
defaultAddress: result[0]
})
} else {
this.setData({
defaultAddress: list[0]
})
}
},
// 判断选择地址
filterAddress(list) {
// 判断是否存储地址 id，是则取用
if (wx.getStorageSync('addressId')) {
let id = wx.getStorageSync('addressId')
console.log(id, "输出存储的 id")
let result = list.filter((item) => {
return item.id == id
})
console.log(result, "输出 result")
if (result.length > 0) {
this.setData({
defaultAddress: result[0]
})
} else {
this.fitAddressVal(list)
}
// 不是取用数组 isDefault 为 1 的数据
} else {
this.fitAddressVal(list)
}
},
// 获取收件地址
getAddressData() {
let params = {
accountId: wx.getStorageSync('userInfo').accountId
}
getAddressList(params).then(res => {
if (res.code == 0) {
if (res.data.length > 0) {
    this.filterAddress(res.data)
    } else {
    this.setData({
    defaultAddress: {}
    })
    }
    }
    })
    },
    // 跳转到支付页
    skipToPay() {
    let info = this.data.transferInfo;
    let orders = this.data.orderInfo;
    if (this.data.checkedIndex == 1) {
    // 跳转对公转账支付页
    let data = JSON.stringify(Object.assign({addressId:
this.data.defaultAddress.id}, this.data.transferInfo))
    wx.redirectTo({
    url: `/order/pages/orderPay/transferPay/transferPay?data=${data}`,
    })
    } else {
    let para = {
    payChannel: this.data.checkedIndex,
    addressId: this.data.defaultAddress.id,
    id: orders.pageFrom == 0 ? orders.feeInfo.paymentId : info.paymentId,
    }
    payForApplet(para).then(res => {
    if (res.code == 0) {
    if (this.data.checkedIndex == 0) {
    // 跳转支付宝支付页
    let pageFrom = this.data.orderInfo.pageFrom
    let orderNo = pageFrom == 0 ? this.data.orderListNo :
this.data.transferInfo.outTradeNo
    wx.redirectTo({
    url:
`/order/pages/orderPay/baoPay/baoPay?orderNo=${orderNo}&addressId=${this.data.d
efaultAddress.id}`,
    })
    } else {
    this.setData({
    wxInfo: JSON.parse(res.data),
    throttle: true
    })
    wx.setStorageSync(’wxPay’, this.data.wxInfo)
    // 跳转微信支付
    wx.navigateTo({
    url:
`/order/pages/orderPay/wxPay/wxPay?addressId=${this.data.defaultAddress.id}`,
    })
    }
    } else {
    wx.showToast({
```

```
title: res.msg,
icon: 'error'
})
}
})
}
},
// 判断支付方式 0-支付宝 1-对公转账 2-微信支付
checkPayWay() {
let checkedIndex = this.data.checkedIndex
let that = this
if (checkedIndex == 2) {
wx.login({
success: function (res) {
let para = {
code: res.code,
}
updateAccountInfo(para).then(res => {
if (res.code == 0) {
wx.setStorageSync('token', res.data)
that.submitOrderData()
} else {
// 调用注销接口强制退出
// that.layoutAccount()
}
})
}
})
} else {
this.submitOrderData()
}
},
// 注销退出
layoutAccount() {
wx.login({
success: function (res) {
layout({
code: res.code
}).then(res => {
if (res.code == 0) {
// 清除所有 storageSync
wx.clearStorageSync()
wx.setStorageSync('hasLogo', false);
wx.redirectTo({
url: "/personal/pages/wxLogin/wxLogin"
})
}
})
}
})
},
// 待付款去支付
submitToPayOrder(params) {
submitToPay(params).then(res => {
if (res.code == 0) {
let r = res.data
wx.showToast({
title: '提交成功！',
})
if (this.data.checkedIndex == 2) {
this.setData({
wxInfo: r.wxPaymentParams
})
}
setTimeout(() => {
this.skipToPay()
}, 100)
} else if (res.code == 50000) {
wx.showToast({
icon: "none",
title: res.msg,
})
setTimeout(() => {
wx.navigateTo({
url: `/personal/pages/orderIndex/order?type=${0}`,
})
}, 100)
}
})
},
// 回显支付方式
getPayInfo(data) {
if (data.pageFrom == 0) {
console.log(data,'*****1');
let info = {
orderNo: data.orderNo + '_' + data.feeInfo.payType,
productName: data.productName,
categoryId: data.categoryId ? data.categoryId : data.type,
totalPrice: data.totalPrice,
tailShould: data.feeInfo.tailShould,
tailstatusPayFlag: data.tailstatusPayFlag,
paymentId: data.feeInfo.paymentId
}
this.setData({
orderNo: data.orderNo,
transferInfo: info,
});
}
if (data.pageFrom != 1) {
this.setData({
orderInfo: data
});
```

```
this.orderListFee();
}
},
orderListFee() {
let fee = this.data.orderInfo.feeInfo;
let serviceFee = this.data.serviceFee;
let totalPriceList = this.data.totalPriceList
if (this.data.orderInfo.tailstatusPayFlag) { //支付尾款修改列表状态
serviceFee[0].payProcess = serviceFee[2].payProcess = false
serviceFee[0].status = serviceFee[2].status = 2
serviceFee[1].payProcess = true
serviceFee[1].status = 0
}
if (fee.isDue) {
serviceFee[0].title = '服务费';
serviceFee[0].value = fee.dueDeposit;
totalPriceList[0].value = fee.dueDeposit;
};
if (fee.hasDeposit) {
serviceFee[0].value = fee.deposit
totalPriceList[0].value = fee.deposit;
serviceFee[1].value = fee.tailPrice ? fee.tailPrice : fee.tailStatus
};
if (fee.hasOffical) {
serviceFee[2].value = fee.offical
totalPriceList[1].value = fee.offical
}
totalPriceList[2].value = totalPriceList[0].value + totalPriceList[1].value
this.setData({
serviceFee: serviceFee.filter((item, i) => {
return item.value != 0 && item.value != ''
}),
orderListNo: this.data.orderInfo.orderNo + '_' + fee.payType,
totalPriceList: totalPriceList,
});
console.log(this.data.totalPriceList, '*******');
},
closeDialog(e) {
this.setData({
dialogStatus: e.detail
})
if (!e.detail) {
wx.disableAlertBeforeUnload({
success: (res) => {},
})
} else {
let pageFrom = this.data.orderInfo.pageFrom
let msg = pageFrom == 1 ? '是否取消订单' : '是否放弃支付'
wx.enableAlertBeforeUnload({
message: `${msg}，返回上一级页面？`,
})
```

```
}
},
// 离开页面弹框（手势离开）
leaveEnable() {
let pageFrom = this.data.orderInfo.pageFrom
let msg = pageFrom == 1 ? '是否取消订单' : '是否放弃支付'
wx.enableAlertBeforeUnload({
message: `${msg}，返回上一级页面？`,
})
},
// 跳转项目申报交易服务协议
toAgreement() {
wx.navigateTo({
url: '/order/pages/tsAgreement/projectTa/projectTa',
})
},
onLoad: function (options) {
console.log(options,'----');
let data = JSON.parse(decodeURIComponent(options.data))//防止值出现特殊字符如?、&、=等，导致转换不了报错
this.filterPayways(data)
if (data.pageFrom == 1) {
this.setOrderInfo(data)
};
this.getPayInfo(data)
if (this.data.dialogStatus) {
this.leaveEnable();
}
if (wx.getStorageSync('userInfo')) {
this.setData({
userInfo: wx.getStorageSync('userInfo')
})
}
},
onShow: function () {
// 订单列表和详情进入此页面无法修改收件地址
let orderInfo = this.data.orderInfo
if (orderInfo.pageFrom == 0 || orderInfo.pageFrom == 2) {
this.getAddressData()
} else {
let defaultAddress = this.data.defaultAddress
defaultAddress.id = orderInfo.receiveAddressId
this.setData({
defaultAddress: defaultAddress
})
}
},
onUnload: function () {
wx.removeStorageSync('addressId')
},
})
```