

IMMI源码手册

2018/08/16



目录

[1. 程序目录结构 3](#_Toc522198336)

[2. 关键代码 3](#_Toc522198337)

[2.1 app.json小程序页面、窗体、导航配置 3](#_Toc522198338)

[2.2 app.js注册小程序，get、post请求封装，本地存储方法的封装 5](#_Toc522198339)

[2.3 app.wxss 小程序总的样式文件 9](#_Toc522198340)

[2.4 config.js 项目请求接口路径汇总 10](#_Toc522198341)

[2.5 首页 12](#_Toc522198342)

[2.6 商品详情 14](#_Toc522198343)

[2.7 购物车 15](#_Toc522198344)

[2.8 我的页面 18](#_Toc522198345)

[2.9 商品列表 19](#_Toc522198346)

[2.10 我的订单列表 20](#_Toc522198347)

[2.11 退货单列表 22](#_Toc522198348)

[2.12 补货单列表 25](#_Toc522198349)

[2.13 退货单、补货单详情 25](#_Toc522198350)

[2.14 收货人地址列表和选择列表 27](#_Toc522198351)

[2.15 收货人地址详情 28](#_Toc522198352)

[2.16 登录、退出登录页面 30](#_Toc522198353)

# 程序目录结构

* common 公共文件夹
  + lib 第三方组件库
  + resource 资源文件（图片）
* components 自定义组件文件夹
* pages 项目页面文件夹
* utils 工具类文件夹
* app.js 注册微信小程序
* app.json 小程序页面、窗体、导航配置
* app.wxss 小程序总样式文件
* config.js 本项目所有接口汇总文件
* project.config.json 项目配置

# 关键代码

## 2.1 app.json小程序页面、窗体、导航配置

**说明：**app.json配置整个页面的页面路径，顶部导航字体颜色、背景颜色、文字， 底部tabBar

**源码：**

{

"pages":[ // 页面配置

"pages/index/index", // 首页

"pages/pdtInfo/index", // 商品详情页

"pages/account/index", // 我的页面

"pages/webview/index", // 第三方页面

"pages/pdtList/index", // 全部商品列表

"pages/orderList/index", // 我的订单列表

"pages/backOrderList/index", // 补货单列表

"pages/returnOrderList/index",//退货单列表

"pages/cart/index",//购物车列表

"pages/contactsChoose/contactsChoose",//收货人地址选择列表

"pages/contactsUpdate/contactsUpdate",//收货人地址详情

"pages/contactsManager/contactsManager",//收货人地址列表

"pages/login/index",//登录

"pages/orderDetailList/index"//退货单、补货单详情

],

"window":{ // 顶部导航配置

"backgroundTextStyle":"light",

"navigationBarBackgroundColor": "#fff",

"navigationBarTitleText": "IMMI",

"navigationBarTextStyle":"black",

"onReachBottomDistance": "50"

},

"tabBar": { // 底部导航配置

"color": "#ffffff",

"selectedColor": "#ffffff",

"borderStyle": "black",

"backgroundColor": "#000000",

"list": [

{

"pagePath": "pages/index/index",

"iconPath": "common/resource/home.png",

"selectedIconPath": "common/resource/home-selected.png",

"text": "首页"

},

{

"pagePath": "pages/cart/index",

"iconPath": "common/resource/cart.png",

"selectedIconPath": "common/resource/cart-selected.png",

"text": "购物车"

},

{

"pagePath": "pages/account/index",

"iconPath": "common/resource/account.png",

"selectedIconPath": "common/resource/account-selected.png",

"text": "我的"

}

]

}

}

## 2.2 app.js注册小程序，get、post请求封装，本地存储方法的封装

**说明：**app({})注册一个小程序，nGet对wx.request({method: ‘get’})进行封装，nPost对wx.request({method: ‘post})进行封装,主要抽取共同处理接口错误码冗余的部分，省去了每个接口重复的工作。本地存储方法封装。弹框全局的封装。

**源码：**

// ======网络请求======

/\*\* get 请求

var data = {

url: config.loginP,

params: {

}

}

app.nGet(data).then(data => {}, res => {});

\*/

nGet: function(data) {

var \_that = this;

let sessionKey = this.getValue('sessionKey') || '';

let cityId = this.getValue('cityCode') || '310000';

let promise = new Promise(function(resolve, reject) {

wx.request({

url: data.url,

data: data.params,

method: "GET",

header: {

'content-type': 'application/x-www-form-urlencoded;charset=utf-8',

'session-Token': sessionKey,

'city': cityId,

},

success: function(res) {},

fail: function(res) {},

complete: function(res) {

if (res.data && res.data.code === 666) {

resolve(res.data);

} else if (res.data && res.data.code === 8006 || res.data && res.data.code === 8005) {

\_that.clearValue();

wx.reLaunch({

url: '/pages/login/index?page=first'

});

} else {

reject(res.data);

if (res.data && res.data.message) {

\_that.showMsg(res.data.message);

}

}

}

});

});

return promise;

},

/\*\* post 请求

var data = {

url: config.loginP,

params: {

}

}

app.nPost(data).then(data => {}, res => {});

\*/

nPost: function(data) {

var \_that = this;

let sessionKey = this.getValue('sessionKey') || '';

let cityId = this.getValue('cityCode') || '310000';

let promise = new Promise(function(resolve, reject) {

wx.request({

url: data.url,

data: data.params,

method: "POST",

header: {

'content-type': 'application/x-www-form-urlencoded;charset=utf-8',

'session-Token': sessionKey,

'city': cityId,

},

success: function(res) {},

fail: function(res) {},

complete: function(res) {

if (res.data && res.data.code === 666) {

resolve(res.data);

} else if (res.data && res.data.code === 8006 || res.data && res.data.code === 8005) {

\_that.clearValue();

wx.reLaunch({

url: '/pages/login/index?page=first'

});

} else {

reject(res.data);

if (res.data && res.data.message) {

\_that.showMsg(res.data.message);

}

}

}

});

});

return promise;

},

// ======本地存储======

saveValue: function(key, value) {

wx.setStorageSync(key, value);

},

getValue: function(key) {

return wx.getStorageSync(key);

},

clearValue: function() {

wx.clearStorageSync();

},

// ======提示框======

showMsg: function(msg, rduration) {

if (!rduration) {

rduration = 2;

}

rduration = 1000 \* rduration;

wx.showToast({

title: msg,

icon: 'none',

duration: rduration

})

}

## 2.3 app.wxss 小程序总的样式文件

**说明：** 小程序总的样式文件和一些第三方样式引入

**源码：**

/\*\*app.wxss\*\*/

@import './common/lib/zanUI/index.wxss';

.container {

width: 100%;

height: 100%;

display: flex;

flex-direction: column;

align-items: center;

justify-content: space-between;

box-sizing: border-box;

font-size: 28rpx;

}

## 2.4 config.js 项目请求接口路径汇总

**说明：** 项目所有接口文件汇总，便于后期的修改和维护

**源码：**

/\*\*

\* 接口配置文件

\*/

const host = "https://immi.dingliantech.com";

// const host = "http://192.168.3.16:8080/trendfinder";

// 接口汇总

let api = {

// 登录

login: `${host}/b2b/login`,

// 退出登录

logout: `${host}/b2b/logout`,

// 首页幻灯片

indexCarouselQuery: `${host}/b2b/product/productCarousel`,

// 商品列表

indexNewPdtQuery: `${host}/b2b/product/productList`,

// 商品分类查询

indexCategoryQuery: `${host}/b2b/product/getCategory`,

// 商品筛选列表

indexFilterQuery: `${host}/b2b/product/getSearchCriteria`,

// 获取商品详情

pdtGet: `${host}/b2b/product/productInfo`,

// 获取规格

specQuery: `${host}/b2b/product/productColorSize`,

// 添加|修改购物车

addCart: `${host}/b2b/cart/saveCartItem`,

// 购物车列表

cartListQuery: `${host}/b2b/cart/cartList`,

//购物车计算

cartCalculate: `${host}/b2b/cart/calculateAmount`,

// 购物车提交

orderCreate: `${host}/b2b/cart/submitOrder`,

// 购物车删除

cartDelete: `${host}/b2b/cart/cartDelete`,

// 获取地址列表

addressList: `${host}/b2b/user/shipAddressList`,

// 地址删除

addressDelete: `${host}/b2b/user/removeShipAddress`,

// 获取地址详情

addressGet: `${host}/b2b/user/shipAddressInfo`,

// 地址修改|保存

addressCreate: `${host}/b2b/user/saveShipAddress`,

// 获取订单列表

orderQuery: `${host}/b2b/order/myOrder`,

// 获取退货单|补货单列表

rOrderQuery: `${host}/b2b/order/orderList`,

// 获取退货单|补货单详情

rOrderGet: `${host}/b2b/order/orderInfo`,

};

var config = {

host,

...api

};

module.exports = config;

## 2.5 首页

**说明：** 首页分为三部分：今日新品、全部、筛选。今日新品分为上面的幻灯片、和下面一行两个商品的商品列表；全部分为搜索、分类和一行三个商品的商品列表；筛选分类各个商品属性的列表。代码将幻灯片，和商品列表作为自定义组件便于多个页面的重复使用。

**源码：**

{ // 首页用到的自定义组件

"usingComponents": {

"zan-tab": "/common/lib/zanUI/tab/index",

"zan-search": "/common/lib/zanUI/search/index",

"carousel": "/components/carousel/carousel",

"one-cell-two-pdt": "/components/oneCellTwoPdt/index",

"one-cell-three-pdt": "/components/oneCellThreePdt/index",

"one-cell-four-icon": "/components/oneCellFourIcon/index",

"filter-item": "/components/filterItem/index",

"zan-loadmore": "/common/lib/zanUI/loadmore/index"

}

}

<!--index.wxml页面代码-->

<view class="container">

<view id="tab-container">

<zan-tab

id="index-tab"

list="{{ tab.list }}"

selected-id="{{ selectedId }}"

bindtabchange="handleTabChange"

/>

</view>

<view id="index-container">

<!-- 今日新品 -->

<view wx:if="{{selectedId === 'new'}}">

<carousel items="{{carouselList}}" style="width: 100%;"></carousel>

<one-cell-two-pdt items="{{recommendInfo.list}}"></one-cell-two-pdt>

<zan-loadmore type="loading" wx:if="{{newLoadMore.isLoad}}"></zan-loadmore>

<zan-loadmore type="text" wx:elif="{{newLoadMore.title}}" text="{{newLoadMore.title}}"></zan-loadmore>

</view>

<!-- 全部 -->

<view wx:elif="{{selectedId === 'all'}}">

<zan-search

placeholder="搜索..."

keyword="{{ inputValue }}"

placeholder="搜索您想要的商品"

searchStyle="height: 70rpx;border-radius: 50rpx;background: {{ searchbg || '#F2F2F2' }};"

inputStyle="height: 30rpx; background: {{ searchbg || '#F2F2F2' }};"

bindchange="searchChange"

bindsearch="searchDone"

/>

<one-cell-four-icon items="{{categoryList}}"></one-cell-four-icon>

<one-cell-three-pdt items="{{allInfo.list}}"></one-cell-three-pdt>

<zan-loadmore type="loading" wx:if="{{allLoadMore.isLoad}}"></zan-loadmore>

<zan-loadmore type="text" text="{{allLoadMore.title}}" wx:elif="{{allLoadMore.title}}"></zan-loadmore>

</view>

<!-- 筛选 -->

<view wx:elif="{{selectedId === 'filter'}}">

<filter-item datas="{{attrsList}}"></filter-item>

</view>

</view>

</view>

## 2.6 商品详情

**说明：** 商品详情即为补货界面，分为两个部分商品基本属性描述、弹框出下单矩阵。弹框和下单矩阵为一个自定义组件，便于项目重复使用。

**源码：**

// 商品详情的自定义组件引入

{

"navigationBarTitleText": "尺寸下单",

"usingComponents": {

"order-matrix": "/components/orderMatrix/index",

"pop-bottom": "/components/popBottom/popBottom"

}

}

<!--pages/pdtInfo/index.wxml-->

<view class="pdtInfo-page">

<!-- 图片列表 -->

<swiper class="carousel-swiper" indicator-dots="{{indicatorDots}}" autoplay="{{autoplay}}" interval="{{interval}}" duration="{{duration}}" circular="{{circular}}">

<block wx:for="{{pdtInfo.images}}" wx:key="{{item}}">

<swiper-item>

<image src="{{item}}" class="carousel-image slide-image" />

</swiper-item>

</block>

</swiper>

<view class="pdtInfo-body">

<view class="productName">{{pdtInfo.productName}}</view>

<view class="styleName">{{pdtInfo.styleName}}</view>

<scroll-view class="pdtAttrList" scroll-x>

<text class="pdtAttr" wx:for="{{pdtInfo.attrList}}" wx:key="{{item}}">{{item.attrName}}</text>

</scroll-view>

<view class="productPrice">

<text class="pricelist">¥{{pdtInfo.pricelist}}</text>

<text class="price">¥{{pdtInfo.price}}</text>

</view>

</view>

</view>

<view class="pdtInfo-btn" bindtap='showPopUp'>尺寸下单</view>

<pop-bottom id="popup" bind:cancel="dismissPop">

<order-matrix productId="{{productId||''}}" from="pdtInfo" orderType="backOrder" bind:reset="reset" bind:success="successOrder"></order-matrix>

</pop-bottom>

## 2.7 购物车

**说明：** 购物车分为补货列表和退货列表。支持左滑删除，全选结算，选择收货人地址。

**源码：**

**//** 购物车自定义组件引入

{

"navigationBarTitleText": "购物车",

"usingComponents": {

"zan-tab": "/common/lib/zanUI/tab/index",

"zan-icon": "/common/lib/zanUI/icon/index",

"prod-list-cell": "/components/prodListCell/prodListCell",

"pop-bottom": "/components/popBottom/popBottom",

"zan-loadmore": "/common/lib/zanUI/loadmore/index"

}

}

**//** 购物车页面

<view class='contents'>

<view class='tab'>

<zan-tab list="{{ tab.list }}" selected-id="{{ tab.selectedId }}" bind:tabchange="tabchange" />

</view>

<block wx:for="{{productsList}}" wx:for-index="idx" wx:for-item="product" wx:key="{{idx}}">

<prod-list-cell checkObj="{{checkStyleObj}}" index="{{idx}}" shopData="{{product}}" showType="{{showType}}" canCheck="true" bind:change="changeAction" bind:check="checkAction" bind:del="deleteAction"></prod-list-cell>

</block>

<zan-loadmore type="loading" wx:if="{{!loadDone}}"></zan-loadmore>

<zan-loadmore type="nodata" text="" wx:if="{{loadDone}}"></zan-loadmore>

</view>

<view class='butt'>

<view class='block block1' bindtap='checkAllAction'>

<view class='check'>

<zan-icon type="{{isCheckAll? 'checked':'check'}}"></zan-icon>

</view>

<view class='msg'>

<text>全选</text>

</view>

</view>

<view class='block block2'>

<view class='row1'>

<text class=''>总计:</text>

<text class='amount'>￥{{sumMoney}}</text>

</view>

<view class='row2'>

<text class=''>总量:</text>

<text class=''>{{sumCount}}</text>

</view>

</view>

<view class='block block3' bindtap='calculationAction'>

结算

</view>

<!-- <button bindtap='calculationAction'>结算</button> -->

</view>

<pop-bottom id="popup" bind:cancel="dismissPop">

<view class='submit'>

<view class='row1'>

<text>总定价:{{sumMoney}}</text>

<!-- <text>总定量:{{sumCount}}</text> -->

</view>

<view class='row2'>

<!-- <text>总定价:{{sumMoney}}</text> -->

<text>总定量:{{sumCount}}</text>

</view>

<view class='row3' bindtap='chooseAddressAction'>

<text>{{address.length>0? address:"请选择收货地址"}}</text>

<view class='arrow'>

<zan-icon type='arrow'></zan-icon>

</view>

</view>

<view class='row bot'>

<button bindtap='submitOrderAction'>提交</button>

</view>

</view>

</pop-bottom>

## 2.8 我的页面

**说明：** 我的页面包含用户名显示，我的订单、退货单、补货单、收货人地址列表入口

**源码：**

<!--pages/account/index.wxml-->

<view class='contents'>

<!-- 遮罩层 -->

<view class='bgView'>

<image src='/common/resource/backgroundImage.jpg' mode="aspectFill"></image>

</view>

<view class='maskView'></view>

<!-- 内容 -->

<view class="accountPage">

<view class="profile-info" bindtap="checkLogin">

<view>

<view class="profile-name">{{userName ? userName : '请登录'}}</view>

<view class="profile-photo">

<image src="../../common/resource/defaultPhoto.png"></image>

</view>

</view>

</view>

<view class='tabview'>

<zan-cell title="我的订单" isLink="true" bindtap="enterOtherPage" data-page="orderList"></zan-cell>

<view class='line'></view>

<zan-cell title="补货单" isLink="true" bindtap="enterOtherPage" data-page="returnOrderList"></zan-cell>

<view class='line'></view>

<zan-cell title="退货单" isLink="true" bindtap="enterOtherPage" data-page="backOrderList"></zan-cell>

<view class='line'></view>

<zan-cell title="我的地址" isLink="true" bindtap="enterOtherPage" data-page="contactsManager"></zan-cell>

</view>

</view>

</view>

## 2.9 商品列表

**说明：** 搜索货号或者商品属性过滤进入符合条件的商品列表。该列表支持上拉加载。图片懒加载。本页面将一行两个商品做成自定组件，页面代码更加简洁。可复用性更强。

**源码：**

**//** 本页面引入的自定义组件

{

"navigationBarTitleText": "商品列表",

"usingComponents": {

"zan-loadmore": "/common/lib/zanUI/loadmore/index",

"one-cell-two-pdt": "/components/oneCellTwoPdt/index"

}

}

<!--pages/pdtList/index.wxml商品列表-->

<view class="pdtList-page">

<one-cell-two-pdt items="{{list}}"></one-cell-two-pdt>

<zan-loadmore type="loading" wx:if="{{loadMore.isLoad}}"></zan-loadmore>

<zan-loadmore type="text" text="{{loadMore.title}}" wx:elif="{{loadMore.title}}"></zan-loadmore>

</view>

## 2.10 我的订单列表

**说明：** 退货单和补货单汇总。该列表支持上拉加载。

**源码：**

/\*\* 加载数据 \*/

prepareData() {

// 分页加载

var page = 1;

var sData = this.data.ordersData;

if (sData.page && sData.page < sData.pages) {

page = sData.page + 1;

} else {

this.data.ordersData.list = [];

page = 1;

}

var paramData = {

url: config.orderQuery,

params: {

search: this.data.inputValue,

size: 8,

page: page,

}

};

app.nGet(paramData).then(ret => {

if (ret.data) {

if (ret.data.page >= ret.data.pages) {

this.setData({

loadDone: true,

});

}

if (ret.data.page === 1 && ret.data.list && ret.data.list.length === 0) {

// console.log('@@@@@' + this.data.loadDone);

this.setData({

title: '暂无数据'

});

// console.log('@@@@@' + this.data.title);

}

ret.data.list = [...this.data.ordersData.list, ...ret.data.list];

this.setData({

ordersData: ret.data

});

}

}, res => {

// console.error(JSON.stringify(res));

});

}

## 2.11 退货单列表

**说明：** 所有退货单汇总

**源码：**

const app = getApp();

import config from '../../config.js';

Page({

/\*\*

\* 页面的初始数据

\*/

data: {

ordersData: {},

loadDone: false,

},

/\*\*

\* 生命周期函数--监听页面加载

\*/

onLoad: function(options) {

this.prepareData();

},

/\*\*

\* 页面相关事件处理函数--监听用户下拉动作

\*/

onPullDownRefresh: function() {

this.data.ordersData = {};

this.preparePageData();

wx.stopPullDownRefresh();

},

/\*\*

\* 页面上拉触底事件的处理函数

\*/

onReachBottom: function() {

if (!this.data.loadDone) {

this.prepareData();

}

},

// 点击订单，进入订单详情页

orderClickAction: function(e) {

// console.log(JSON.stringify(e.currentTarget.dataset.data));

wx.navigateTo({

url: '/pages/orderDetailList/index?orderType=' + '02' + '&orderID=' + e.currentTarget.dataset.data.orderId,

});

},

// ----------------------------

/\*\* 加载数据 \*/

prepareData() {

// 分页加载

var page = 1;

var sData = this.data.ordersData;

if (sData.page && sData.page < sData.pages) {

page = sData.page + 1;

} else {

this.data.ordersData.list = [];

page = 1;

}

var data = {

url: config.rOrderQuery,

params: {

page: page,

size: 10,

orderType: 'backOrder',

}

};

app.nGet(data).then(ret => {

if (ret.data) {

if (ret.data.page >= ret.data.pages) {

this.setData({

loadDone: true,

});

}

ret.data.list = [...this.data.ordersData.list, ...ret.data.list];

this.setData({

ordersData: ret.data

});

}

}, res => {

// console.error(JSON.stringify(res));

});

},

})

## 2.12 补货单列表

**说明：** 所有补货单汇总

**源码：** 代码结构同退货单列表

## 2.13 退货单、补货单详情

**说明：** 退货单详情和补货单详情为同一个页面，通过不同的参数获取不同的数据。

**源码：**

// pages/orderList/index.js

const app = getApp();

import config from '../../config.js';

Page({

/\*\*

\* 页面的初始数据

\*/

data: {

orderConfig: {},

ordersData: [],

orderType: '', // '02':补货单; '03':退货单;

orderID: '', // 补退货单 订单号

showType: {} // 下单组件所需要参数

},

/\*\*

\* 生命周期函数--监听页面加载

\*/

onLoad: function(options) {

var orderType = options.orderType || '';

var orderID = options.orderID || '';

var title = "";

var showType = {

from: 'rOrderInfo',

orderType: ''

};

// console.log(orderType);

if (orderType == '02') {

title = "补货单";

showType.orderType = "backOrder";

} else if (orderType == '03') {

title = "退货单";

showType.orderType = "returnOrder";

}

wx.setNavigationBarTitle({

title: title,

})

this.setData({

orderType: orderType,

orderID: orderID,

showType: showType

});

this.preparePageData(); // 加载页面数据

},

/\*\* 加载数据 \*/

preparePageData() {

var paramData = {

url: config.rOrderGet,

params: {

orderId: this.data.orderID

}

};

app.nGet(paramData).then(ret => {

if (ret.data) {

// for (var i = 0; i < ret.data.list.length; i++) {

// ret.data.list[i].isCheck = false;

// }

let orderConfig = {

orderNumber: ret.data.orderNumber,

orderStatus: ret.data.orderStatus,

totalAmount: ret.data.totalAmount,

totalQty: ret.data.totalQty,

address: ret.data.address,

};

this.setData({

orderConfig: orderConfig,

ordersData: ret.data.list

});

// console.log(JSON.stringify(ret.data.list));

}

}, res => {

// console.error(JSON.stringify(res));

});

},

})

## 2.14 收货人地址列表和选择列表

**说明：** 收货人地址列表，抽取了共同每列做成了自定义组件。

**源码：**

**//** 选择收货人引入的组件

{

"navigationBarTitleText": "选择联系人",

"usingComponents": {

"contacts-cell": "/components/contactsCell/contactsCell"

},

"enablePullDownRefresh": false

}

**//** 地址列表页面代码

<view class='contents'>

<block wx:for="{{contactsData}}" wx:for-index="idx" wx:for-item="contact" wx:key="{{idx}}">

<view class='list' bindtap='chooseContact' data-contact='{{contact}}'>

<contacts-cell contactData="{{contact}}"></contacts-cell>

</view>

</block>

</view>

<view class='butt'>

<button bindtap='addAddressAction'>添加新地址</button>

</view>

## 2.15 收货人地址详情

**说明：** 收货人用户名，手机号，收件地址，详细地址进行维护。

**源码：**

<view class='contents'>

<view class='row'>

<input class='inp' placeholder='请输入收货人姓名' value='{{consignee}}' bindinput="inputChange" data-type='01'></input>

</view>

<view class='row'>

<input class='inp' placeholder='请输入手机号码' input-type='number' value='{{phone}}' bindinput="inputChange" data-type='02'></input>

</view>

<picker mode="region" bindchange="bindRegionChange" value="{{region}}">

<view class='row'>

<text class='picker'>{{region[0]}}{{region[1]}}{{region[2]}}</text>

<view class='arrow'>

<zan-icon type="arrow"></zan-icon>

</view>

</view>

</picker>

<view class='row'>

<textarea class='txa' placeholder='请输入详细地址(最多50字)' auto-height maxlength="50" value='{{detail}}' bindinput="inputChange" data-type='03'></textarea>

</view>

</view>

<view class='butt'>

<button bindtap='addAddressAction'>保存</button>

</view>

<!-- addressList 获取地址列表 -->

## 2.16 登录、退出登录页面

**说明：** 通过判断storage里是否有username判断用户是否登录，如果未登录，用户登录；如果登录，允许退出登录

**源码：**

<!--pages/login/index.wxml-->

<view class="login-page">

<view wx:if="{{!isLogin}}">

<zan-field title="{{ userName.title }}" placeholder="{{ userName.placeholder }}" type="input" value="{{ userName.value }}" bind:change="handleUserNameChange">

</zan-field>

<zan-field title="{{ password.title }}" placeholder="{{ password.placeholder }}" value="{{ password.value }}" type="input" inputType="password" bind:change="handlePawwordChange">

</zan-field>

</view>

<view class="login-btn">

<zan-button size="" bindtap="handleBtn">{{isLogin ? "退出登录" : "登录"}}</zan-button>

</view>

</view>