一、程序要实现的功能

(1)商品的展示

(2)商品加入购物车

(3)商品的结算(因为无法申请到支付模块，所以只能完成到商品的结算，无法支付)

(4)商品根据商品类型展示出来

(5)商品列表的城市选择

(6)商品展示页轮播图

(7)商品详情页轮播图

(8)商品展示的上拉刷新、下拉还原

(9)个人中心

(10)登录及注册

(11)消息页面的打开展示

(12)客服页面的打开展示

(13)个人中心的订单、账户、优惠卷、收货地址页面的展示

(14)收货地址页面的地址的添加及删除

(15)个人中心的头像的设置

(16)app的设置、关于及缓存

二、功能的实现过程及相关代码

(1)商品的展示

1.通过Ajax请求云数据库，拿到商品数据，然后放到template模板上展示出来，具体实现代码：

function fnGetWareList(loadMore\_)

//判断第几次打开

if (loadMore\_) {

skip += LIMIT;

} else {

skip = 0;

}

// 根据城市和商品分类获得相应的商品列表

var params = {

fields: {},

where: {

// supportAreaId: currentCity.id,

wareTypeId: wareTypeList[api.pageParam.wareTypeIndex].id

},

skip: skip,

limit: LIMIT

}

params = $api.jsonToStr(params);

//计算加密后的appKey

var now = Date.now();

var appKey = SHA1(api.appId + "UZ" + "66DBD6E6-E309-5421-B020-A24C69A98BDB" + "UZ" + now) + "." + now;

api.ajax({

url: 'http://d.apicloud.com/mcm/api/ware?filter=' + params,

method: 'get',

headers: {

"X-APICloud-AppId": api.appId,

"X-APICloud-AppKey": appKey

}

}, function(ret, err) {

if (ret) {

// 恢复下拉刷新的状态

api.refreshHeaderLoadDone();

// 在界面中更新商品列表显示

fnUpdateWareList(ret, loadMore\_);

} else {

api.toast({

msg: '加载数据失败',

duration: 2000,

location: 'bottom'

});

}

// 隐藏加载状态对话框

api.hideProgress();

});

}

// 更新商品列表显示

function fnUpdateWareList(data\_, loadMore\_) {

var list = $api.byId('list');

// 编译模板函数

var tempFn = doT.template($api.byId('template').innerHTML);

// 使用模板函数生成HTML文本

var resultHTML = tempFn(data\_);

// 判断是否是加载更多，如果是加载更多，则追加到list中

if (loadMore\_) {

$api.append(list, resultHTML);

// 如果服务器端已经没有更多数据返回，更新提示信息

if (data\_.length < LIMIT) {

var pushStatus = $api.byId('pushStatus');

pushStatus.innerHTML = "没有啦！";

}

} else {

// 否则，直接替换list中的内容

$api.html(list, resultHTML);

// updateBanner();

}

// 由于是动态的将元素添加到Dom树上，所以需要手动触发tapmode检查，列表中的元素才能实现点击加速的效果

api.parseTapmode();

// 如果商品已存在购物成中，显示商品在购物成中的数量

fnShowWareCountInShoppingCart(data\_);

}

(2)商品加入购物车

1.通过监听商品的点击事件来判断商品是否加入了购物车，具体代码实现:

// 添加到购物车按钮的监听函数

function fnAdd(wareId\_) {

// 阻止事件继续向上冒泡

event.stopPropagation();

var count = $api.byId('count\_' + wareId\_);

var panel = $api.byId('panel\_' + wareId\_);

// 更新商品数量及显示状态

var countNumber = parseInt(count.innerHTML);

countNumber += 1;

count.innerHTML = countNumber;

panel.style.display = (countNumber > 0) ? 'block' : 'none';

// 发送购物车更新自定义事件，传递商品Id和商品数量

api.sendEvent({

name: 'updateShoppingCart',

extra: {

wareId: wareId\_,

wareCount: countNumber

}

});

}

2.从购物车删除按钮的监听函数

function fnMinus(wareId\_) {

event.stopPropagation();

var count = $api.byId('count\_' + wareId\_);

var panel = $api.byId('panel\_' + wareId\_);

var countNumber = parseInt(count.innerHTML);

countNumber -= 1;

count.innerHTML = countNumber;

panel.style.display = (countNumber > 0) ? 'block' : 'none';

api.sendEvent({

name: 'updateShoppingCart',

extra: {

wareId: wareId\_,

wareCount: countNumber

}

});

}

3.显示商品在购物车中的数量

function fnShowWareCountInShoppingCart(data\_) {

// 遍历商品列表

for (var i = 0; i < data\_.length; i++) {

var result = $db.select(data\_[i].id);

if (result && result.data.length > 0) {

// 获得商品在购物车中的数量，更新数量和状态

var wareCount = result.data[0].wareCount;

var count = $api.byId('count\_' + data\_[i].id);

var panel = $api.byId('panel\_' + data\_[i].id);

count.innerHTML = wareCount;

panel.style.display = (wareCount > 0) ? 'block' : 'none';

}

}

}

(3)商品的结算(因为无法申请到支付模块，所以只能完成到商品的结算，无法支付)

1.打开购物车函数

api.openFrame({

name: 'shoppingcart\_frame',

url: './shoppingcart\_frame.html',

rect: {

marginTop: headerH,

marginBottom: footerH,

w: 'auto'

},

pageParam: {

wareId: api.pageParam.wareId

},

bounces: true

});

};

2.设置购物车中的商品数量和金额，被购物车Frame调用

function fnSetWareCountAndAmount (count\_, amount\_) {

var count = $api.byId('count');

var amount = $api.byId('amount');

count.innerHTML = count\_;

amount.innerHTML = '￥' + amount\_;

}

3.打开商品确认订单页面

function fnOpenOrderWin () {

api.openWin({

name: 'order',

url: './order.html'

});

}

(4)商品根据商品类型展示出来

1.打开商品列表FrameGroup和mini购物车Frame

function openFrames() {

// 判断这些Frame是否已经打开

if (framesOpened) {

return;

}

// 构造FrameGroup中的Frame数组

var frames = [];

for (var i = 0; i < menus.length; i++) {

frames.push({

name: 'main\_frame\_' + i,

url: './main\_frame.html',

pageParam: {

wareTypeIndex: i

}

});

}

// 打开FrameGroup

api.openFrameGroup({

name: 'mainFrameGroup',

scrollEnabled: true, // 支持手势滑动切换

rect: {

marginTop: headerH + navH, // 采用margin布局

w: 'auto', // 自动填充所在Window的宽度

h: 'auto' // 自动填充所在Window的高度

},

index: 0, // 默认显示第一个Frame

frames: frames,

preload: frames.length // 预加载所有Frame

}, function(ret, err) {

// Frame切换完成的响应函数，更新分类菜单的选中状态

fnSetNavMenuSelected(ret.index);

});

2.打开mini购物车Frame

api.openFrame({

name: 'minicart\_frame',

url: './minicart\_frame.html',

rect: {

x: 0,

y: api.winHeight - 55,

w: 150,

h: 34

},

bounces: false // 关闭弹动

});

// 将mini购物车Frame放置在首页Window所有Frame的最上层

api.bringFrameToFront({

from: 'minicart\_frame'

});

// 标记为已经打开

framesOpened = true;

}

3.分类菜单点击的响应函数，切换Frame

function fnSetNavMenuIndex(index\_) {

// 首先更新菜单选中状态

fnSetNavMenuSelected(index\_);

// 切换FrameGroup中的当前Frame

api.setFrameGroupIndex({

name: 'mainFrameGroup',

index: index\_,

scroll: true

});

}

4.更新分类菜单的选中状态

function fnSetNavMenuSelected(index\_) {

for (var i = 0; i < menus.length; i++) {

if (index\_ == i) {

$api.addCls(menus[i], 'selected');

} else {

$api.removeCls(menus[i], 'selected');

}

}

}

(5)商品列表的城市选择

1.初始化事件监听

function initEventListenner() {

// 监听城市切换事件（自定义事件）

api.addEventListener({

name: 'cityChange'

}, function(ret, err) {

if (ret) {

// 更新城市名称

updateCityName(ret.value.currentCity.name);

// 更新箭头状态

resetArrowState();

// 打开商品列表等首页的Frames

openFrames();

}

});

2.拦截Android的返回键，在首页点击返回键，提示退出应用

api.addEventListener({

name: 'keyback'

}, function(ret, err) {

api.confirm({

title: '提示',

msg: '是否退出应用',

buttons: ['确定', '取消']

}, function(ret, err) {

if (ret.buttonIndex == 1) {

// 关闭当前的主Widget，就可以实现推出APP的效果

api.closeWidget({

silent: true //直接退出，无需提示

});

}

});

});

}

3.加载百度地图模块获取当前选择的城市的经纬坐标

function initLocation() {

// 加载百度地图模块

bMap = api.require('bMap');

// 获取当前经纬度

bMap.getLocation({

accuracy: '1000m', // 定位所在城市信息，不需要精准定位

autoStop: true, // 定位到经纬度后，立即自动停止定位服务

filter: 100000 // 基本无需处理位置更新的情况

}, function(ret, err) {

if (ret.status) {

// 根据经纬度获得城市名称

fnGetCityNameFromLocation(ret);

} else {

alert(err.code);

}

});

}

4.根据经纬度获得城市名称

function fnGetCityNameFromLocation(location\_) {

bMap.getNameFromCoords({

lon: location\_.lon,

lat: location\_.lat

}, function(ret, err) {

if (ret.status) {

// 判断当前所在的城市与上一次选择的城市是否一致

fnCheckCityChange(ret.city);

}

});

}

5.判断当前所在的城市与上一次选择的城市是否一致，不一致则打开城市选择Frame

function fnCheckCityChange(cityName\_) {

if (!city) {

return;

}

if (city.innerHTML != cityName\_) {

fnOpenCitySelectorFrame();

}

}

6.打开城市选择Frame

function fnOpenCitySelectorFrame() {

// 设置箭头状态

setArrowStateActive();

// 根据UI架构设计（ui-architecture.xmind），打开城市选择Frame

api.openFrame({

name: 'cityselector\_frame',

url: './cityselector\_frame.html',

bgColor: 'rgba(0, 0, 0, 0.8)',

rect: {

marginTop: headerH, // 采用margin布局

w: 'auto', // 自动填充所在Window的宽度

h: 'auto' // 自动填充所在Window的高度

},

bounces: true // 打开弹动

});

}

(6)商品展示页轮播图

引入了h5banner轮播图模块的css样式和js

1.更新商品Banner

function updateBanner() {

var banner = $api.byId('banner');

var imagesAndUrl = [{

imgSrc: '../image/1.webp',

linkHref: "#"

}, {

imgSrc: '../image/2.webp',

linkHref: '1'

}, {

imgSrc: '../image/3.webp',

linkHref: '#'

}, {

imgSrc: '../image/4.webp',

linkHref: '#'

}, {

imgSrc: '../image/5.webp',

linkHref: '#'

}];

imagesAndUrl.width = 20;

imagesAndUrl.height = 30;

var banner = new Banner(banner, imagesAndUrl, {

transitionTime: 800, // 动画过渡时间，默认为800ms

intervalTime: 1000, // 图片切换时间，默认为5s

});

}

(7)商品详情页轮播图

1.从数据库中查询对应商品的wareInfo中的图片

function getWareInfo(wareId\_) {

var params = {

fields: {},

where: {

id: wareId\_

},

skip: 0,

limit: 1,

include: ['wareInfoPointer']

}

params = $api.jsonToStr(params);

//计算加密后的appKey

var now = Date.now();

var appKey = SHA1(api.appId + "UZ" + "66DBD6E6-E309-5421-B020-A24C69A98BDB" + "UZ" + now) + "." + now;

api.ajax({

url: 'http://d.apicloud.com/mcm/api/ware?filter=' + params,

method: 'get',

headers: {

"X-APICloud-AppId": api.appId,

"X-APICloud-AppKey": appKey

}

}, function(ret, err) {

if (ret) {

api.hideProgress();

// 更新商品详情展示

fnUpdateWareInfo(ret[0]);

} else {

// alert(JSON.stringify(err));

}

});

}

2.更新商品详情展示

function fnUpdateWareInfo(ware\_) {

// 商品名称

var wareName = $api.byId('wareName');

wareName.innerHTML = ware\_.name;

// 商品描述

var description = $api.byId('description');

description.innerHTML = ware\_.description;

// 商品价格

var price = $api.byId('price');

price.innerHTML = '￥' + ware\_.price;

// 商品原价

var originPrice = $api.byId('originPrice');

originPrice.innerHTML = '原价：￥' + ware\_.originPrice;

// 商品规格

var unit = $api.byId('unit');

unit.innerHTML = '产地：' + ware\_.unit;

// 商品产地

var detail = $api.byId('detail');

detail.innerHTML = '产地：' + ware\_.wareInfo.detail;

3.更新轮播图中的图片

var paths = [];

for (var i = 0; i < 6; i++) {

if (ware\_.wareInfo['picture' + i]) {

paths.push(ware\_.wareInfo['picture' + i].url);

}

}

UIScrollPicture.reloadData({

data: {

paths: paths

}

});

}

4.导入轮播图模块，将查询到的轮播图显示出来

var UIScrollPicture;

function initUIScrollPicture() {

// 获取占位的div元素，并设置元素的宽高

var picture = $api.byId('picture');

picture.style.width = api.winWidth + 'px';

picture.style.height = api.winWidth + 'px';

var rect = $api.offset(picture);

// 引入UIScrollPicture模块，并指定位置和宽高

UIScrollPicture = api.require('UIScrollPicture');

UIScrollPicture.open({

rect: {

x: rect.l,

y: rect.t,

w: rect.w,

h: rect.h

},

data: {

paths: [

'widget://image/default\_rect.png',

'widget://image/default\_rect.png'

]

},

styles: {

caption: {

height: 35,

color: '#E0FFFF',

size: 13,

bgColor: '#696969',

position: 'bottom'

},

indicator: {

align: 'center',

color: '#FFFFFF',

activeColor: '#DA70D6'

}

},

placeholderImg: 'widget://image/default\_rect.png',

contentMode: 'scaleToFill',

interval: 5,

fixedOn: api.frameName,

loop: true,

fixed: false

}, function(ret, err) {

if (ret) {

//alert(JSON.stringify(ret));

} else {

alert(JSON.stringify(err));

}

});

}

(8)商品展示的上拉刷新

1.实现下拉刷新功能

function initRefresh() {

api.setRefreshHeaderInfo({

visible: true,

loadingImg: 'widget://image/refresh.png',

bgColor: '#fff',

textColor: '#e1017e',

showTime: true

}, function(ret, err) {

// 刷新商品列表

fnGetWareList(false);

});

}

(9)个人中心

1.打开我的订单Window

function fnOpenMyOrderWin() {

api.openWin({

name: 'myorder',

url: './myorder.html'

});

}

2.打开我的账户Window

function fnOpenAccountWin() {

api.openWin({

name: 'account',

url: './account.html'

});

}

3.打开我的优惠券Window

function fnOpenCouponWin() {

api.openWin({

name: 'coupon',

url: './coupon.html'

});

}

4.打开收货地址Window

function fnOpenAddressWin() {

api.openWin({

name: 'address',

url: './address.html'

});

}

(10)登录及注册

1.使用登录模块实现登录

function fnInitUIInput() {

// 用户名输入框模块

var username = $api.byId('username');

var usernameRect = $api.offset(username);

UIInput = api.require('UIInput');

UIInput.open({

rect: {

x: usernameRect.l,

y: usernameRect.t,

w: usernameRect.w,

h: usernameRect.h

},

styles: {

bgColor: '#fff',

size: 16,

color: '#000',

placeholder: {

color: '#ccc'

}

},

autoFocus: false,

maxRows: 1,

placeholder: '用户名',

keyboardType: 'default',

fixedOn: api.frameName

}, function(ret, err) {

if (ret) {

usernameUIInputId = ret.id;

if (ret && ret.eventType == "show") {

setTimeout(function() {

UIInput.popupKeyboard({

id: usernameUIInputId

});

}, 200);

} else if (ret.eventType == "change") {

UIInput.value({

id: usernameUIInputId

}, function(ret, err) {

if (ret) {

if (ret.status) {

usernameValue = ret.msg;

}

}

});

}

}

});

2.密码输入框模块

var password = $api.byId('password');

var passwordRect = $api.offset(password);

UIInput.open({

rect: {

x: passwordRect.l,

y: passwordRect.t,

w: passwordRect.w,

h: passwordRect.h

},

styles: {

bgColor: '#fff',

size: 16,

color: '#000',

placeholder: {

color: '#ccc'

}

},

autoFocus: false,

maxRows: 1,

placeholder: '密码',

keyboardType: 'default',

inputType: 'password',

fixedOn: api.frameName

}, function(ret, err) {

if (ret) {

passwordUIInputId = ret.id;

if (ret.eventType == "change") {

UIInput.value({

id: passwordUIInputId

}, function(ret, err) {

if (ret) {

if (ret.status) {

passwordValue = ret.msg;

}

}

});

}

}

});

}

3.登录,通过从数据库查询判断用户名密码是否正确

function fnLogin() {

if (!usernameValue) {

if (api.systemType == 'android') {

setTimeout(function() {

UIInput.popupKeyboard({

id: usernameUIInputId

});

}, 200);

} else {

UIInput.popupKeyboard({

id: usernameUIInputId

});

}

return;

}

if (!passwordValue) {

if (api.systemType == 'android') {

setTimeout(function() {

UIInput.popupKeyboard({

id: passwordUIInputId

});

}, 200);

} else {

UIInput.popupKeyboard({

id: passwordUIInputId

});

}

return;

}

//计算加密后的appKey

var now = Date.now();

var appKey = SHA1(api.appId + "UZ" + "66DBD6E6-E309-5421-B020-A24C69A98BDB" + "UZ" + now) + "." + now;

api.ajax({

url: 'http://d.apicloud.com/mcm/api/user/login',

method: 'post',

headers: {

"X-APICloud-AppId": api.appId,

"X-APICloud-AppKey": appKey

},

data: {

values: {

"username": usernameValue,

"password": passwordValue

}

}

}, function(ret, err) {

if (ret) {

// 保存用户信息

$api.setStorage('userInfo', ret);

api.alert({

title: '提示',

msg: '登录成功',

}, function(ret, err) {

if (ret) {

api.closeWin();

}

});

} else {

api.toast({

msg: '登录失败，用户名或密码错误',

duration: 2000,

location: 'middle'

});

}

});

}

4.被注册页面调用，用于从注册页面传递用户名

function fnSetUsername(username\_) {

usernameValue = username\_;

UIInput.value({

id: usernameUIInputId,

msg: username\_

}, function(ret, err) {});

}

(11)消息页面的打开展示

1.打开消息Window

function fnOpenMessageWin () {

api.openWin({

name: 'message',

url: './message.html'

});

}

2.页面内容

<body>

<section id="list" class="list">

<div class="message" tapmode="highlight">

<div class="type">优惠提醒</div>

<div class="content">【天天水果】尊敬的客户，您领到一张新的优惠卷。快去看看吧~！</div>

<div class="tag">

<div class="time">2016-02-26 12:02:08</div>

<div class="arrow">查看详情</div>

</div>

</div>

<div class="message" tapmode="highlight">

<div class="type">优惠提醒</div>

<div class="content">【天天水果】尊敬的客户，您领到一张新的优惠卷。快去看看吧~！</div>

<div class="tag">

<div class="time">2016-02-25 11:30:33</div>

<div class="arrow">查看详情</div>

</div>

</div>

</section>

</body>

(12)客服页面的打开展示

1.打开客服Window

function fnOpenCustomerServiceWin () {

api.openWin({

name: 'customerservice',

url: './customerservice.html'

});

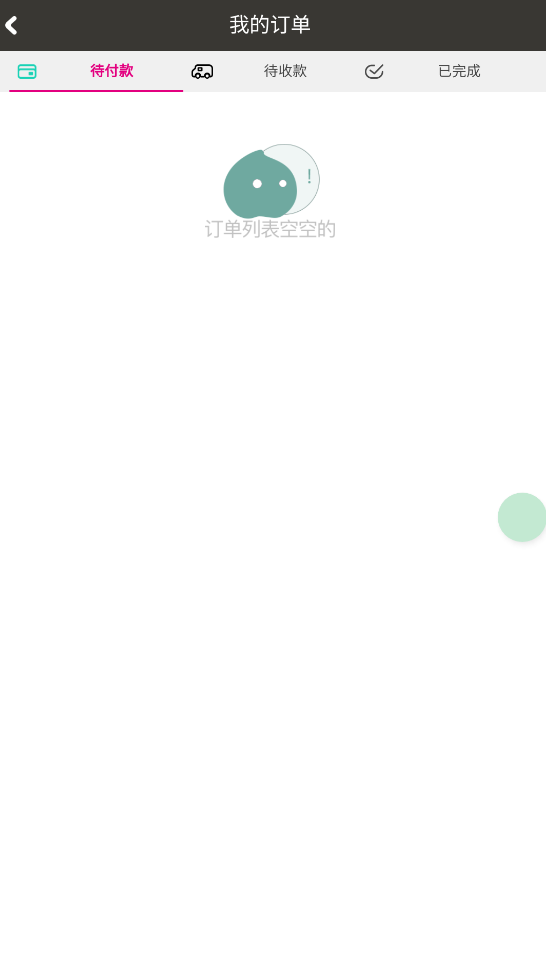
}

2.客服页面展示

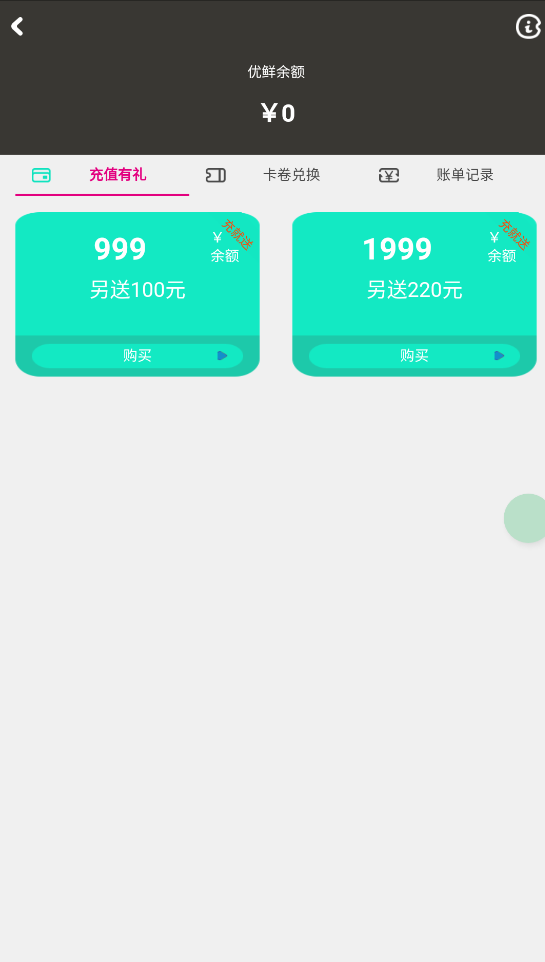


(13)个人中心的订单、账户、优惠卷、收货地址页面的展示

1. 订单



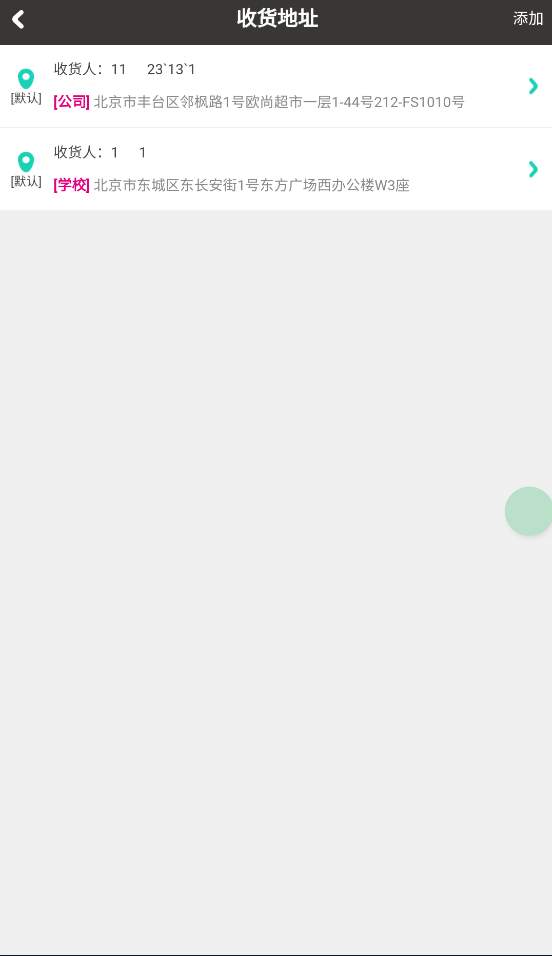
2.账户



3. 优惠卷



4.收货地址



(14)收货地址页面的地址的添加及删除

1.添加,导入UIActionSelector选择模块，设置地区json数据源

function fnOpenActionSelector() {

mobile.blur();

consignee.blur();

var UIActionSelector = api.require('UIActionSelector');

UIActionSelector.open({

datas: 'widget://res/city.json',

layout: {

row: 5,

col: 3,

height: 30,

size: 12,

sizeActive: 14,

rowSpacing: 5,

colSpacing: 10,

maskBg: 'rgba(0,0,0,0.2)',

bg: '#fff',

color: '#888',

colorActive: '#f00',

colorSelected: '#f00'

},

animation: true,

cancel: {

text: '取消',

size: 12,

w: 90,

h: 35,

bg: '#fff',

bgActive: '#ccc',

color: '#888',

colorActive: '#fff'

},

ok: {

text: '确定',

size: 12,

w: 90,

h: 35,

bg: '#fff',

bgActive: '#ccc',

color: '#888',

colorActive: '#fff'

},

title: {

text: '请选择',

size: 12,

h: 44,

bg: '#eee',

color: '#888'

},

fixedOn: api.frameName

}, function(ret, err) {

if (ret) {

var area = $api.byId('area');

area.innerHTML = ret.level1 + ret.level2 + (ret.level3 ? ret.level3 : '');

} else {

alert(JSON.stringify(err));

}

});

}

2.具体地址的选址实现，代码如下

var eList; // 列表容器对象

var tempFn; // 获取doT的模版对象

apiready = function() {

eList = $api.byId('list');

var template = $api.byId('template');

tempFn = doT.template(template.innerHTML);

};

//导入百度地图模块，实现对地址的输入查询选择

function fnSearchInCity(msg\_) {

if(!msg\_) {

$api.html(eList, '');

return;

}

var currentCity = $api.getStorage('currentCity');

var bMap = api.require('bMap');

bMap.searchInCity({

city: currentCity.name,

keyword: msg\_,

pageIndex: 0,

pageCapacity: 20

}, function(ret) {

if (ret.status) {

fnUpdateSearchList(ret);

}

});

}

//更新选择的地址

function fnUpdateSearchList(data\_) {

var html = tempFn(data\_);

$api.html(eList, html);

}

function fnSelectAddress(address\_) {

api.execScript({

name: 'setaddress',

frameName: 'setaddress\_frame',

script: 'setAddressDetail(\'' + address\_ + '\');'

});

api.closeWin();

}

3.删除

function fnSelectAddressType(index\_) {

var type = $api.byId('type');

var typeItems = $api.domAll(type, '.item');

addressInfo.type = index\_;

for (var i = 0; i < typeItems.length; i++) {

if (i == index\_) {

$api.addCls(typeItems[i], 'selected');

} else {

//点击删除，删除掉地址显示列表的该条地址

$api.removeCls(typeItems[i], 'selected');

}

}

}

(15)个人中心的头像的设置

1.获取个人信息

function getUserInfo() {

var userInfo = $api.getStorage('userInfo');

//计算加密后的appKey

var now = Date.now();

var appKey = SHA1(api.appId + "UZ" + "66DBD6E6-E309-5421-B020-A24C69A98BDB" + "UZ" + now) + "." + now;

api.ajax({

url: 'https://d.apicloud.com/mcm/api/user/' + userInfo.userId,

method: 'get',

headers: {

"X-APICloud-AppId": api.appId,

"X-APICloud-AppKey": appKey,

"Authorization": userInfo.id

}

}, function(ret, err) {

if (ret) {

fnUpdateLocalAvatar(ret);

} else {

alert(JSON.stringify(err));

}

});

}

2.更新头像显示

function fnUpdateLocalAvatar(data\_) {

if (!data\_.avatar) {

return;

}

3.缓存头像

var icon = $api.byId('icon');

api.imageCache({

url: data\_.avatar.url

}, function(ret, err) {

if (ret) {

icon.style.backgroundImage = 'url(' + ret.url + ')';

} else {

api.toast({

msg: '获取头像失败',

duration: 2000,

location: 'bottom'

});

}

});

}

4.选择头像

function fnSetAvatar() {

api.actionSheet({

title: '选择',

cancelTitle: '取消',

buttons: ['拍照', '相册']

}, function(ret, err) {

if (ret) {

var sourceTypes = [

'camera',

'album'

];

if (ret.buttonIndex == (sourceTypes.length + 1)) {

return;

}

api.getPicture({

sourceType: sourceTypes[ret.buttonIndex - 1],

allowEdit: true,

quality: 50, // 指定图片质量

targetWidth: 100, // 指定图片宽度

targetHeight: 100 // 指定图片宽度

}, function(ret, err) {

if (ret) {

fnUploadAtavar(ret.data);

}

});

}

});

}

5.上传头像文件

function fnUploadAtavar(avatarUrl\_) {

//计算加密后的appKey

var now = Date.now();

var appKey = SHA1(api.appId + "UZ" + "66DBD6E6-E309-5421-B020-A24C69A98BDB" + "UZ" + now) + "." + now;

api.ajax({

url: 'https://d.apicloud.com/mcm/api/file',

method: 'post',

headers: {

"X-APICloud-AppId": api.appId,

"X-APICloud-AppKey": appKey

},

data: {

values: {

filename: 'icon'

},

files: {

file: avatarUrl\_

}

}

}, function(ret, err) {

if (ret) {

fnUpdateUserAtavar(ret);

} else {

alert(JSON.stringify(err));

}

});

}

6.更新用户头像

function fnUpdateUserAtavar(avatar\_) {

var userInfo = $api.getStorage('userInfo');

//计算加密后的appKey

var now = Date.now();

var appKey = SHA1(api.appId + "UZ" + "66DBD6E6-E309-5421-B020-A24C69A98BDB" + "UZ" + now) + "." + now;

api.ajax({

url: 'https://d.apicloud.com/mcm/api/user/' + userInfo.userId,

method: 'put',

headers: {

"X-APICloud-AppId": api.appId,

"X-APICloud-AppKey": appKey,

"Authorization": userInfo.id

},

data: {

values: {

avatar: avatar\_

}

}

}, function(ret, err) {

if (ret) {

fnUpdateLocalAvatar(ret);

} else {

alert(JSON.stringify(err));

}

});

}

(16)app的设置、关于及缓存和退出登录

1. 缓存

//计算缓存大小

function getCacheSize() {

api.getCacheSize(function(ret) {

var size = ret.size;

var cacheSize = $api.byId('cacheSize');

size = parseInt((size / 1024 / 1024) \* 100) / 100;

cacheSize.innerHTML = size + ' M';

});

}

//清除缓存

function fnClearCache() {

api.clearCache(function() {

api.toast({

msg: '缓存清除完毕'

});

setTimeout(function() {

getCacheSize();

}, 300);

});

}

2.打开关于页面

function fnOpenAboutWin () {

api.openWin({

name: 'about',

url: './about.html'

});

}

3.退出登录

function fnLogout() {

api.confirm({

title: '提示',

msg: '是否退出登录',

buttons: ['确定', '取消']

}, function(ret, err) {

if (ret) {

if (1 == ret.buttonIndex) {

$api.rmStorage('userInfo');

api.closeToWin({

name: 'main'

});

}

}

});

}