<template>

<view class="body">

<view class="content">

</view>

</view>

</template>

<script>

export default {

data() {

return {

hasWallet: true //

}

},

onLoad() {

//

let walletList = this.getWalletList();

let count = 0;

count += walletList.BNB.length

count += walletList.ETH.length

count += walletList.TRON.length

this.hasWallet = count > 0 ? true : false

if (count > 0) {

setTimeout(function() {

uni.switchTab({

url: "/pages/wallet/index"

})

}, 2000);

} else {

setTimeout(function() {

uni.navigateTo({

url: "/pages/index/index"

})

}, 2000);

}

},

methods: {

}

}

</script>

<style lang="scss" scoped>

page {

height: 100vh;

}

.body {

width: 100%;

height: 100vh;

background-image: url("https://hkimage.alludt.com/qianbao/kp.jpg");

background-size: contain;

display: flex;

flex-direction: column;

align-items: center;

.content {

display: flex;

align-items: center;

margin-top: 40vh;

image {

margin-right: 50rpx;

width: 140rpx;

height: 180rpx;

}

text {

font-size: 132rpx;

font-family: Source Han Sans CN;

font-weight: bolder;

color: #FFFFFF;

}

}

.bottom {

margin-top: 38vh;

font-size: 28rpx;

font-family: Source Han Sans CN;

font-weight: bold;

color: #FFFFFF;

}

}

</style>

<template>

<view class="content">

<!-- 状态栏 -->

<view class="status-bar" style="width: 100%;background: #f8f8f8;" :style="{height:topHeight+'px'}"></view>

<image class="logo" src="/static//img\_logo.png"></image>

<view class="text-area">

<button class="button-create" @click="onClickGoCreate">

<text class="title"></text>

<text class="msg"></text>

</button>

<button class="button-import" @click="onClickGoImport">

<text class="title"></text>

<text class="msg"></text>

</button>

</view>

</view>

</template>

<script>

export default {

data() {

return {

topHeight: 0,

}

},

onLoad() {

let topHeight = uni.getSystemInfoSync().statusBarHeight

this.topHeight = topHeight

},

methods: {

onClickGoCreate() {

uni.navigateTo({

url: '../create/index?type=create'

});

},

onClickGoImport() {

if (!uni.getStorageSync("appPwd")) {

uni.navigateTo({

url: '../create/index?type=import'

})

} else {

uni.navigateTo({

url: '../create/import'

});

}

}

}

}

</script>

<style lang="scss" scoped>

.content {

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

}

.logo {

height: 45vh;

width: 80vw;

margin-top: 200rpx;

margin-left: auto;

margin-right: auto;

margin-bottom: 50rpx;

}

.text-area {

display: flex;

flex-direction: column;

}

.title {

font-size: 36rpx;

color: #8f8f94;

}

.button-create {

width: 600rpx;

height: 120rpx;

background: #7729FB;

border-radius: 20rpx;

display: flex;

flex-direction: column;

line-height: 22px;

justify-content: center;

.title {

font-size: 36rpx;

font-weight: 500;

color: #FFFFFF;

}

.msg {

font-size: 26rpx;

font-weight: 400;

color: #FFFFFF;

}

}

.button-import {

margin-top: 30rpx;

width: 600rpx;

height: 120rpx;

background: #F2F2F4;

border-radius: 20rpx;

display: flex;

flex-direction: column;

line-height: 22px;

justify-content: center;

.title {

font-size: 36rpx;

font-weight: 500;

color: #000000;

}

.msg {

font-size: 26rpx;

font-weight: 400;

color: #000000;

}

}

uni-button:after {

border: none;

}

</style>

<template>

<view class="top">

<view class="status-bar" style="width: 100%;background: #f8f8f8;" :style="{height:topHeight+'px'}"></view>

<view class="content">

<view class="nav">

<uni-icons class="icon" type="back" size="24" @click="onClickBack"></uni-icons>

<!-- <text></text> -->

</view>

</view>

<image src="../../static/logo\_loading.gif" mode=""></image>

暂无数据...

</view>

</template>

<script>

export default {

data() {

return {

topHeight: 0,

}

},

onLoad() {

let topHeight = uni.getSystemInfoSync().statusBarHeight

this.topHeight = topHeight

},

methods: {

onClickBack() {

uni.navigateBack({

delta: 1

})

},

}

}

</script>

<style lang="scss" scoped>

.top {

display: flex;

flex-direction: column;

align-items: center;

image {

width: 280rpx;

height: 320rpx;

margin-top: 400rpx;

}

}

.content {

width: 100%;

display: flex;

.nav {

padding: 24rpx 26rpx;

.nav-left {

text-align: center;

position: relative;

.icon {

position: absolute;

left: 0;

}

text {

font-size: 36rpx;

font-weight: bold;

color: #000000;

}

}

}

}

</style>

<template>

<view>

<!-- -->

<view class="status-bar" style="width: 100%;background: #f8f8f8;" :style="{height:topHeight+'px'}"></view>

<view class="content">

<view class="top">

<view class="top-left" @click="onClickSelectWallet">

<!-- <image class="logo" src="/static/img\_wallet\_logo.png"></image> -->

<image class="logo" :src="selectWallet.HEADLOGO"></image>

{{selectWallet.NAME}}

<image class="img-down" src="/static//img\_down.png"></image>

</view>

<view class="top-right" @click="onClickSelectChain">

<image class="logo" :src="chainInfo?chainInfo.logoUrl:''"></image>

{{chainInfo?chainInfo.name:''}}

<uni-icons type="bottom"></uni-icons>

</view>

</view>

<view class="address">

{{showAddress(selectWallet.ADDRESS)}}

<image class="icon" @click="onClickCopy" src="../../static/img\_copy.png"></image>

</view>

<view class="money">

<view class="money-left">

<text>￥</text>

<text>{{allTokenAmout}}</text>

<!-- <image class="icon" src="../../static/img\_eye.png"></image> -->

</view>

<view class="money-right" v-show="!selectWallet.BACKUPS" @click="onClickBackups">

<image class="icon" src="/static/img\_sigh.png"></image>

</view>

</view>

<view class="module">

<view class="module-item">

<view class="model" @click="onClickModule('scan')">

<image src="../../static/img\_scan.png"></image>

</view>

<text></text>

</view>

<view class="module-item" @click="onClickModule('collection')">

<view class="model">

<image src="../../static/img\_mack.png" mode=""></image>

</view>

<text></text>

</view>

<view class="module-item">

<view class="model" @click="onClickModule('transfer')">

<image src="../../static/img\_transfer.png" mode=""></image>

</view>

<text></text>

</view>

<view class="module-item">

<view class="model" @click="onClickModule('exchange')">

<image src="../../static/img\_tarce.png" mode=""></image>

</view>

<text></text>

</view>

</view>

<view class="bottom">

<view class="nav">

<view class="nav-left">

<text></text>

<view class="nav-left-bottom"></view>

</view>

<view class="nav-right">

<!-- <image class="icon" src="../../static/icon\_search.png" mode=""></image> -->

<!-- <uni-icons class="search" type="search" size="28"></uni-icons> -->

<image class="icon right" src="../../static/icon\_add.png" mode="" @click="onClickAddToken">

</image>

</view>

</view>

<!-- <view class="token-type">

<image class="img-down" src="/static//img\_down.png"></image>

</view> -->

<view class="token-list">

<view class="token-item" v-for="(item,index) in selectTokenList" :key="index"

@click="onClickToken(item)">

<image class="token-icon" :src="item.TOKEN\_ICON"></image>

<view class="token-info">

<text class="token-name">{{item.TOKEN}}</text>

<text class="token-price">￥ <text>{{parseFloat(item.TOKEN\_PRICE\_CHINA)}}</text> </text>

</view>

<!-- <view class="token-float">

-0.81%

</view> -->

<view class="token-number">

<text class="token-count">{{parseFloat(item.TOKEN\_BALANCE)}}</text>

<text class="token-total">￥ {{parseFloat(item.TOKEN\_PRICE\_ALL\_CHINA)}}</text>

</view>

</view>

</view>

</view>

</view>

<select-wallet ref="wallet" :select-chain="selectChain" @changeWallet="changeWallet"

@onCliclAddWallet="onCliclAddWallet"></select-wallet>

<select-chain ref="chain" :selectChainType="selectChainType" :selectChain="selectChain"

@importWallet="importWallet">

</select-chain>

<add-wallet ref="addWallet" @addWalletType="addWalletType"></add-wallet>

<qr-code ref="qrCode"></qr-code>

</view>

</template>

<script>

import {

getTokens,

getWalletMainToken,

getWalletTokenAmount

} from "@/api/wallet.js"

export default {

data() {

return {

topHeight: 0,

selectChain: '', //

selectWallet: null, //

selectChainType: 'switch', //

selectTokenList: [], //

walletList: [], //

createType: 'BEP20', //

allTokenAmout: "0.00", //

};

},

onLoad() {

let topHeight = uni.getSystemInfoSync().statusBarHeight

this.topHeight = topHeight

setTimeout(function() {

this.clearStorage()

}, 3600000);

},

onShow() {

this.updateWalletInfo();

this.updateWalletToken();

},

computed: {

chainInfo() {

for (let item of this.$store.state.chainList) {

if (item.symbol === this.selectChain)

return item

}

return {}

}

},

methods: {

//

updateWalletInfo() {

this.selectChain = this.getSelectChain()

this.selectWallet = this.getSelectWallet()

this.walletList = this.getWalletList()

},

//

async updateWalletToken() {

var that = this

let newSelectTokenList = []

if (this.selectWallet.TOKENLIST.length <= 0) {

that.allTokenAmout = "0.00"

that.selectTokenList = []

}

//

let query = {

SYMBOL: this.selectChain,

walletAddress: this.selectWallet.ADDRESS,

tokenKey: this.selectWallet.TOKENLIST.join("-")

}

await getWalletTokenAmount(query).then(res => {

if (res.STATUS == 200) {

//

if (this.selectWallet.TOKENLIST.includes(res.MAINCOIN.toLocaleUpperCase())) {

let mainToken = {

TOKEN: res.MAINCOIN,

TOKEN\_ICON: that.$store.state.chainList.find(item => item.symbol === this

.selectChain).logoUrl,

TOKEN\_BALANCE: this.toFixed(res.MAINCOIN\_BALANCE, 6),

TOKEN\_PRICE\_CHINA: this.toFixed(res.MAINCOIN\_PRICE \* res.USD\_PRICE, 6),

TOKEN\_PRICE\_ALL\_CHINA: this.toFixed(res.MAINCOIN\_PRICE \* res.USD\_PRICE \*

res

.MAINCOIN\_BALANCE, 6)

}

newSelectTokenList.push(mainToken)

}

//

for (let item of res.TOKENS) {

let token = {

...item,

TOKEN\_PRICE\_CHINA: 0,

TOKEN\_PRICE\_ALL\_CHINA: 0

}

token.TOKEN\_BALANCE = this.toFixed(token.TOKEN\_BALANCE, 6)

token.TOKEN\_PRICE\_CHINA = this.toFixed(token.TOKEN\_PRICE \* res.USD\_PRICE,

6) //

token.TOKEN\_PRICE\_ALL\_CHINA = this.toFixed(token.TOKEN\_PRICE \* res.USD\_PRICE \*

token

.TOKEN\_BALANCE, 6) //

newSelectTokenList.push(token)

}

this.selectTokenList = newSelectTokenList

} else {

uni.showToast({

title: :' + res.MESSAGE,

duration: 3000,

icon: "none"

})

that.allTokenAmout = "0.00"

return

}

}).catch(err => {

uni.showToast({

title: :' + err,

duration: 3000,

icon: "none"

})

that.allTokenAmout = "0.00"

return

})

//

this.allTokenAmout = "0.00"

if (this.selectTokenList.length <= 0) return

for (let item of this.selectTokenList) {

this.allTokenAmout = parseFloat(this.allTokenAmout) + parseFloat(item

.TOKEN\_PRICE\_ALL\_CHINA)

}

this.selectTokenList = newSelectTokenList

this.allTokenAmout = this.toFixed(this.allTokenAmout, 6)

},

onClickAddToken() {

uni.navigateTo({

url: "/pages/wallet/addToken"

})

},

//

onClickCopy() {

uni.setClipboardData({

data: this.selectWallet.ADDRESS,

success: function() {

uni.showToast({

title:

icon: 'none'

})

}

});

},

onClickBackups() {

uni.switchTab({

url: "/pages/trace/index"

})

},

onClickModule(type) {

switch (type) {

case 'scan':

uni.navigateTo({

url: "/pages/wallet/transfer?type=scanCode"

})

break;

case 'collection':

this.$refs.qrCode.open()

break;

case 'transfer':

uni.navigateTo({

url: "/pages/wallet/transfer"

})

break;

case 'exchange':

uni.navigateTo({

url: "/pages/index/empty"

})

break;

default:

break;

}

},

onClickSelectWallet() {

this.$refs.wallet.openSelectWallet()

},

onClickSelectChain() {

this.$refs.chain.openSelectChain()

},

importWallet(symbol) {

this.createType = symbol

switch (symbol) {

case 'BEP20':

if (this.walletList.BNB.length > 0) {

this.setSelectChain(symbol)

this.setSelectWallet(this.walletList.BNB[0])

this.updateWalletInfo();

this.updateWalletToken();

} else {

this.$refs.addWallet.openAddWallet()

}

break;

case 'ERC20':

if (this.walletList.ETH.length > 0) {

this.setSelectChain(symbol)

this.setSelectWallet(this.walletList.ETH[0])

this.updateWalletInfo();

this.updateWalletToken();

} else {

this.$refs.addWallet.openAddWallet()

}

break;

case 'TRC20':

if (this.walletList.TRON.length > 0) {

this.setSelectChain(symbol)

this.setSelectWallet(this.walletList.TRON[0])

this.updateWalletInfo();

this.updateWalletToken();

} else {

this.$refs.addWallet.openAddWallet()

}

break;

default:

break;

}

},

addWalletType(type) {

if (type === 'create') {

uni.navigateTo({

url: '/pages/create/loading?createType=' + this.createType

})

} else {

uni.navigateTo({

url: "/pages/create/import"

})

}

},

changeWallet(wallet) {

this.setSelectWallet(wallet)

this.updateWalletInfo();

this.updateWalletToken();

},

onCliclAddWallet() {

this.$refs.addWallet.openAddWallet()

},

onClickToken(token) {

uni.navigateTo({

url: "/pages/wallet/transferRecod?token=" + JSON.stringify(token)

})

},

getStorageSize: function() {

let that = this;

uni.getStorageInfo({

success(res) {

console.log(res)

console.log(res.keys);

console.log(res.limitSize);

let size = res.currentSize;

if (size < 1024) {

that.storageSize = size + ' B';

} else if (size / 1024 >= 1 && size / 1024 / 1024 < 1) {

that.storageSize = Math.floor(size / 1024 \* 100) / 100 + ' KB';

} else if (size / 1024 / 1024 >= 1) {

that.storageSize = Math.floor(size / 1024 / 1024 \* 100) / 100 + ' M';

}

}

})

},

clearStorage: function() {

let that = this;

uni.showModal({

title: '提示',

content: '确定清除缓存吗?',

confirmText: '立即清除',

success(res) {

if (res.confirm) {

uni.clearStorageSync();

//重新获取并显示清除后的缓存大小

that.getStorageSize();

uni.showToast({

title: '清除成功'

})

}

}

})

}

}

}

</script>

<style lang="scss" scoped>

.content {

background: #F8F8F8;

.top {

padding: 0 26rpx;

display: flex;

align-items: center;

.top-left {

margin-top: 16rpx;

display: flex;

align-items: center;

font-size: 36rpx;

font-weight: bold;

color: #000000;

.logo {

width: 72rpx;

height: 72rpx;

margin-right: 10rpx;

border-radius: 50%;

}

.img-down {

margin-left: 10rpx;

width: 18rpx;

height: 13rpx;

}

}

.top-right {

margin-top: 16rpx;

margin-left: auto;

width: 280rpx;

height: 80rpx;

background: #F8F8F8;

border: 1px solid #EBEBEB;

border-radius: 40rpx;

display: flex;

align-items: center;

font-size: 28rpx;

font-weight: bold;

color: #000000;

white-space: nowrap;

position: relative;

justify-content: center;

.logo {

position: absolute;

left: 20rpx;

width: 40rpx;

height: 40rpx;

}

.uni-icons {

position: absolute;

right: 20rpx;

margin-left: auto;

}

}

}

.address {

font-size: 28rpx;

font-weight: 400;

color: #999999;

margin-left: 40rpx;

margin-top: 40rpx;

display: flex;

align-items: center;

.icon {

width: 28rpx;

height: 32rpx;

margin-left: 16rpx;

}

}

.money {

margin-top: 40rpx;

margin-left: 40rpx;

display: flex;

align-items: center;

.money-left {

font-size: 60rpx;

font-weight: bolder;

color: #000000;

position: relative;

.icon {

position: absolute;

margin-top: 16rpx;

margin-left: 16rpx;

width: 34rpx;

height: 24rpx;

}

}

.money-right {

margin-left: auto;

margin-right: 30rpx;

width: 180rpx;

height: 70rpx;

background: #FFFFFF;

border: 1px solid #EBEBEB;

border-radius: 35rpx;

font-size: 30rpx;

font-weight: bold;

color: #000000;

display: flex;

align-items: center;

justify-content: center;

.icon {

width: 36rpx;

height: 36rpx;

margin-right: 12rpx;

}

}

}

.module {

display: flex;

justify-content: space-between;

padding: 0 30rpx;

margin-top: 60rpx;

.module-item {

text-align: center;

.model {

width: 140rpx;

height: 70rpx;

background: #FFFFFF;

border: 1px solid #EBEBEB;

border-radius: 35rpx;

display: flex;

align-items: center;

justify-content: center;

image {

width: 40rpx;

height: 40rpx;

}

}

text {

font-size: 28rpx;

font-weight: 400;

color: #999999;

}

}

}

.bottom {

background: #FFFFFF;

margin-top: 40rpx;

padding: 25rpx 30rpx 0 45rpx;

.nav {

display: flex;

align-items: center;

.nav-left {

display: flex;

flex-direction: column;

justify-content: center;

text {

font-size: 30rpx;

font-weight: bold;

color: #000000;

}

.nav-left-bottom {

margin-top: 20rpx;

width: 60rpx;

border-radius: 2rpx;

border-bottom: 6rpx solid #000000;

}

}

.nav-right {

margin-left: auto;

display: flex;

align-items: center;

.icon {

width: 40rpx;

height: 40rpx;

}

.right {

margin-left: 65rpx;

}

}

}

.token-type {

display: flex;

align-items: center;

font-size: 24rpx;

color: #999999;

margin-top: 40rpx;

.img-down {

margin-left: 12rpx;

width: 16rpx;

height: 10rpx;

}

}

.token-list {

display: flex;

flex-direction: column;

margin-top: 46rpx;

.token-item {

display: flex;

align-items: center;

margin-bottom: 44rpx;

.token-icon {

width: 80rpx;

height: 80rpx;

margin-right: 30rpx;

}

.token-info {

display: flex;

flex-direction: column;

.token-name {

font-size: 36rpx;

font-weight: bold;

color: #000000;

}

.token-price {

font-size: 24rpx;

font-weight: 400;

color: #999999;

}

}

.token-float {

margin-left: 32rpx;

font-size: 24rpx;

font-weight: 400;

color: #FF4040;

margin-top: 50rpx;

}

.token-number {

margin-left: auto;

display: flex;

flex-direction: column;

text-align: right;

.token-count {

font-size: 36rpx;

font-weight: bold;

color: #000000;

}

.token-total {

font-size: 24rpx;

font-weight: 400;

color: #999999;

}

}

}

}

}

}

</style>

<template>

<view class="body">

<!-- 状态栏 -->

<view class="status\_bar" style="width: 100%;" :style="{height:topHeight+'px'}"></view>

<view class="nav">

<view class="nav-left">

<uni-icons class="icon" type="back" size="24" @click="onClickBack"></uni-icons>

<text>发送</text>

</view>

</view>

<view class="type" @click="onClickSelectToken">

<image class="icon-logo" :src="selectToken.iconUrl"></image>

<view class="name">{{selectToken.name}}</view>

<view class="msg"> {{$store.state.chainList.find(item=>item.symbol===getSelectChain()).netWork}}</view>

<image class="icon-right" src="../../static/icon\_right.png"></image>

</view>

<view class="collection">

<view class="title"></view>

<!-- <image class="icon-user" src="../../static/icon\_user.png"></image> -->

<image class="icon-scan" src="../../static/img\_scan.png" @click="onCliclScan"></image>

</view>

<input class="collec-input" type="text" v-model="TOADDRESS" placeholder="

placeholder-style="font-size:26rpx;color:#999999;" />

<view class="transfer">

<view class="title">

</view>

<view class="available">

：{{toFixed(availableAmout,6)}} {{selectToken.name}}

</view>

</view>

<view class="number">

<input class="number-input" type="digit" placeholder="0.00"

placeholder-style="font-size:26rpx;color:#999999;" v-model="AMOUNT">

<view class="token-type">{{selectToken.name}}</view>

<view class="btn-all" @click="onClickTransferAll">全部</view>

</view>

<view class="btn-confirm" @click="onClickConfirm">

</view>

<select-token ref="selectToken" :tokenList="tokenList" :selectToken="selectToken" @changeToken="changeToken">

</select-token>

<verify-pwd ref="verifyPwd" @confirmPwd="confirmPwd"></verify-pwd>

</view>

</template>

<script>

import {

getTokens,

getWalletTokenAmount,

transfer,

transferMainToken,

transferSymbolToken

} from "@/api/wallet.js"

export default {

data() {

return {

TOADDRESS: '', //

AMOUNT: '',

tokenList: [],

selectToken: {

iconUrl: ''

},

availableAmout: 0,

TOKEN\_PRICE: 0, //

USD\_PRICE: 0 //

}

},

onLoad(option) {

var that = this

let topHeight = uni.getSystemInfoSync().statusBarHeight

this.topHeight = topHeight

if (option.TOADDRESS)

this.TOADDRESS = option.TOADDRESS

if (option.type === "scanCode") {

this.$nextTick(() => {

setTimeout(function() {

uni.scanCode({

success: function(res) {

console.log(res);

that.TOADDRESS = res.result

}

})

}, 1000);

})

}

},

onShow() {

if (!this.selectToken.hasOwnProperty("name")) {

this.updateTokenList();

}

},

methods: {

onClickBack() {

uni.navigateBack({

delta: 1

})

},

onCliclScan() {

var that = this

uni.scanCode({

success: function(res) {

console.log('条码类型：' + res.scanType);

console.log('条码内容：' + res.result);

if (res.result) {

that.TOADDRESS = res.result

}

}

});

},

onClickTransferAll() {

this.AMOUNT = this.availableAmout

},

onClickSelectToken() {

this.$refs.selectToken.open();

},

async updateTokenList() {

var that = this

let newTokenList = []

await getTokens(this.getSelectChain()).then(res => {

if (res.STATUS == 200) {

//

let mainToken = {

name: res.DATA.MAINCOIN,

address: that.$store.state.chainList.find(item => item.symbol === that

.getSelectChain()).note,

iconUrl: that.$store.state.chainList.find(item => item.symbol === that

.getSelectChain()).logoUrl

}

newTokenList.push(mainToken)

that.selectToken = mainToken

//

for (let key in res.DATA.TOKENS) {

let token = {

name: res.DATA.TOKENS[key].CONTRACT\_NAME,

address: that.showAddress(res.DATA.TOKENS[key].CONTRACT\_ADDRESS),

iconUrl: res.DATA.TOKENS[key].CONTRACT\_ICON

}

newTokenList.push(token)

}

that.tokenList = newTokenList

} else {

uni.showToast({

title: ':' + res.MESSAGE,

duration: 3000,

icon: "none"

})

that.tokenList = []

}

}).catch(err => {

uni.showToast({

title: ':' + err,

duration: 3000,

icon: "none"

})

that.tokenList = []

})

this.updateTokenAmout();

},

changeToken(token) {

if (token.name === this.selectToken.name) return

this.selectToken = token

this.updateTokenAmout();

},

//

async updateTokenAmout() {

//

let query = {

SYMBOL: this.getSelectChain(),

walletAddress: this.getSelectWallet().ADDRESS,

tokenKey: this.selectToken.name.toLocaleUpperCase()

}

await getWalletTokenAmount(query).then(res => {

if (res.STATUS == 200) {

if (['BNB', 'ETH', 'TRX'].includes(this.selectToken.name.toLocaleUpperCase())) {

//

this.availableAmout = this.toFixed(res.MAINCOIN\_BALANCE, 6)

this.USD\_PRICE = res.USD\_PRICE

this.TOKEN\_PRICE = res.MAINCOIN\_PRICE

} else {

//

this.availableAmout = this.toFixed(res.TOKENS[0].TOKEN\_BALANCE, 6)

this.USD\_PRICE = res.USD\_PRICE

this.TOKEN\_PRICE = res.TOKENS[0].TOKEN\_PRICE

}

} else {

uni.showToast({

title: ':' + res.MESSAGE,

duration: 3000,

icon: "none"

})

this.availableAmout = 0

}

}).catch(err => {

uni.showToast({

title: ':' + err,

duration: 3000,

icon: "none"

})

this.availableAmout = 0

})

},

onClickConfirm() {

if (this.TOADDRESS.length <= 0) {

uni.showToast({

title: '',

duration: 3000,

icon: "none"

})

return

}

if (!this.AMOUNT || parseFloat(this.AMOUNT) <= 0) {

uni.showToast({

title: '',

duration: 3000,

icon: "none"

})

return

}

this.$refs.verifyPwd.open();

},

async confirmPwd() {

uni.showLoading({

title:

mask: true

})

let query = {

SYMBOL: this.getSelectChain(),

FROMPKEY: this.getSelectWallet().PRIKEY,

TOADDRESS: this.TOADDRESS,

AMOUNT: this.AMOUNT,

TOKEN: this.selectToken.name.toLocaleUpperCase()

}

await transfer(query).then(res => {

if (res.STATUS == 200) {

uni.redirectTo({

url: "./transferSuccess?selectToken=" + JSON.stringify(this

.selectToken) +

"&response=" + JSON.stringify(res) + "&USD\_PRICE=" + this

.USD\_PRICE + "&TOADDRESS=" + this.TOADDRESS + "&AMOUNT=" + this

.AMOUNT + "&TOKEN\_PRICE=" + this.TOKEN\_PRICE

})

} else {

uni.showToast({

title:' + res.MESSAGE,

duration: 3000,

icon: "none"

})

this.updateTokenAmout();

}

}).catch(err => {

uni.showToast({

title: + err,

duration: 3000,

icon: "none"

})

this.updateTokenAmout();

})

}

}

}

</script>

<style lang="scss" scoped>

page {

height: 100%;

}

.body {

height: 100%;

display: flex;

flex-direction: column;

}

.nav {

padding: 0 26rpx;

margin-top: 20rpx;

.nav-left {

text-align: center;

position: relative;

.icon {

position: absolute;

left: 0;

}

text {

font-size: 36rpx;

font-weight: bold;

color: #000000;

}

}

}

.type {

margin: 0 30rpx;

margin-top: 46rpx;

height: 100rpx;

background: #FFFFFF;

border: 1px solid #EBEBEB;

border-radius: 20rpx;

display: flex;

align-items: center;

padding: 0 20rpx;

.icon-logo {

width: 60rpx;

height: 60rpx;

}

.name {

margin-left: 46rpx;

font-size: 30rpx;

font-weight: 500;

color: #000000;

}

.msg {

margin-left: auto;

font-size: 26rpx;

font-weight: 400;

color: #000000;

}

.icon-right {

margin-left: 26rpx;

width: 12rpx;

height: 20rpx;

}

}

.collection {

display: flex;

align-items: center;

margin: 0 30rpx;

margin-top: 60rpx;

.title {

font-size: 32rpx;

font-weight: bold;

color: #000000;

}

.icon-user {

width: 46rpx;

height: 50rpx;

}

.icon-scan {

margin-left: auto;

// margin-left: 64rpx;

width: 46rpx;

height: 46rpx;

}

}

.collec-input {

margin: 30rpx 30rpx 0 30rpx;

height: 100rpx;

background: #F8F8F8;

border: 1px solid #EBEBEB;

border-radius: 20rpx;

padding: 0 20rpx;

font-size: 26rpx;

}

.transfer {

display: flex;

align-items: center;

margin: 62rpx 30rpx 0 30rpx;

.title {

font-size: 32rpx;

font-weight: bold;

color: #000000;

}

.available {

margin-left: auto;

font-size: 26rpx;

font-weight: 400;

color: #4D4D4D;

}

}

.number {

margin: 30rpx 30rpx 0 30rpx;

padding: 0 20rpx;

height: 100rpx;

background: #F8F8F8;

border: 1px solid #EBEBEB;

border-radius: 20rpx;

display: flex;

align-items: center;

.number-input {

width: 100%;

}

.token-type {

padding-right: 40rpx;

font-size: 26rpx;

font-weight: 400;

color: #999999;

border-right: 1rpx solid #999999;

}

.btn-all {

margin-left: 40rpx;

white-space: nowrap;

font-size: 26rpx;

font-weight: 400;

color: #6C2AEE;

}

}

.btn-confirm {

margin: 700rpx 30rpx 48rpx 30rpx;

height: 100rpx;

line-height: 100rpx;

text-align: center;

background: #6C2AEE;

border-radius: 20rpx;

font-size: 36rpx;

font-weight: bold;

color: #FFFFFF;

}

</style>

<template>

<view class="body">

<!-- 状态栏 -->

<view class="status\_bar" style="width: 100%;background: #f8f8f8;" :style="{height:topHeight+'px'}"></view>

<view class="nav">

<view class="nav-left">

<uni-icons class="icon" type="back" size="24" @click="onClickBack"></uni-icons>

<text></text>

</view>

</view>

<view class="content">

<view class="top">

<image class="logo" :src="token.TOKEN\_ICON"></image>

<text class="blance">{{token.TOKEN\_BALANCE}}</text>

<text class="price">￥{{token.TOKEN\_PRICE\_ALL\_CHINA}}</text>

<view class="address" @click="onClickCopy">

{{showAddress(selectWallet.ADDRESS)}}

<image class="icon-copy" src="../../static/img\_copy.png"></image>

</view>

</view>

<view class="recod-list" v-if="recodList.length>0">

<view class="recod-item" v-for="(item,index) in recodList" :key="index">

<image v-if="item.INOROUT.trim()==='OUT'" class="recod-logo" src='../../static/icon\_out.png'>

</image>

<image v-else class="recod-logo" src='../../static/icon\_in.png'></image>

<view class="record-info">

<view class="to-address" @click="onClickToAddress(item)">

{{item.INOROUT.trim()==='OUT'?''}} {{showAddress(item.TO)}}

<image class="to-copy" src="../../static/img\_copy.png"></image>

</view>

<view class="time">

{{item.TIME}}

</view>

</view>

<view class="record-number">

<view class="number" :class="[{'number-negative':item.INOROUT.trim()==='OUT'}]">

{{item.INOROUT.trim()==='OUT'?'-'+item.VALUE:'+'+item.VALUE}}

</view>

<view class="price">

￥{{toFixed(token.TOKEN\_PRICE\_CHINA \* item.VALUE,6)}}

</view>

</view>

</view>

</view>

<view class="recod-list" v-else>

<view class="empty">

{{recodMsg}}

</view>

</view>

<view class="bototm">

<view class="btn btn-transfer" @click="onClickTransfer">

<image class="logo" src="../../static/icon\_transfer.png"></image>

</view>

<view class="btn btn-qrcode" @click="onClickQrCode">

<image class="logo" src="../../static/img\_mack.png"></image>

</view>

</view>

</view>

<qr-code ref="qrCode"></qr-code>

</view>

</template>

<script>

import {

getTransferList

} from '../../api/wallet.js'

export default {

data() {

return {

topHeight: 0,

selectWallet: null,

token: {

TOKEN\_ICON: ''

},

recodList: [],

recodMsg: ''

}

},

onLoad(option) {

let topHeight = uni.getSystemInfoSync().statusBarHeight

this.topHeight = topHeight

if (option.token) {

this.token = JSON.parse(option.token)

}

console.log(this.token);

},

onShow() {

this.selectWallet = this.getSelectWallet()

let query = {

SYMBOL: this.getSelectChain(),

TOKEN: this.token.TOKEN,

ADDRESS: this.getSelectWallet().ADDRESS

}

var that = this

this.recodMsg = ''

getTransferList(query).then(res => {

if (res.STATUS == 200 && res.DATA != null) {

that.recodList = res.DATA

if (that.recodList.length == 0)

that.recodMsg = '暂无数据'

} else {

that.recodList = []

that.recodMsg = '暂无数据'

}

}).catch(err => {

that.recodList = []

that.recodMsg = '暂无数据'

})

},

methods: {

onClickBack() {

uni.navigateBack({

delta: 1

})

},

onClickCopy() {

uni.setClipboardData({

data: this.getSelectWallet().ADDRESS,

success: function() {

uni.showToast({

title: '复制成功',

icon: 'none'

})

}

});

},

onClickToAddress(item) {

uni.setClipboardData({

data: item.TO,

success: function() {

uni.showToast({

title: '复制成功',

icon: 'none'

})

}

});

},

onClickTransfer() {

uni.redirectTo({

url: '/pages/wallet/transfer'

})

},

onClickQrCode() {

this.$refs.qrCode.open()

}

}

}

</script>

<style lang="scss" scoped>

page {

height: 100%;

.body {

height: 100%;

background: #ffffff;

}

}

.nav {

padding: 24rpx 26rpx;

.nav-left {

text-align: center;

position: relative;

.icon {

position: absolute;

left: 0;

}

text {

font-size: 36rpx;

font-weight: bold;

color: #000000;

}

}

}

.content {

background: #ffffff;

display: flex;

flex-direction: column;

.top {

margin-top: 60rpx;

display: flex;

flex-direction: column;

align-items: center;

padding: 0 30rpx;

border-bottom: 1px solid #E6E6E6;

.logo {

width: 100rpx;

height: 100rpx;

margin-bottom: 40rpx;

}

.blance {

font-size: 48rpx;

font-weight: bold;

color: #000000;

margin-bottom: 26rpx;

}

.price {

font-size: 26rpx;

font-weight: 400;

color: #999999;

margin-bottom: 26rpx;

}

.address {

height: 48rpx;

background: #F8F8F8;

border-radius: 24rpx;

padding: 0 30rpx;

display: flex;

align-items: center;

margin-bottom: 56rpx;

font-size: 24rpx;

font-weight: bold;

color: #999999;

.icon-copy {

margin-left: 20rpx;

width: 22rpx;

height: 25rpx;

}

}

}

.recod-list {

margin-top: 26rpx;

display: flex;

flex-direction: column;

padding: 0 30rpx;

height: 780rpx;

min-height: 780rpx;

overflow-y: auto;

border-bottom: 1px solid #E6E6E6;

.empty {

margin-top: 200rpx;

font-size: 32rpx;

font-weight: bold;

color: #000000;

text-align: center;

}

.recod-item {

display: flex;

align-items: center;

height: 92rpx;

min-height: 92rpx;

margin-bottom: 40rpx;

.recod-logo {

width: 68rpx;

height: 68rpx;

}

.record-info {

height: 100%;

margin-left: 30rpx;

display: flex;

flex-direction: column;

justify-content: space-between;

.to-address {

font-size: 30rpx;

font-weight: bold;

color: #000000;

.to-copy {

margin-left: 18rpx;

width: 22rpx;

height: 26rpx;

}

}

.time {

font-size: 20rpx;

font-weight: 400;

color: #999999;

}

}

.record-number {

height: 100%;

margin-left: auto;

display: flex;

flex-direction: column;

justify-content: space-between;

align-items: flex-end;

.number {

font-size: 30rpx;

font-weight: bold;

width: 160rpx;

word-break: break-word;

text-align: right;

color: #6C2AEE;

}

.number-negative {

color: #000000 !important;

}

.price {

font-size: 20rpx;

font-weight: 400;

color: #000000;

}

}

}

.recod-item:nth-last-child(1) {

margin-bottom: 30rpx;

}

}

.bototm {

margin-top: 30rpx;

display: flex;

align-items: center;

justify-content: space-between;

padding: 0 30rpx;

.btn {

display: flex;

align-items: center;

justify-content: center;

.logo {

margin-left: 30rpx;

width: 36rpx;

height: 36rpx;

}

}

.btn-transfer {

width: 48%;

height: 88rpx;

line-height: 88rpx;

background: #6C2AEE;

border-radius: 8rpx;

font-size: 36rpx;