import Vue from 'vue'

import App from './App'

Vue.config.productionTip = false

App.mpType = 'app'

// 此处为演示Vue.prototype使用，非uView的功能部分

Vue.prototype.vuePrototype = '枣红'

// 引入全局uView

import uView from 'uview-ui'

Vue.use(uView)

// 此处为演示vuex使用，非uView的功能部分

import store from '@/store'

let vuexStore = require('@/store/$u.mixin.js') // 引入uView提供的对vuex的简写法文件

Vue.mixin(vuexStore)

let mpShare = require('uview-ui/libs/mixin/mpShare.js'); // 引入uView对小程序分享的mixin封装

Vue.mixin(mpShare)

// 自定义

import { http } from '@/request/index.js' // 全局挂载引入，配置相关在该index.js文件里修改

import requestUrl from '@/request/interface.js' // 接口数据

import say from '@/utils/say.js' // 话语

import tools from '@/utils/tools.js'

Vue.prototype.$http = http

Vue.prototype.requestUrl = requestUrl

Vue.prototype.$say = say;

Vue.prototype.$tools = tools;

Vue.prototype.$onLaunched = new Promise(resolve => {

Vue.prototype.$isResolve = resolve

})

// #ifdef H5

import VConsole from 'vconsole'

if (process.env.NODE\_ENV === 'development') {

new VConsole()

}

// #endif

const app = new Vue({

store,

...App

})

app.$mount()

import { mapMutations, mapState } from 'vuex';

import { saveImagePopupDataMixin } from '@/mixins/save-image-popup.js';

import topicItem from '@/components/topic/item.vue'

import textCase from '@/components/text-case/text-case.vue'

import textResult from '@/components/text-case/result.vue'

import articleItem from '@/components/article/item.vue'

export default {

components: {

textCase,

textResult,

topicItem,

articleItem

},

mixins: [saveImagePopupDataMixin],

data () {

return {

keywords: '', // 搜索内容

pageIndex: 1, // 页码

currentTab: 0,

triggered: false, // 设置当前下拉刷新状态，true 表示下拉刷新已经被触发，false 表示下拉刷新未被触发

// tabGroup: [

// { name: '话术', pageIndex: 1, pageSize: 10, dataList: [], isGetData: false, noDataShow: false, noMoreDate: false },

// { name: '文案', pageIndex: 1, pageSize: 10, dataList: [], isGetData: false, noDataShow: false, noMoreDate: false },

// { name: '问答', pageIndex: 1, pageSize: 10, dataList: [], isGetData: false, noDataShow: false, noMoreDate: false },

// { name: '文章', pageIndex: 1, pageSize: 10, dataList: [], isGetData: false, noDataShow: false, noMoreDate: false },

// { name: '拓展', pageIndex: 1, pageSize: 10, dataList: [], isGetData: false, noDataShow: false, noMoreDate: false },

// { name: '趣味', pageIndex: 1, pageSize: 10, dataList: [], isGetData: false, noDataShow: false, noMoreDate: false },

// ],

tabGroup: [],

isDelShow: false,

loading: false, // 加载中

downloadToday: 0, // 今日下载量

womanSetting: false, // 女版设置

swiperList: [], // 轮播列表

adKeywordList: [], // 广告关键词列表

adKeyWords: "", // 缓存的广告关键词

preKeywords: "", // 前一个广告关键词

fromStorageKewWords: false, // 从缓存取关键广告词

isFirstNeedAdvertBanner: true, // 是否首次需要获取广告图片

}

},

computed: {

...mapState(['hasLogin', 'userInfo', 'globalConfig', 'advert'])

},

async onLoad (options) {

let that = this;

this.tabGroup = [

{ name: '话术', value: 'word', pageIndex: 1, pageSize: 10, dataList: [], isGetData: false, noDataShow: false, noMoreDate: false },

// { name: '文案', value: 'poetry', pageIndex: 1, pageSize: 10, dataList: [], isGetData: false, noDataShow: false, noMoreDate: false },

// { name: '文章', value: 'article,', pageIndex: 1, pageSize: 10, dataList: [], isGetData: false, noDataShow: false, noMoreDate: false },

// { name: '拓展', value: 'encyclopedia', pageIndex: 1, pageSize: 10, dataList: [], isGetData: false, noDataShow: false, noMoreDate: false },

// { name: '趣味', value: 'tool', pageIndex: 1, pageSize: 10, dataList: [], isGetData: false, noDataShow: false, noMoreDate: false },

]

// value: 'word,tool,encyclopedia,answer,poetry,article,share,bargin', label: '话术+工具趣味+拓展百科+作答+文案+文章+分享+砍价'

if (this.globalConfig.PLATE\_LIST.includes('poetry')) {

this.tabGroup.push({ name: '文案', value: 'poetry', pageIndex: 1, pageSize: 10, dataList: [], isGetData: false, noDataShow: false, noMoreDate: false },)

}

if (this.globalConfig.PLATE\_LIST.includes('article')) {

this.tabGroup.push({ name: '文章', value: 'article,', pageIndex: 1, pageSize: 10, dataList: [], isGetData: false, noDataShow: false, noMoreDate: false })

}

if (this.globalConfig.PLATE\_LIST.includes('encyclopedia')) {

this.tabGroup.push({ name: '拓展', value: 'encyclopedia', pageIndex: 1, pageSize: 10, dataList: [], isGetData: false, noDataShow: false, noMoreDate: false })

}

if (this.globalConfig.PLATE\_LIST.includes('tool')) {

this.tabGroup.push({ name: '趣味', value: 'tool', pageIndex: 1, pageSize: 10, dataList: [], isGetData: false, noDataShow: false, noMoreDate: false })

}

if (!this.hasLogin) {

uni.navigateTo({ url: '/pages/login/authorize' })

}

if (this.hasLogin) {

this.getUserInfo();

}

let womanSetting = uni.getStorageSync('womanSetting')

if (womanSetting && womanSetting === 'Y') {

this.womanSetting = true;

}

let adKeywordListStr = uni.getStorageSync('adKeywordList') || '';

if (adKeywordListStr) {

this.adKeywordList = JSON.parse(adKeywordListStr);

}

this.keywords = options.keywords && options.keywords !== "" ? options.keywords : '';

if (this.isFirstNeedAdvertBanner) {

this.getBannerList()

}

this.init();

},

onReady () {

},

onShow () {

/\* 建议放在onReady里执行，提前加载广告 \*/

let obj = {}

let advertId = null

// 插屏广告

// #ifdef MP-WEIXIN

obj = wx

advertId = this.advert.wxIndexInterstitialAd

// #endif

// #ifdef MP-QQ

obj = qq

advertId = this.advert.qqIndexInterstitialAd

// #endif

if (this.globalConfig.ADVERT\_SHOW === '0') {

// 显示广告

if (obj.createInterstitialAd && advertId) {

let interstitialAd = obj.createInterstitialAd({ adUnitId: advertId })

interstitialAd.onLoad(() => {

interstitialAd.show().catch((err) => {

console.error(err)

})

console.log('onLoad event emit')

})

interstitialAd.onError((err) => {

console.log('onError event emit', err)

})

interstitialAd.onClose((res) => {

console.log('onClose event emit', res)

})

}

}

},

watch: {

keywords (newVal) {

if (newVal) {

this.isDelShow = true; //HM修改 直接点击页面预设关键字样式异常，注销

} else {

this.isDelShow = false;

}

},

},

methods: {

...mapMutations(['login']),

// 初始化

init () {

let that = this

if (this.keywords === undefined || this.keywords === '') {

uni.showToast({ title: '请输入关键词', icon: 'none' })

return

}

if (this.keywords.length > 24) {

uni.showToast({ title: '最长24个字符，请重新输入', icon: 'none' })

return

}

this.tabGroup.forEach(v => {

v.isGetData = false

v.noDataShow = false

v.dataList = [] // 列表

v.pageIndex = 1

})

this.getDataFunc()

},

// 触底

onLower () {

if (this.tabGroup[this.currentTab].pageIndex

&& this.tabGroup[this.currentTab].totalPages

&& this.tabGroup[this.currentTab].pageIndex > this.tabGroup[this.currentTab].totalPages) {

return

}

this.getDataFunc();

},

// 获取数据

getDataFunc () {

let that = this;

if (this.keywords == '' || this.keywords.trim() == '') {

uni.showToast({ title: '请输入您要搜索的内容', icon: 'none' })

return

}

if (this.userInfo.energy <= 0) {

uni.showModal({

title: '搜索次数不足，无法查看',

content: that.$tools.isMpIos() ? 'ios用户请联系客服' : '升级会员，1000万+话术无限制查看拷贝',

showCancel: true,

confirmText: '做任务',

cancelText: '升级会员',

success: function (res) {

if (res.confirm) {

// #ifdef MP

uni.reLaunch({ url: '/pages/my/index' });

// #endif

// #ifdef APP-PLUS || H5

that.navTo('/pages/common/member');

// #endif

} else if (res.cancel) {

that.navTo('/pages/common/member');

}

}

});

return

} else {

this.consume()

}

if (this.tabGroup[this.currentTab].name === '话术') {

this.getLoveWord()

} else if (this.tabGroup[this.currentTab].name === '文案') {

this.getTextCaseList()

} else if (this.tabGroup[this.currentTab].name === '问答') {

this.getTopicList()

} else if (this.tabGroup[this.currentTab].name === '文章') {

this.getArticleList()

} else if (this.tabGroup[this.currentTab].name === '拓展') {

this.getEncyclopediaList()

} else if (this.tabGroup[this.currentTab].name === '趣味') {

this.getToolList()

}

},

// 消耗体力

consume () {

let that = this

let params = { type: 'word' }

if (this.globalConfig && this.globalConfig.IS\_CONSUME\_TIMES === '0') {

that.$http.post(that.requestUrl.api.consume, params).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

console.log("消耗体力")

} else {

console.log("未消耗体力")

}

})

}

},

// 获取话术列表

getLoveWord () {

let that = this

let params = {}

let tabItem = this.tabGroup[this.currentTab];

params = {

pageIndex: tabItem.pageIndex,

pageSize: tabItem.pageSize,

state: 0,

active: 0,

orderBy: 'download\_count DESC, length(`content`) ASC',

keywords: this.$tools.innerHTML(that.keywords.trim()),

}

if (that.globalConfig.IS\_COMMON\_DATA === '0') {

params.appId = that.requestUrl.api.commonAppId

} else if (that.globalConfig.IS\_COMMON\_DATA === '1') {

params.appId = that.requestUrl.api.appId

} else if (that.globalConfig.IS\_COMMON\_DATA === '2') {

params.appId = ''

}

if (!that.isFirstNeedAdvertBanner) {

that.getBannerList();

}

that.$http.post(that.requestUrl.api.getLoveWord, params).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

let resp = res.data.data.list

if (resp.length > 0) {

resp.forEach((v, i) => {

v.btnTitle = "复制"

if (that.userInfo && that.userInfo.memberGrade == 'basic' && i > 0) {

v.content = "————领取会员，查看此条————\n————领取会员，查看此条————"

v.btnTitle = "领取"

if (that.globalConfig && that.globalConfig.PAY\_MODEL && that.globalConfig.PAY\_MODEL === 'pay') {

v.messageType = "pay"

}

}

// #ifdef APP-PLUS || H5

if (that.userInfo && that.userInfo.memberGrade == 'basic' && tabItem.pageIndex !== 1) {

v.content = "————领取会员，查看此条————\n————领取会员，查看此条————"

v.btnTitle = "领取"

if (that.globalConfig && that.globalConfig.PAY\_MODEL && that.globalConfig.PAY\_MODEL === 'pay') {

v.messageType = "pay"

}

}

// #endif

if (v.appId && v.appId.toString() === that.requestUrl.api.wenanAppId) {

v.contentList = [v.content]

} else {

v.contentList = v.content.split("\n")

}

})

}

tabItem.pageIndex += 1

tabItem.dataList = tabItem.dataList.concat(resp)

tabItem.isGetData = true;

tabItem.totalPages = res.data.data.pages

tabItem.totalSize = res.data.data.total

tabItem.noDataShow = tabItem.dataList.length === 0 ? true : false

if (tabItem.totalSize === 10 || (tabItem.pageIndex > tabItem.totalPages && tabItem.totalPages > 1)) {

// 没有更多数据

tabItem.noMoreDate = true

}

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

}

}).catch(reason => {

console.log('Failed: ' + reason);

});

},

// 获取文案列表

getTextCaseList () {

let that = this

let tabItem = this.tabGroup[this.currentTab];

tabItem.noDataShow = false

let params = {}

params = {

pageIndex: tabItem.pageIndex,

pageSize: tabItem.pageSize,

state: 0,

active: 0,

collectUserId: '',

fIdName: tabItem.title,

orderBy: 'download\_count DESC',

keywords: that.keywords.trim()

}

if (this.hasLogin) {

params.collectUserId = this.userInfo.userId

}

if (this.globalConfig.IS\_COMMON\_DATA === '0') {

params.appId = that.requestUrl.api.wenanAppId

} else if (this.globalConfig.IS\_COMMON\_DATA === '1') {

params.appId = that.requestUrl.api.appId

}

that.$http.post(that.requestUrl.api.getLoveWord, params).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

let resp = res.data.data.list

if (resp.length > 0) {

resp.forEach(v => {

v.allShow = v.content.length <= that.wordLength ? true : false

if (that.hasLogin) {

v.userName = that.userInfo.name

v.avatar = that.userInfo.headIcon

} else {

v.userName = that.mpInfo.mpName

v.avatar = that.mpInfo.logo

}

})

}

tabItem.pageIndex += 1

tabItem.dataList = tabItem.dataList.concat(resp)

tabItem.isGetData = true;

tabItem.totalPages = res.data.data.pages

tabItem.totalSize = res.data.data.total

tabItem.noDataShow = tabItem.dataList.length === 0 ? true : false

if (tabItem.totalSize === 10 || (tabItem.pageIndex > tabItem.totalPages && tabItem.totalPages > 1)) {

// 没有更多数据

tabItem.noMoreDate = true

}

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

}

})

},

// 获取话题列表

getTopicList () {

let that = this

let params = {}

let tabItem = this.tabGroup[this.currentTab];

tabItem.noDataShow = false

params = {

pageIndex: tabItem.pageIndex,

pageSize: tabItem.pageSize,

state: 0,

orderBy: 'recommend',

appId: this.requestUrl.api.appId,

keywords: that.keywords.trim()

}

that.loaded = false;

that.$http.post(that.requestUrl.api.getTopicList, params).then(res => {

that.loaded = true;

if (res.statusCode === 200 && res.data.code === 200) {

let resp = res.data.data.list

if (res.data.data.list.length > 0) {

resp.forEach(v => {

var images = []

if (v.images && v.images.length > 0) {

images = v.images.split(",")

}

if (images.length > 1) {

images = images.slice(0, 1)

}

v.allShow = v.content.length <= that.wordLength ? true : false

v.images = images

})

}

tabItem.dataList = tabItem.dataList.concat(resp)

tabItem.pageIndex += 1

tabItem.isGetData = true

tabItem.totalPages = res.data.data.pages

tabItem.totalSize = res.data.data.total

tabItem.noDataShow = tabItem.dataList.length === 0 ? true : false

if (tabItem.totalSize === 10 || (tabItem.pageIndex > tabItem.totalPages && tabItem.totalPages > 1)) {

// 没有更多数据

tabItem.noMoreDate = true

}

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

}

}).catch(reason => {

console.log('Failed: ' + reason);

});

},

// 获取文章列表

getArticleList () {

let that = this

let params = {}

let tabItem = this.tabGroup[this.currentTab];

tabItem.noDataShow = false

params = {

pageIndex: tabItem.pageIndex,

pageSize: tabItem.pageSize,

state: 0,

fId: tabItem.id,

orderBy: 'is\_top ASC, create\_time DESC',

keywords: that.keywords.trim()

}

if (this.globalConfig.IS\_COMMON\_DATA === '0') {

params.appId = that.requestUrl.api.commonAppId

} else if (this.globalConfig.IS\_COMMON\_DATA === '1') {

params.appId = that.requestUrl.api.appId

}

that.$http.post(that.requestUrl.api.getArticleList, params).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

let resp = res.data && res.data.data && res.data.data.list ? res.data.data.list : []

if (res.data.data.list.length > 0) {

resp.forEach(v => {

var images = []

if (v.images.length > 0) {

images = v.images.split(";")

}

v.images = images

})

}

tabItem.pageIndex += 1

tabItem.dataList = tabItem.dataList.concat(resp)

tabItem.isGetData = true;

tabItem.totalPages = res.data.data.pages

tabItem.totalSize = res.data.data.total

tabItem.noDataShow = tabItem.dataList.length === 0 ? true : false

if (tabItem.totalSize === 10 || (tabItem.pageIndex > tabItem.totalPages && tabItem.totalPages > 1)) {

// 没有更多数据

tabItem.noMoreDate = true

}

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

}

}).catch(reason => {

console.log('Failed: ' + reason);

});

},

// 获取拓展列表

getEncyclopediaList () {

let that = this

let params = {}

let tabItem = this.tabGroup[this.currentTab];

tabItem.noDataShow = false

params = {

pageIndex: tabItem.pageIndex,

pageSize: tabItem.pageSize,

state: 0,

fId: tabItem.id,

orderBy: 'is\_top ASC, create\_time DESC',

keywords: that.keywords.trim()

}

if (this.globalConfig.IS\_COMMON\_DATA === '0') {

params.appId = that.requestUrl.api.wenanAppId

} else if (this.globalConfig.IS\_COMMON\_DATA === '1') {

params.appId = that.requestUrl.api.appId

}

that.$http.post(that.requestUrl.api.getArticleList, params).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

let resp = res.data && res.data.data && res.data.data.list ? res.data.data.list : []

tabItem.pageIndex += 1

tabItem.dataList = tabItem.dataList.concat(resp)

tabItem.isGetData = true;

tabItem.totalPages = res.data.data.pages

tabItem.totalSize = res.data.data.total

tabItem.noDataShow = tabItem.dataList.length === 0 ? true : false

if (tabItem.totalSize === 10 || (tabItem.pageIndex > tabItem.totalPages && tabItem.totalPages > 1)) {

// 没有更多数据

tabItem.noMoreDate = true

}

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

}

}).catch(reason => {

console.log('Failed: ' + reason);

});

},

// 获取工具列表

getToolList () {

let that = this

let params = {}

let tabItem = this.tabGroup[this.currentTab];

tabItem.noDataShow = false

params = {

pageIndex: tabItem.pageIndex,

pageSize: tabItem.pageSize,

state: 0,

fId: tabItem.id,

// orderBy: 'download\_count DESC, update\_time DESC',

orderBy: 'update\_time DESC',

keywords: that.keywords.trim()

}

if (this.globalConfig.IS\_COMMON\_DATA === '0') {

params.appId = that.requestUrl.api.toolAppId

} else if (this.globalConfig.IS\_COMMON\_DATA === '1') {

params.appId = that.requestUrl.api.appId

} else if (this.globalConfig.IS\_COMMON\_DATA === '2') {

params.appId = ''

}

that.$http.post(that.requestUrl.api.getLoveWord, params).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

let resp = res.data && res.data.data && res.data.data.list ? res.data.data.list : []

tabItem.pageIndex += 1

tabItem.dataList = tabItem.dataList.concat(resp)

tabItem.isGetData = true;

tabItem.totalPages = res.data.data.pages

tabItem.totalSize = res.data.data.total

tabItem.noDataShow = tabItem.dataList.length === 0 ? true : false

if (tabItem.totalSize === 10 || (tabItem.pageIndex > tabItem.totalPages && tabItem.totalPages > 1)) {

// 没有更多数据

tabItem.noMoreDate = true

}

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

}

}).catch(reason => {

console.log('Failed: ' + reason);

});

},

// 清除输入框

clear () {

//HM修改 收起键盘

// uni.hideKeyboard();

this.keywords = '';

},

// 复制文本

handleCopy (p, v) {

let that = this;

if (!this.hasLogin) {

uni.navigateTo({ url: '/pages/login/authorize' })

return

}

if (!this.handleAuth()) {

return;

}

if (this.globalConfig.DAY\_MAX\_COPY\_TIMES < this.userInfo.downloadToday && this.hasLogin && this.userInfo.memberState != 1) {

// 今日下载数大于最大限制 且 会员不在有效期 不给下载

uni.showModal({

title: '已达今日最大拷贝数',

content: "成为会员，立享1000万+高情商话术无限制拷贝",

cancelText: '开通会员',

confirmText: '任务领取',

success: function (res) {

if (res.confirm) {

// #ifdef MP

that.switchTo('/pages/my/index')

// #endif

// #ifdef APP-PLUS || H5

that.navTo('/pages/common/member')

// #endif

} else if (res.cancel) {

that.navTo('/pages/common/member')

// uni.reLaunch({ url: '/pages/user/zone' });

}

}

});

return

}

let firstTimes = this.globalConfig.DOWNLOAD\_NO\_ADVERT\_TIMES ? this.globalConfig.DOWNLOAD\_NO\_ADVERT\_TIMES : -1;

let secondTimes = -1;

let thirdTimes = -1;

// #ifdef MP

if (firstTimes && firstTimes !== -1) {

secondTimes = 10 + firstTimes;

thirdTimes = 25 + firstTimes;

}

// #endif

let flag = this.downloadToday == firstTimes || this.downloadToday == secondTimes || this.downloadToday == thirdTimes ? true : false;

if (this.globalConfig.ADVERT\_SHOW === '0' && this.downloadToday >= 0 && flag) {

this.showRewardAdvert(p, v);

return;

} else {

this.copyDo(p, v)

}

},

// 复制文本

copyDo (p, v) {

let that = this;

uni.setClipboardData({

data: p,

success: function () {

that.downloadToday = that.downloadToday + 1;

that.loveWordOperate(v.id)

that.consume()

}

});

},

// 复制文本

copyCustomer () {

let that = this;

uni.setClipboardData({

data: this.globalConfig && this.globalConfig.CONTACT\_WX && this.globalConfig.CONTACT\_WX.toString() ? this.globalConfig.CONTACT\_WX.toString() : 'glh\_426',

success: function () {

uni.showModal({

title: '温馨提示',

content: '客服微信号' + that.globalConfig.CONTACT\_WX.toString() + '已复制，去微信加好友开通吧',

showCancel: false,

confirmText: '确定',

success: function (res) {

}

});

}

});

},

// 文案操作 download collect unCollect

loveWordOperate (id) {

let that = this

if (this.loading) return

this.loading = true

let params = {}

params = {

wordId: id,

operate: 'download',

}

that.$http.post(that.requestUrl.api.loveWordOperate, params).then(res => {

that.loading = false;

if (res.statusCode === 200 && res.data.code === 200) {

uni.showToast({ title: '复制成功' })

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

}

}).catch(reason => {

that.loading = false;

console.log('Failed: ' + reason);

})

},

navTo (url) {

if (!this.hasLogin) {

uni.navigateTo({ url: '/pages/login/authorize' })

} else {

uni.navigateTo({ url: url })

}

},

switchTo (url) {

if (!this.hasLogin) {

uni.navigateTo({ url: '/pages/login/authorize' })

} else {

uni.switchTab({ url: url })

}

},

// 显示激励广告

showRewardAdvert (p, v) {

let obj = {};

let that = this;

let advertId = null

// #ifdef MP-WEIXIN

obj = wx

advertId = this.advert.wxActivityOrderRewardAd

// #endif

// #ifdef MP-QQ

obj = qq

advertId = this.advert.qqActivityOrderRewardAd

// #endif

if (this.globalConfig.ADVERT\_SHOW === '0') {

let videoAd = null

if (obj.createRewardedVideoAd && advertId) {

videoAd = obj.createRewardedVideoAd({

adUnitId: advertId

})

videoAd.onLoad(() => { })

videoAd.onError((err) => { })

videoAd.onClose((res) => {

if (res && res.isEnded || res === undefined) {

that.copyDo(p, v);

} else {

//不下发奖励

obj.showToast({ title: '看个视频，给小编鼓励一下', icon: 'none' })

}

})

}

// 用户触发广告后，显示激励视频广告

if (videoAd) {

videoAd.show().catch(() => {

// 失败重试

videoAd.load()

.then(() => videoAd.show())

.catch(err => {

that.loading = false

console.log('激励视频 广告显示失败')

})

})

}

}

},

// 读取用户信息

getUserInfo () {

let that = this;

that.$http.get(that.requestUrl.api.getUserInfo).then(res => {

if (res.statusCode === 200 && res.data && res.data.code === 200) {

that.login(res.data.data); // 存储登录信息

that.downloadToday = res.data.data.downloadToday;

}

})

},

// 未授权

handleAuth () {

let that = this;

let flag = true;

// #ifdef MP-WEIXIN || APP-PLUS || H5

if (this.hasLogin && this.userInfo.memberGrade == 'basic') {

uni.showModal({

title: '非会员或会员已过期',

content: that.$tools.isMpIos() ? 'ios用户请联系客服' : '成为会员，立享1000万+高情商话术全量查询拷贝',

cancelText: that.$tools.isMpIos() ? '我知道了' : '开通会员',

confirmText: '任务领取',

success: function (res) {

if (res.confirm) {

// #ifdef MP

that.switchTo('/pages/my/index')

// #endif

// #ifdef APP-PLUS || H5

that.navTo('/pages/common/member')

// #endif

} else if (res.cancel) {

that.navTo('/pages/common/member')

// uni.reLaunch({ url: '/pages/user/zone' });

}

}

});

flag = false

}

// #endif

return flag;

},

// 切换tab

changeTab (e) {

this.currentTab = e.index

},

// 切换swiper

changeSwiper (e) {

this.currentTab = e.detail.current;

if (!this.tabGroup[this.currentTab].isGetData) {

this.getDataFunc()

}

},

// 下拉刷新

onRefresh () {

if (this.triggered) return;

if (!this.triggered)

//界面下拉触发，triggered可能不是true，要设为true

this.triggered = true;

this.tabGroup[this.currentTab].dataList = []

this.tabGroup[this.currentTab].pageIndex = 1

this.tabGroup[this.currentTab].noDataShow = false

this.getDataFunc()

setTimeout(() => {

//触发onRestore，并关闭刷新图标

this.triggered = false;

this.loading = false;

}, 1000)

},

throttleSearch () {

uni.$u.throttle(this.init(), 500)

},

// 一键保存

onSave (isAuth, saveImageResult, hasImage) {

this.isAuth = isAuth

this.saveImageResult = saveImageResult

this.hasImage = hasImage

this.resultShow = true

},

// 轮播图

getBannerList () {

let that = this

let flag = false

if (that.isFirstNeedAdvertBanner) {

flag = true

that.adKeyWords = that.keywords

} else {

if (!that.fromStorageKewWords) {

// 没有从缓存拿过

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

flag = true;

that.adKeyWords = uni.getStorageSync('adKeyWords');

that.fromStorageKewWords = true;

}

}

if (that.adKeywordList.length) {

that.adKeywordList.forEach(v => {

if (that.keywords.indexOf(v) !== -1) {

// 命中

flag = true

that.adKeyWords = v

that.fromStorageKewWords = true;

}

})

}

}

if (flag) {

let params = { pageIndex: 1, pageSize: 10, state: 0, orderBy: 'banner\_index ASC', appId: that.requestUrl.api.appId }

if (!that.isFirstNeedAdvertBanner) {

params.adWords = that.adKeyWords

uni.setStorageSync('adKeyWords', that.adKeyWords); // 缓存起来

}

that.$http.post(that.requestUrl.api.getBannerList, params).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

that.isFirstNeedAdvertBanner = false

that.swiperList = res.data.data.list.filter(v => {

return v.imgType === 'resultFixed'

})

}

})

}

},

// 点击了自定义广告

handleBannerAdvert (e) {

this.$tools.handleSkip(this.hasLogin, e)

},

}

}

import { mapMutations, mapState } from 'vuex';

import mSearch from '@/components/mehaotian-search-revision/mehaotian-search-revision.vue';

import articleItem from '@/components/article/item.vue'

import textCase from '@/components/text-case/text-case.vue'

import FloatAd from './components/float.vue'

import PopupAd from './components/popup.vue'

export default {

components: {

//引用mSearch组件，如不需要删除即可

mSearch,

textCase,

articleItem,

FloatAd,

PopupAd

},

data () {

return {

total: 1588,

refresh: false, // 下拉刷新

showTips: true, // 提示

allLabelList: [], // 话术分类

articleList: [], // 文章

hotWordList: [], // 热门关键词

swiperList: [], // 轮播列表

popupList: [], // 弹窗列表

floatList: [], // 浮窗列表

encyclopediaList: [], // 百科列表

toolList: [], // 工具列表

indexFixedList: [], // 首页固定广告图

colorList: ['#FA5701', '#E6A23C', '#409EFF', '#FDC941', '#98A3F7', '#fd8fad', '#61D67B', '#F56C6C', '#8CC5FF', '#F7B17D', '#FD8D48', '#79BBFF'],

}

},

computed: {

...mapState(['hasLogin', 'userInfo', 'globalConfig', 'readTipsTime', 'mpInfo'])

},

// 下拉刷新

onPullDownRefresh () {

this.refresh = true

setTimeout(() => {

this.refresh = false

uni.stopPullDownRefresh()

this.init()

}, 500);

},

onLoad () {

this.total = this.$u.random(1000, 5000);

},

onShow () {

},

async onReady () {

await this.$onLaunched;

if (JSON.stringify(this.globalConfig) !== '{}') {

this.init();

} else {

setTimeout(() => {

this.init();

}, 1000)

}

},

methods: {

...mapMutations(['login', 'handleHotWord']),

init () {

this.getHotWord();

this.getLabelAll();

this.getToolList();

if (this.globalConfig && this.globalConfig.PLATE\_LIST && this.globalConfig.PLATE\_LIST.includes('article')) {

this.getArticleList();

}

if (this.globalConfig && this.globalConfig.PLATE\_LIST && this.globalConfig.PLATE\_LIST.includes('encyclopedia')) {

this.getEncyclopediaList();

}

this.getBannerList();

},

navTo (url) {

if (!this.hasLogin) {

uni.navigateTo({ url: '/pages/login/authorize' })

} else {

uni.navigateTo({ url: url })

}

},

switchTo (url) {

if (!this.hasLogin) {

uni.navigateTo({ url: '/pages/login/authorize' })

} else {

uni.switchTab({ url: url })

}

},

// 获取工具

getToolList () {

let that = this

let params = {

pageIndex: 1,

pageSize: 12,

state: 0,

orderBy: 'label\_index ASC, update\_time DESC'

}

if (this.globalConfig.IS\_COMMON\_DATA === '0') {

params.appId = that.requestUrl.api.toolAppId

} else if (this.globalConfig.IS\_COMMON\_DATA === '1') {

params.appId = that.requestUrl.api.appId

}

that.loading = true;

that.$http.post(that.requestUrl.api.getLabelTwo, params).then(res => {

that.loading = false;

if (res.statusCode === 200 && res.data.code === 200) {

that.toolList = res.data.data.list

}

}).catch(reason => {

that.loading = false;

console.log('Failed: ' + reason);

});

},

// 获取类目

getLabelAll () {

let that = this

let params = {}

if (this.globalConfig.IS\_COMMON\_DATA === '0') {

params.appId = that.requestUrl.api.commonAppId

} else if (this.globalConfig.IS\_COMMON\_DATA === '1') {

params.appId = that.requestUrl.api.appId

}

that.loading = true;

that.$http.post(that.requestUrl.api.getLabelAll, params).then(res => {

that.loading = false;

if (res.statusCode === 200 && res.data.code === 200) {

that.allLabelList = res.data.data

}

}).catch(reason => {

that.loading = false;

console.log('Failed: ' + reason);

});

},

// 获取文章列表

getArticleList () {

let that = this

let params = {}

params = { pageIndex: 1, pageSize: 4, state: 0, orderBy: 'is\_top ASC, create\_time DESC' }

if (this.globalConfig.IS\_COMMON\_DATA === '0') {

params.appId = that.requestUrl.api.commonAppId

} else if (this.globalConfig.IS\_COMMON\_DATA === '1') {

params.appId = that.requestUrl.api.appId

}

that.$http.post(that.requestUrl.api.getArticleList, params).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

let resp = res.data.data.list

if (res.data.data.list.length > 0) {

resp.forEach(v => {

var images = []

if (v.images.length > 0) {

images = v.images.split(";")

}

v.images = images

v.title = v.title.length > 20 ? v.title.slice(0, 20) + '...' : v.title

})

}

that.articleList = resp

}

})

},

// 获取百科列表

getEncyclopediaList () {

let that = this

let params = {}

params = { pageIndex: 1, pageSize: 4, state: 0, orderBy: 'is\_top ASC, create\_time DESC' }

if (this.globalConfig.IS\_COMMON\_DATA === '0') {

params.appId = that.requestUrl.api.wenanAppId

} else if (this.globalConfig.IS\_COMMON\_DATA === '1') {

params.appId = that.requestUrl.api.appId

}

that.$http.post(that.requestUrl.api.getArticleList, params).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

let resp = res.data.data.list

if (res.data.data.list.length > 0) {

resp.forEach(v => {

var images = []

if (v.images.length > 0) {

images = v.images.split(";")

}

v.images = images

v.title = v.title.length > 20 ? v.title.slice(0, 20) + '...' : v.title

})

}

that.encyclopediaList = resp

}

})

},

// 配置图

getBannerList () {

let that = this

let params = { pageIndex: 1, pageSize: 10, state: 0, orderBy: 'banner\_index ASC', appId: that.requestUrl.api.appId }

that.$http.post(that.requestUrl.api.getBannerList, params).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

let flag = true

that.swiperList = res.data.data.list.filter(v => {

return v.imgType === 'banner'

})

that.popupList = res.data.data.list.filter(v => {

return v.imgType === 'popup'

})

that.floatList = res.data.data.list.filter(v => {

return v.imgType === 'float'

})

that.indexFixedList = res.data.data.list.filter(v => {

return v.imgType === 'indexFixed'

})

}

})

},

// 热门关键词

getHotWord () {

let that = this

let params = {}

params = { pageIndex: 1, pageSize: 999, state: 0, orderBy: 'is\_top ASC, update\_time DESC' }

if (this.globalConfig.IS\_COMMON\_DATA === '0') {

params.appId = that.requestUrl.api.commonAppId

} else if (this.globalConfig.IS\_COMMON\_DATA === '1') {

params.appId = that.requestUrl.api.appId

}

that.$http.post(that.requestUrl.api.getHotWord, params).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

let list = [];

if (res.data.data.list.length > 0) {

res.data.data.list.forEach(v => {

list.push(v.title)

})

}

that.handleHotWord(list); // 存储热门关键词

that.hotWordList = this.$u.randomArray(list).splice(0, 12)

}

})

},

// 点击了自定义广告

handleBannerAdvert (e) {

this.$tools.handleSkip(this.hasLogin, e)

},

// 操作

handleTool (e, type) {

let gotoType = ''

let navType = ''

let url = ''

if (type === 'banner') {

e.navType = e.navType

e.url = e.url

e.gotoType = e.gotoType ? e.gotoType : 'navigateTo'

}

if (type === 'tool') {

e.navType = e.linkType

e.url = e.gotoUrl

e.gotoType = e.gotoType

}

this.$tools.handleSkip(this.hasLogin, e)

},

}

}

import { mapMutations, mapState } from 'vuex';

import taskList from '@/components/task/list.vue'

export default {

components: {

taskList

},

data () {

return {

loading: false,

menuTwoList: [

{ name: '订单列表', icon: 'file-text', url: '/pages/pay/orderRecord', },

{ name: '客服中心', icon: 'server-man', url: '/pages/common/customer' },

{ name: '更多好玩', icon: 'grid', url: '/pages/my/fun' },

{ name: '设置', icon: 'setting', url: '/pages/my/setting' },

],

memberLeftTime: 10, // 会员剩余天数提醒

memberLeftCurDay: 0, // 会员当前剩余天数

memberProtectTime: null, // 会员保护时间

taskConfigList: [], // 任务列表

dataInfo: {}, // 用户信息

activeCode: '',

}

},

computed: {

...mapState(['hasLogin', 'userInfo', 'globalConfig']),

},

onLoad () {

if (!this.hasLogin) {

this.getTaskConfig()

}

},

// 下拉刷新

onPullDownRefresh () {

setTimeout(() => {

if (this.hasLogin) {

this.getUserInfo()

}

uni.stopPullDownRefresh()

}, 500);

},

onShow () {

if (this.hasLogin) {

this.getUserInfo()

}

},

methods: {

...mapMutations(['login']),

// 读取用户信息

getUserInfo () {

let that = this;

that.$http.get(that.requestUrl.api.getUserInfo).then(res => {

if (res.statusCode === 200 && res.data && res.data.code === 200) {

that.login(res.data.data); // 存储登录信息

that.dataInfo = res.data.data;

that.memberProtectTime = that.dataInfo.memberProtectTime ? that.$tools.formatDateTime(that.dataInfo.memberProtectTime, 'yyyy-MM-dd') : '' // 会员保护时间

that.memberLeftCurDay = that.$tools.getDifferDay((new Date()).getTime(), that.dataInfo.memberProtectTime); // 会员剩余时间

that.getTaskConfig()

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

}

})

},

navTo (url) {

if (this.hasLogin) {

if (url === '/pages/my/contact' && this.globalConfig.BARGAIN\_SHOW !== '0') {

return;

}

uni.navigateTo({ url: url })

} else {

uni.navigateTo({ url: '/pages/login/authorize' })

}

},

// 图片预览

preview (urls) {

// 预览图片

uni.previewImage({

current: 0,

urls: urls,

longPressActions: {

itemList: ['发送给朋友', '保存图片', '收藏'],

success: function (data) {

console.log('选中了第' + (data.tapIndex + 1) + '个按钮,第' + (data.index + 1) + '张图片');

},

fail: function (err) {

console.log(err.errMsg);

}

}

});

},

// 获取任务配置

getTaskConfig () {

let that = this;

let param = { appId: that.requestUrl.api.appId, pageIndex: 1, pageSize: 5, state: 0, orderBy: 'label\_index DESC' }

that.$http.post(that.requestUrl.api.getTaskConfig, param).then(res => {

if (res.statusCode === 200 && res.data && res.data.code === 200 && res.data.data.list.length > 0) {

that.taskConfigList = res.data.data.list;

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

}

})

},

// 跳转

switchTo (url) {

if (this.hasLogin) {

uni.switchTab({ url: url })

} else {

uni.navigateTo({ url: '/pages/login/authorize' })

}

},

// 搜索次数提示

energyTips () {

let that = this;

uni.showModal({

title: '提示',

content: '搜索查询会消耗搜索次数，可通过完成任务或升级会员获取',

cancelText: '我知道了',

confirmText: '会员中心',

success (res) {

if (res.confirm) {

that.navTo('/pages/common/member');

} else if (res.cancel) {

console.log('用户点击取消');

}

}

})

},

// 会员到期

memberExpireTips () {

let that = this;

uni.showModal({

title: '提示',

content: that.memberProtectTime + '会员到期，提前续约会员享好礼',

cancelText: '我知道了',

confirmText: '会员中心',

success (res) {

if (res.confirm) {

that.navTo('/pages/common/member');

} else if (res.cancel) {

console.log('用户点击取消');

}

}

})

},

// 复制文本

copyDo (p) {

uni.setClipboardData({

data: p.toString(),

success: function () {

uni.showToast({ title: "复制成功", icon: 'none' })

}

});

},

// 激活码打开

inputDialogToggle () {

if (this.hasLogin) {

this.$refs.inputDialog.open()

} else {

uni.navigateTo({ url: '/pages/login/authorize' })

}

},

// 激活码关闭

dialogInputClose () {

this.$refs.inputDialog.close()

},

// 激活码确认

dialogInputConfirm (val) {

let that = this;

if (!val) {

uni.showToast({ title: "激活码不能为空", icon: 'none' })

return

}

if (val.length > 10) {

uni.showToast({ title: "激活码长度最大10个字符", icon: 'none' })

return

}

if (val.length != val.replace(/[^\w]/g, '').length) {

uni.showToast({ title: "激活码格式不正确", icon: 'none' })

return

}

that.activeCode = val

uni.$u.throttle(this.activeCodeUpdate, 1000)

},

activeCodeUpdate () {

let that = this;

let param = { activeCode: that.activeCode }

that.$http.post(that.requestUrl.api.activeCodeUpdate, param).then(res => {

if (res.statusCode === 200 && res.data && res.data.code === 200) {

that.activeCode = ""

that.$refs.inputDialog.close()

uni.showToast({ title: "恭喜您成功升级会员", icon: 'none' })

setTimeout(() => {

that.getUserInfo()

}, 1000)

} else {

that.activeCode = ""

uni.showToast({ title: res.data.data, icon: 'none' })

}

})

},

}

}

import { mapMutations, mapState } from 'vuex';

export default {

data () {

return {

copyChecked: false,

womanSetting: false,

}

},

onLoad () {

this.copyChecked = uni.getStorageSync('agreeClipboardData') && uni.getStorageSync('agreeClipboardData') === 'Y' ? true : false;

this.womanSetting = uni.getStorageSync('womanSetting') && uni.getStorageSync('womanSetting') === 'Y' ? true : false;

},

computed: {

...mapState(['hasLogin', 'userInfo', 'globalConfig']),

},

methods: {

...mapMutations(['logout']),

// 剪切板

changeCopy (e) {

let that = this;

that.copyChecked = e.detail.value;

uni.setStorage({

key: 'agreeClipboardData',

data: that.copyChecked ? 'Y' : 'N',

success () {

uni.showToast({ title: that.copyChecked ? '已开启授权剪切板' : '已关闭授权剪切板', icon: 'none' })

}

});

},

// 剪切板

changeSex (e) {

let that = this;

that.womanSetting = e.detail.value;

uni.setStorage({

key: 'womanSetting',

data: that.womanSetting ? 'Y' : 'N',

success () {

uni.showToast({ title: '设置成功', icon: 'none' })

}

});

},

navTo (url) {

if (this.hasLogin) {

uni.navigateTo({ url: url })

} else {

uni.navigateTo({ url: '/pages/login/authorize' })

}

},

// 检查更新

checkUpgrade () {

let cr = 0

if (uni.getSystemInfoSync().platform === 'android') {

cr = this.$tools.compareVersion(this.requestUrl.api.version, this.globalConfig.ANDROID\_VERSION)

} else if (uni.getSystemInfoSync().platform === 'ios') {

cr = this.$tools.compareVersion(this.requestUrl.api.version, this.globalConfig.IOS\_VERSION)

}

if (cr < 0) {

this.handleUpdate()

} else {

uni.showToast({ title: "当前已是最新版本", icon: 'none' });

}

},

handleUpdate () {

uni.showModal({

title: '温馨提示',

content: '有最新版本，是否升级',

confirmText: '确定',

cancelText: '取消',

success: function (res) {

if (res.confirm) {

let platform = uni.getSystemInfoSync().platform

if (platform === 'android') {

uni.showLoading({

title: '下载更新中....'

});

uni.downloadFile({

url: this.globalConfig.ANDROID\_ADDRESS,

success: (downloadResult) => {

uni.hideLoading();

if (downloadResult.statusCode === 200) {

plus.runtime.install(downloadResult.tempFilePath, {

force: false

}, function () {

plus.runtime.restart();

}, function (e) {

console.error('install fail...');

});

}

},

fail () {

uni.showToast({ title: '更新失败，请重试', icon: 'none' })

uni.hideLoading();

}

});

} else if (platform === 'ios') {

//在App Store Connect中的App Store下的app信息，可找到appleId

let appleId = 1586132017

plus.runtime.launchApplication({

action: `itms-apps://itunes.apple.com/cn/app/id${appleId}?mt=8`

}, function (e) {

console.log('Open system default browser failed: ' + e.message);

});

}

} else if (res.cancel) {

}

}

});

},

// 服务协议

serviceAgreement () {

let url = this.requestUrl.api.serviceAgreement

uni.navigateTo({ url: '/pages/webview/webview?url=' + encodeURIComponent(JSON.stringify(url)) })

},

// 隐私政策

privacyPolicy () {

let url = this.requestUrl.api.privacyPolicy

uni.navigateTo({ url: '/pages/webview/webview?url=' + encodeURIComponent(JSON.stringify(url)) })

},

// 退出登录

userLoginOut () {

let that = this;

uni.showModal({

title: '提示',

content: '退出后不会删除任何历史数据，下次登录依然可以使用账号。',

showCancel: true,

cancelText: '取消',

confirmText: '退出登录',

success: function (res) {

if (res.confirm) {

that.$http.post(that.requestUrl.api.userLoginOut, null).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

that.logout();

uni.showToast({ title: '已退出登录' })

// #ifdef APP-PLUS || H5

uni.reLaunch({

url: '/pages/login/authorize'

})

// #endif

// #ifdef MP

uni.reLaunch({

url: '/pages/index/index'

})

// #endif

} else {

uni.showToast({ title: res.data.data, icon: 'none' });

}

}).catch(err => {

uni.showToast({

title: err.errMsg,

icon: 'none'

});

});

}

}

});

}

}

}

import awardPopup from '@/components/common/award-popup.vue'

import {

mapMutations,

mapState

} from 'vuex';

export default {

components: {

awardPopup

},

data () {

return {

openid: '',

inviteId: '', // 邀请者id

maskShow: false,

dataSession: '', // 是否已经注册过的凭证，有值代表已注册

popupShow: false, // 奖励弹窗

}

},

// 监听页面卸载

onUnload () { },

onLoad () {

// #ifdef APP-PLUS || H5

uni.redirectTo({

url: '/pages/login/appLogin/loginByTel'

});

return;

// #endif

this.inviteId = uni.getStorageSync('inviteId') ? uni.getStorageSync('inviteId') : '';

// #ifdef MP-QQ || MP-WEIXIN || MP-TOUTIAO || MP-ALIPAY || MP-BAIDU

this.getCode();

// #endif

},

onHide () { },

computed: {

...mapState(['mpInfo', 'hasLogin']),

},

methods: {

...mapMutations(['login']),

// 获得临时登录凭证 code

getCode () {

let that = this;

that.maskShow = true

uni.login({

provider: uni.getStorageSync('PLATFORM') === 'MP-WEIXIN' ? 'weixin' :

uni.getStorageSync('PLATFORM') === 'MP-QQ' ? 'qq' :

uni.getStorageSync('PLATFORM') === 'MP-TOUTIAO' ? 'toutiao' :

uni.getStorageSync('PLATFORM') === 'MP-BAIDU' ? 'baidu' :

uni.getStorageSync('PLATFORM') === 'MP-ALIPAY' ? 'alipay' : '',

success: function (loginRes) {

// 登录凭证校验。通过 wx.login 接口获得临时登录凭证 code 后传到开发者服务器调用此接口完成登录流程。更多使用方法详见 小程序登录。

let params = {

"code": loginRes.code //类型：String 必有字段 备注：快捷登录返回的openid //类型：String 必有字段 备注：创建人

}

let url = uni.getStorageSync('PLATFORM') === 'MP-WEIXIN' ? that.requestUrl.api.wxLogin :

uni.getStorageSync('PLATFORM') === 'MP-QQ' ? that.requestUrl.api.qqLogin :

uni.getStorageSync('PLATFORM') === 'MP-TOUTIAO' ? that.requestUrl.api.toutiaoLogin :

uni.getStorageSync('PLATFORM') === 'MP-BAIDU' ? that.requestUrl.api.baiduLogin :

uni.getStorageSync('PLATFORM') === 'MP-ALIPAY' ? that.requestUrl.api.alipayLogin :

that.requestUrl.api.wxLogin;

that.$http.post(url, params).then(res => {

if (res.statusCode === 200 && res.data && res.data.code === 200) {

console.log('获得临时登录凭证成功')

if (res.data.data.session) {

uni.setStorage({

key: 'authorization',

data: res.data.data.session,

success: function () {

that.maskShow = false

that.dataSession = res.data.data.session

console.log('保存临时登录凭证成功')

}

});

} else {

that.maskShow = false

that.openid = res.data.data.wx\_openid || res.data.data.qq\_openid || res.data.data.toutiao\_openid || res.data.data.baidu\_openid;

}

} else {

console.log('获得临时登录凭证失败')

uni.showToast({ title: res.data.data, icon: 'none' })

that.maskShow = false

}

}).catch(reason => {

console.log('获得临时登录凭证失败')

that.maskShow = false;

console.log('Failed: ' + reason);

});

}

});

},

// 授权获取信息登录注册

infoLoginBaidu (e) {

let that = this;

let params = {

"encrypted\_data": e.detail.encryptedData, //类型：String 必有字段 备注：无

"iv": e.detail.iv, //类型：String 必有字段 备注：无

"openid": e.detail.cloudId,//类型：String 必有字段 备注：快捷登录返回的openid //类型：String 必有字段 备注：创建人

"appId": that.requestUrl.api.appId, // 当前应用appId

"avatarUrl": e.detail.userInfo.avatarUrl,

"gender": e.detail.userInfo.gender,

"nickName": e.detail.userInfo.nickName,

}

if (that.inviteId) params.inviteId = that.inviteId

that.$http.post(that.requestUrl.api.baiduInfoLogin, params).then(res => {

if (res.statusCode === 200 && res.data && res.data.code === 200) {

uni.setStorage({

key: 'authorization',

data: res.data.data,

success: function () {

that.getUserInfo();

}

});

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

that.maskShow = false

}

}).catch(reason => {

that.maskShow = false;

console.log('Failed: ' + reason);

});

},

// 授权获取信息登录注册

infoLogin () {

let that = this;

that.maskShow = true;

let platForm = uni.getStorageSync('PLATFORM');

if (that.dataSession && that.dataSession !== '') {

that.getUserInfo()

} else {

if (platForm === 'MP-WEIXIN') {

uni.getUserProfile({

desc: '用于完善会员资料',

success: (infoRes) => {

let params = {}

params = {

"encrypted\_data": infoRes.encryptedData, //类型：String 必有字段 备注：无

"iv": infoRes.iv, //类型：String 必有字段 备注：无

"openid": that.openid,//类型：String 必有字段 备注：快捷登录返回的openid //类型：String 必有字段 备注：创建人

"appId": that.requestUrl.api.appId, // 当前应用appId

}

if (that.inviteId) params.inviteId = that.inviteId

// params.inviteId = '100001'

let url = uni.getStorageSync('PLATFORM') === 'MP-WEIXIN' ? that.requestUrl.api.wxInfoLogin : uni.getStorageSync('PLATFORM') === 'MP-QQ' ? that.requestUrl.api.qqInfoLogin : that.requestUrl.api.wxInfoLogin;

that.$http.post(url, params).then(res => {

if (res.statusCode === 200 && res.data && res.data.code === 200) {

uni.setStorage({

key: 'authorization',

data: res.data.data,

success: function () {

that.getUserInfo();

}

});

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

that.maskShow = false

}

})

},

fail: res => {

console.log("uni.getUserProfile fail res:" + JSON.stringify(res));

that.maskShow = false

//拒绝授权

uni.showToast({ title: '您拒绝了请求', icon: 'none' })

return;

}

});

} else if (platForm === 'MP-QQ' || platForm === 'MP-TOUTIAO' || platForm === 'MP-ALIPAY' || platForm === 'MP-BAIDU') {

uni.getUserInfo({

provider: uni.getStorageSync('PLATFORM') === 'MP-WEIXIN' ? 'weixin' :

uni.getStorageSync('PLATFORM') === 'MP-QQ' ? 'qq' :

uni.getStorageSync('PLATFORM') === 'MP-TOUTIAO' ? 'toutiao' :

uni.getStorageSync('PLATFORM') === 'MP-ALIPAY' ? 'alipay' :

uni.getStorageSync('PLATFORM') === 'MP-BAIDU' ? 'baidu' : 'weixin',

success: function (infoRes) {

console.log("infoRes:")

console.log(infoRes)

let params = {}

params = {

"encrypted\_data": infoRes.userInfo.encryptedData, //类型：String 必有字段 备注：无

"iv": infoRes.userInfo.iv, //类型：String 必有字段 备注：无

"openid": that.openid,//类型：String 必有字段 备注：快捷登录返回的openid //类型：String 必有字段 备注：创建人

"appId": that.requestUrl.api.appId, // 当前应用appId

"avatarUrl": infoRes.userInfo.avatarUrl,

"city": infoRes.userInfo.city,

"country": infoRes.userInfo.country,

"gender": infoRes.userInfo.gender,

"language": infoRes.userInfo.language,

"nickName": infoRes.userInfo.nickName,

"province": infoRes.userInfo.province,

}

// #ifdef MP-QQ

// qq小程序返回结构有变化

params.iv = infoRes.iv ? infoRes.iv : ''

params.encrypted\_data = infoRes.encryptedData ? infoRes.encryptedData : ''

// #endif

if (that.inviteId) params.inviteId = that.inviteId

let url = uni.getStorageSync('PLATFORM') === 'MP-WEIXIN' ? that.requestUrl.api.wxInfoLogin :

uni.getStorageSync('PLATFORM') === 'MP-QQ' ? that.requestUrl.api.qqInfoLogin :

uni.getStorageSync('PLATFORM') === 'MP-TOUTIAO' ? that.requestUrl.api.toutiaoInfoLogin :

uni.getStorageSync('PLATFORM') === 'MP-ALIPAY' ? that.requestUrl.api.alipayInfoLogin :

uni.getStorageSync('PLATFORM') === 'MP-BAIDU' ? that.requestUrl.api.baiduInfoLogin :

that.requestUrl.api.wxInfoLogin;

that.$http.post(url, params).then(res => {

if (res.statusCode === 200 && res.data && res.data.code === 200) {

uni.setStorage({

key: 'authorization',

data: res.data.data,

success: function () {

that.getUserInfo();

}

});

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

that.maskShow = false

}

}).catch(reason => {

that.maskShow = false;

console.log('Failed: ' + reason);

});

},

fail: res => {

console.log("uni.getUserInfo fail res:" + JSON.stringify(res));

that.maskShow = false

//拒绝授权

uni.showToast({ title: '您拒绝了请求', icon: 'none' })

return;

}

});

}

}

},

// 获取用户信息

getUserInfo () {

let that = this

that.$http.get(that.requestUrl.api.getUserInfo).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

let resp = res.data.data;

that.login(resp); // 存储登录信息

var pages = getCurrentPages(); // 获取当前页面栈

var prePage = pages[pages.length - 2]; // 上一个页面

prePage.data.init = true // A 页面 init方法 为true

if (!that.dataSession && that.globalConfig && that.globalConfig.MEMBER\_PROTECT\_DAYS > 0) {

// 是否已经注册过的凭证，有值代表已注册

that.maskShow = false

that.popupShow = true

} else {

setTimeout(() => {

uni.showToast({ title: '登录成功' })

that.maskShow = false

uni.navigateBack({ delta: 1 }); // 返回上一页

}, 0)

}

} else {

that.maskShow = false

uni.showToast({ title: res.data.data, icon: 'none' })

}

}).catch(reason => {

that.maskShow = false;

console.log('Failed: ' + reason);

});

},

// 取消登录

cancelLogin () {

var pages = getCurrentPages(); // 获取当前页面栈

var prePage = pages[pages.length - 2]; // 上一个页面

prePage.data.init = true // A 页面 init方法 为true

setTimeout(() => {

uni.navigateBack({ delta: 1 }); // 返回上一页

}, 0)

},

// 领取奖励

getReward () {

this.popupShow = false

uni.showToast({ title: '领取成功' })

setTimeout(() => {

uni.navigateBack({ delta: 1 }); // 返回上一页

}, 300)

}

}

}

import {

mapMutations

} from 'vuex';

// import Json from '@/Json' //测试用数据

const form = require("@/utils/form/formValidation.js")

const util = require('@/utils/form/util.js')

import inputSms from './inputSMS.vue'

import awardPopup from '@/components/common/award-popup.vue'

export default {

components: {

inputSms,

awardPopup

},

data () {

return {

mobile: '',

btnText: "获取验证码",

password: '',

confirmPassword: '',

inviteId: '', // 邀请码

loading: false,

smsShow: false, // 验证码弹框

popupShow: false, // 奖励弹窗

}

},

onLoad () {

},

methods: {

...mapMutations(['login']),

inputChange (e) {

const key = e.currentTarget.dataset.key;

this[key] = e.detail.value;

},

navBack () {

uni.navigateBack();

},

navTo (url) {

uni.navigateTo({

url: url

})

},

// 检查表单

checkForm () {

let that = this;

//表单规则

let rules = [{

name: "mobile",

rule: ["required", "isMobile"],

msg: ["请输入手机号", "请输入正确的手机号"]

}, {

name: "password",

rule: ["required", "isEnAndNo"],

msg: ["请输入密码", "密码为8~20位数字和字母组合"]

}, {

name: "confirmPassword",

rule: ["required", "isEnAndNo"],

msg: ["请输入确认密码", "密码为8~20位数字和字母组合"]

}];

//进行表单检查

let formData = {

mobile: that.mobile,

password: that.password,

confirmPassword: that.confirmPassword,

};

let checkRes = "";

if (that.password !== that.confirmPassword) {

checkRes = "两次密码不匹配"

}

checkRes = form.validation(formData, rules);

if (checkRes) {

uni.showToast({ title: checkRes, icon: "none" });

return false

}

return true

},

// 下一步

nextStep () {

let that = this;

if (that.checkForm()) {

let params = {

phone: that.mobile,

msgType: 1 //类型：Number 必有字段 备注：0-通用 1-注册，2-登录，3-重置密码，4-修改信息

}

that.$http.post(that.requestUrl.api.sendVcode, params).then(res => {

// that.smsShow = true

// that.$refs.inputSms.reloadSms()

if (res.statusCode === 200 && res.data.code === 200) {

//获取短信

that.smsShow = true

that.$refs.inputSms.reloadSms()

} else {

uni.showToast({ title: res.data.data, icon: 'none' });

}

}).catch(err => {

uni.showToast({

title: err.errMsg,

icon: 'none'

});

});

}

},

// 注册

toRegister (vcode) {

let that = this;

if (!this.checkForm()) {

return

}

let params = {

phone: that.mobile,

vcode: vcode,

password: that.password,

}

if (this.inviteId) {

params.inviteId = that.inviteId

}

that.loading = true;

that.$http.post(that.requestUrl.api.registerByVCode, params).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

uni.showToast({ title: '注册成功', icon: 'success' })

if (that.globalConfig && that.globalConfig.MEMBER\_PROTECT\_DAYS > 0) {

that.loading = false

that.popupShow = true

} else {

setTimeout(() => {

uni.redirectTo({

url: '/pages/login/appLogin/loginByTel?mobile=' + that.mobile,

})

that.loading = false

}, 1500);

}

} else {

that.loading = false;

uni.showToast({ title: res.data.data, icon: 'none' });

}

})

},

// 领取奖励

getReward () {

let that = this;

that.popupShow = false

uni.showToast({ title: '领取成功' })

setTimeout(() => {

uni.redirectTo({

url: '/pages/login/appLogin/loginByTel?mobile=' + that.mobile,

})

}, 300)

}

},

}

import {

mapMutations,

mapState

} from 'vuex';

import setPassword from './setPassword.vue'

export default {

components: {

setPassword

},

data () {

return {

autoStatus: true,

errCode: '',

setPasswordShow: false,

resolve: {

'20202': '终端未开启SIM流量',

'30005': '设备不支持/未开启数据流量/其他原因',

'40101': '检查一下手机卡类型是否是正常运营商手机卡，关闭飞行模式后重新尝试',

'40201': '检查一下手机卡类型是否是正常运营商手机卡，关闭飞行模式后重新尝试',

'40301': '检查一下手机卡类型是否是正常运营商手机卡，关闭飞行模式后重新尝试',

}

}

},

computed: {

...mapState(['mpInfo', 'hasLogin']),

},

onLoad () {

this.isAutoLogin()

},

methods: {

...mapMutations(['login']),

navTo (url) {

uni.navigateTo({

url: url

})

},

// 判断是否支持一键登陆

isAutoLogin () {

let that = this

uni.getProvider({ //获取可用的服务提供商

service: 'oauth',

success: function (res) {

console.log('获取可用的服务提供商')

console.log(res.provider) // ['weixin', qq', 'univerify']

}

});

uni.preLogin({ //预登录

provider: 'univerify', //用手机号登录

success () {

console.log('预登录成功')

that.autoStatus = true

},

fail (err) {

//预登录失败

console.log('预登录失败')

that.autoStatus = false

console.log('错误码：' + err.errCode)

console.log(err.errMsg)

that.errCode = err.errCode

}

})

},

submitlogin () {

let that = this;

uni.login({ //正式登录，弹出授权窗

provider: 'univerify',

univerifyStyle: { // 自定义登录框样式

"fullScreen": true, // 是否全屏显示，true表示全屏模式，false表示非全屏模式，默认值为false。

"backgroundColor": "#ffffff", // 授权页面背景颜色，默认值：#ffffff

"phoneNum": {

"color": "#2281F5", // 手机号文字颜色 默认值：#000000

},

"authButton": {

"normalColor": "#3479f5", // 授权按钮正常状态背景颜色 默认值：#3479f5

"highlightColor": "#2861c5", // 授权按钮按下状态背景颜色 默认值：#2861c5（仅ios支持）

"disabledColor": "#73aaf5", // 授权按钮不可点击时背景颜色 默认值：#73aaf5（仅ios支持）

"textColor": "#ffffff", // 授权按钮文字颜色 默认值：#ffffff

"title": "本机号码一键登录" // 授权按钮文案 默认值：“本机号码一键登录”

}

},

success (res) {

// 正式登录成功

console.log(res.authResult); // {openid:'登录授权唯一标识',access\_token:'接口返回的 token'}

// 在得到access\_token后，通过callfunction调用云函数

uniCloud.callFunction({

name: 'getPhoneNumber', // 云函数名称

data: { //传给云函数的参数

'access\_token': res.authResult.access\_token, // 客户端一键登录接口返回的access\_token

'openid': res.authResult.openid // 客户端一键登录接口返回的openid

},

success (callRes) {

console.log('调用云函数成功')

console.log(callRes)

if (callRes && callRes.result && callRes.result.code === 0 && callRes.result.success && callRes.result.phoneNumber) {

that.registerByOneKey(callRes.result.phoneNumber)

} else {

uni.showToast({ title: "一键登录云函数解析出错", icon: 'none' });

}

},

fail (callErr) {

console.log('调用云函数出错' + callErr)

},

complete () {

uni.closeAuthView() //关闭授权登录界面

}

})

},

fail (err) {

// 正式登录失败

uni.showToast({ title: "正式登录失败", icon: 'none' });

that.errCode = err.errCode

console.log(err.errCode)

console.log(err.errMsg)

uni.closeAuthView() //关闭授权登录界面

}

})

},

// 一键登录

registerByOneKey (phone) {

let that = this;

that.$http.post(that.requestUrl.api.registerByOneKey, { phone: phone }).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

// 返回的token作了处理,要判断是否返回了password

uni.setStorage({

key: 'authorization',

data: res.data.data.token,

success: function () {

if (res.data.data.isSetPassword) {

// 已设置密码,直接登录

that.getUserInfo()

} else {

// 未设置密码

that.setPasswordShow = true

}

}

});

} else {

uni.showToast({ title: res.data.data, icon: 'none' });

}

})

},

// 获取用户信息

getUserInfo () {

let that = this;

that.$http.get(that.requestUrl.api.getUserInfo).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

that.login(res.data.data);

uni.showToast({ title: '登录成功' })

setTimeout(() => {

uni.reLaunch({ url: '/pages/index/index' });

}, 500)

} else {

uni.showToast({ title: res.data.data, icon: 'none' });

}

})

},

}

}

import { mapMutations, mapState } from 'vuex';

import advert from '@/components/advert/advert.vue'

export default {

components: {

advert

},

data () {

return {

answer: '',

noticeStr: '', // 轮播内容

question: {},

resultShow: false, // 结果弹框

result: null, // 0-答对，1-答错

rightTimes: 0, // 连对次数

errorTimes: 0, // 连错次数

tips: '', // 提示

emoji: '',

wordType: 'word',

rightEmoji: ['/static/images/biaoqing/love.png', '/static/images/biaoqing/happy.png', '/static/images/biaoqing/cool.png'],

}

},

computed: {

...mapState(['hasLogin', 'userInfo', 'globalConfig', 'advert'])

},

onLoad () {

let that = this;

let nameList = [

"据说答题好的人智商和情商都很高",

"快来试一下，测一下自己的应变能力",

"问题答得好，生活没烦恼",

"如果是闯关，你能一站到底吗",

"才貌双全，说的就是你吧",

"文明发言，健康答题，你是最棒的",

]

let noticeList = []

that.$u.randomArray(nameList).forEach((v, i) => {

that.noticeStr = nameList[i]

})

that.tips = that.$say.say(0, that.rightTimes)

if (this.globalConfig.PLATE\_LIST && this.globalConfig.PLATE\_LIST.includes('answer')) {

this.init()

}

},

// 下拉刷新

onPullDownRefresh () {

setTimeout(() => {

uni.stopPullDownRefresh()

this.init()

}, 500);

},

methods: {

...mapMutations(['login']),

init () {

if (this.hasLogin) {

if (this.userInfo.active === 0) {

this.getLoveWordAnswerList(); // 用户获取题目

} else {

uni.showToast({ title: '账号已被封号，无法操作', icon: 'none' })

}

}

},

// 用户获取题目

getLoveWordAnswerList () {

let that = this;

let params = {

pageIndex: 1,

pageSize: 1,

state: 1,

active: 1,

reportCount: 0,

orderBy: 'user',

}

that.$http.post(that.requestUrl.api.getLoveWordAnswerList, params).then(res => {

if (res.data && res.data.code === 200 && res.data.data.list.length > 0) {

that.total = res.data.data.total

that.question = res.data.data.list[0]

that.question.title = that.question.content

if (that.question.appId === 'wx91963481527504a7' || that.question.appId === 'wx3e84c6dd1c76bba1' || that.question.appId === '1110669493') {

// 文案

that.wordType = 'wenan'

}

that.question.downloadInitCount = Math.ceil(Math.random() \* 1000)

that.question.collectInitCount = Math.ceil(Math.random() \* 1000) + Math.ceil(Math.random() \* 1000)

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

}

})

},

// 读取用户信息

getUserInfo () {

let that = this;

that.$http.get(that.requestUrl.api.getUserInfo).then(res => {

if (res.data && res.data.code === 200) {

console.log('读取用户信息成功')

that.login(res.data.data); // 存储登录信息

}

})

},

nextGo () {

this.answer = ""

this.resultShow = false;

this.analysisShow = false;

this.getLoveWordAnswerList();

},

clear () {

this.answer = ''

},

navTo (url) {

uni.navigateTo({ url: url })

},

// 提交

submit () {

let that = this;

if (this.userInfo && this.userInfo.active !== 0) {

uni.showToast({ title: "您已被封号，无法提交", icon: 'none' })

return

}

if (!this.answer) {

uni.showToast({ title: "还没有作答哦，请先答题", icon: 'none' })

return

}

if (this.answer && this.answer.length < 6) {

uni.showToast({ title: "内容太少了，再加点吧", icon: 'none' })

return

}

let re = /select|update|delete|exec|count|'|"|=|;|>|<|%/i;

if (re.test(this.answer)) {

uni.showToast({ title: "内容不合规，请重新编辑", icon: 'none' })

return

}

let params = {

appId: this.requestUrl.api.appId,

belongId: this.question.id,

content: this.question.title + '\n' + this.answer,

title: this.question.title,

userId: this.userInfo.userId,

fId: '',

fIdName: '',

userName: this.userInfo.name,

avatar: this.userInfo.headIcon

}

this.$http.post(this.requestUrl.api.wordAnswerSave, params).then(res => {

if (res.data && res.data.code === 200) {

if (that.rightTimes < 4) {

that.rightTimes = that.rightTimes + 1

}

that.emoji = that.rightEmoji[that.$u.random(0, 2)]

that.tips = that.$say.say(0, that.rightTimes)

that.resultShow = true

that.getReward()

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

}

});

},

// 下发奖励

getReward () {

let that = this;

let params = { configKey: that.requestUrl.params.WORD }

that.$http.post(that.requestUrl.api.getReward, params).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

console.log("下发奖励成功")

}

})

},

report () {

let that = this;

let params = { wordId: this.question.id, operate: 'report' }

that.$http.post(that.requestUrl.api.loveWordOperate, params).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

uni.showToast({ title: '感谢您的提议', icon: 'none' })

that.nextGo();

}

})

},

}

}

import {

mapActions,

mapMutations,

mapState

} from 'vuex';

import wenan from '@/components/text-case/text-case.vue'

import result from '@/components/text-case/result.vue'

export default {

components: {

wenan,

result

},

data () {

return {

keywords: '', // 搜索词

wordList: [], // 文案列表

noDataShow: false, // 没有结果

loading: false, // 加载中

resultShow: false, // 保存结果

saveImageResult: false, // 保存图片结果

hasImage: false, // 是否有图片

isAuth: true, // 是否已授权

wordLength: 120, // 字符截取长度开启全文

pageSetting: {

pageIndex: 1, // 页码

pageSize: 10, // 每页页数

totalPages: 0, // 总页数

totalSize: 0, // 总数

},

}

},

computed: {

...mapState(['hasLogin', 'userInfo', 'mpInfo', 'globalConfig', 'advert'])

},

async onLoad (options) {

if (this.globalConfig.PLATE\_LIST && this.globalConfig.PLATE\_LIST.includes('poetry')) {

this.init();

}

},

onShow () {

/\* 建议放在onReady里执行，提前加载广告 \*/

let obj = {}

let advertId = null

// 插屏广告

// #ifdef MP-WEIXIN

obj = wx

advertId = this.advert.wxIndexInterstitialAd

// #endif

// #ifdef MP-QQ

obj = qq

advertId = this.advert.qqIndexInterstitialAd

// #endif

if (this.globalConfig.ADVERT\_SHOW === '0') {

// 显示广告

if (obj.createInterstitialAd && advertId) {

let interstitialAd = obj.createInterstitialAd({ adUnitId: advertId })

interstitialAd.onLoad(() => {

interstitialAd.show().catch((err) => {

console.error(err)

})

console.log('onLoad event emit')

})

interstitialAd.onError((err) => {

console.log('onError event emit', err)

})

interstitialAd.onClose((res) => {

console.log('onClose event emit', res)

})

}

}

},

onReady () { },

// 下拉刷新

onPullDownRefresh () {

setTimeout(() => {

uni.stopPullDownRefresh()

this.init()

}, 500);

},

// 到底

onReachBottom () {

if (this.pageSetting.pageIndex <= this.pageSetting.totalPages && !this.loading && this.hasLogin) {

this.getLoveWord()

}

},

methods: {

...mapActions(['handlePreviewImage',]),

init () {

this.pageSetting.pageIndex = 1;

this.wordList = [];

this.getLoveWord()

},

// 获取列表

getLoveWord () {

let that = this

this.noDataShow = false

let params = {}

params = {

pageIndex: this.pageSetting.pageIndex,

pageSize: this.pageSetting.pageSize,

state: 0,

active: 0,

byUser: true,

orderBy: 'create\_time DESC'

}

if (this.globalConfig.IS\_COMMON\_DATA === '0') {

params.appId = that.requestUrl.api.wenanAppId

} else if (this.globalConfig.IS\_COMMON\_DATA === '1') {

params.appId = that.requestUrl.api.appId

}

that.loading = true;

that.$http.post(that.requestUrl.api.getLoveWord, params).then(res => {

that.loading = false;

if (res.statusCode === 200 && res.data.code === 200) {

let resp = res.data.data.list

if (resp.length > 0) {

resp.forEach(v => {

v.allShow = v.content.length <= that.wordLength ? true : false

})

}

that.pageSetting.pageIndex += 1

if (that.pageSetting.pageIndex > res.data.data.pages) {

that.pageSetting.pageIndex = 1

}

that.wordList = that.wordList.concat(resp)

that.pageSetting.totalPages = res.data.data.pages

that.pageSetting.totalSize = res.data.data.total

// 无数据展示

if (that.wordList.length === 0) {

that.noDataShow = true

}

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

}

})

},

// 预览图片

previewImage (p) {

let params = {

images: p,

current: 0

}

this.handlePreviewImage(params)

},

// 一键保存

onSave (isAuth, saveImageResult, hasImage) {

this.isAuth = isAuth

this.saveImageResult = saveImageResult

this.hasImage = hasImage

this.resultShow = true

},

navTo (url) {

if (!this.hasLogin) {

uni.navigateTo({ url: '/pages/login/authorize' })

} else {

uni.navigateTo({ url: url })

}

},

}

}

import { mapState } from 'vuex';

export default {

components: {

},

data () {

return {

show: false,

list: [],

content: '',

title: '',

fId: '',

fIdName: '',

image: [],

action: 'https://www.tuodanlianai.top/',

sizeType: ['compressed'],

hasUpload: false,

pickerIndex: 0,

}

},

computed: {

...mapState(['hasLogin', 'userInfo', 'globalConfig', 'advert'])

},

onLoad (option) {

if (this.globalConfig && this.globalConfig.SAFE\_SHOW === '0') {

uni.setNavigationBarTitle({ title: '记录' });

} else {

uni.setNavigationBarTitle({ title: '备用' });

}

this.getLabelTwo()

},

onShow () {

/\* 建议放在onReady里执行，提前加载广告 \*/

let obj = {}

let advertId = null

// 插屏广告

// #ifdef MP-WEIXIN

obj = wx

advertId = this.advert.wxIndexInterstitialAd

// #endif

// #ifdef MP-QQ

obj = qq

advertId = this.advert.qqIndexInterstitialAd

// #endif

if (this.globalConfig.ADVERT\_SHOW === '0') {

// 显示广告

if (obj.createInterstitialAd && advertId) {

let interstitialAd = obj.createInterstitialAd({ adUnitId: advertId })

interstitialAd.onLoad(() => {

interstitialAd.show().catch((err) => {

console.error(err)

})

console.log('onLoad event emit')

})

interstitialAd.onError((err) => {

console.log('onError event emit', err)

})

interstitialAd.onClose((res) => {

console.log('onClose event emit', res)

})

}

}

},

onUnload () {

if (!this.hasUpload && this.image !== "") {

this.deleteImgs()

}

},

methods: {

// 获取类目

getLabelTwo () {

let that = this

let params = {}

params = { state: 0, pageIndex: 1, pageSize: 9999 }

if (this.globalConfig.IS\_COMMON\_DATA === '0') {

params.appId = that.requestUrl.api.wenanAppId

} else if (this.globalConfig.IS\_COMMON\_DATA === '1') {

params.appId = that.requestUrl.api.appId

} else if (this.globalConfig.IS\_COMMON\_DATA === '2') {

params.appId = ''

}

that.loading = true;

that.$http.post(that.requestUrl.api.getLabelTwo, params).then(res => {

that.loading = false;

that.list = []

if (res.statusCode === 200 && res.data.code === 200) {

let reps = res.data.data.list

if (reps.length > 0) {

reps.forEach((v, i) => {

v.value = v.id;

v.label = v.title;

if (i !== 0) {

that.list.push(v.title)

}

})

}

}

})

},

// 调用接口，编辑数据

saveLoveWord () {

let that = this

if (this.userInfo.active !== 0) {

uni.showToast({ title: '账号已被封号，无法操作', icon: 'none' })

return

}

if (this.userInfo.wordToday >= this.globalConfig.WORD\_LIMIT) {

uni.showToast({ title: '今日提交次数已达上限', icon: 'none' })

return

}

let params = {

title: this.title,

content: this.content,

state: 1,

active: 1,

isTop: 1,

downloadInitCount: 0,

collectInitCount: 0,

image: '',

fId: this.fId,

fIdName: this.fIdName,

userId: this.userInfo.userId,

userName: this.userInfo.name,

avatar: this.userInfo.headIcon

}

if (this.image.length > 0) {

params.image = this.image.map(\_ => \_.url).toString()

}

if (this.globalConfig.IS\_COMMON\_DATA === '0') {

params.appId = that.requestUrl.api.commonAppId

} else if (this.globalConfig.IS\_COMMON\_DATA === '1') {

params.appId = that.requestUrl.api.appId

}

that.$http.post(that.requestUrl.api.saveLoveWord, params).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

that.hasUpload = true

uni.showToast({ title: '提交成功', mask: true })

setTimeout(() => {

that.getReward();

}, 1000)

setTimeout(() => {

uni.navigateBack({ delta: 1 }); // 返回上一页

}, 1500)

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

}

})

},

// 下发奖励

getReward () {

let that = this;

let params = { configKey: that.requestUrl.params.WORD }

that.$http.post(that.requestUrl.api.getReward, params).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

console.log("下发奖励成功")

}

})

},

onConfirm (e) {

this.pickerIndex = parseInt(e.detail.value)

this.fId = e.detail.value

this.fIdName = this.list[parseInt(e.detail.value)].title

},

// 读取文件 -- 头像

afterReadImage (e) {

this.multipleUpload(e)

},

// 多张图片上传

async multipleUpload (event) {

let that = this;

// 当设置 mutiple 为 true 时, file 为数组格式，否则为对象格式

let lists = [].concat(event.file)

if (lists.length > 9) {

uni.showToast({

title: '最多上传9张图片',

icon: 'none'

})

return

}

for (let i = 0; i < lists.length; i++) {

const result = await this.uploadDo(lists[i])

}

},

// 执行上传

uploadDo (event) {

let that = this;

let url = that.requestUrl.api.baseUrl + that.requestUrl.api.fileUpload;

// #ifdef H5

url = that.requestUrl.api.h5Sign + that.requestUrl.api.fileUpload;

// #endif

return new Promise((resolve, reject) => {

uni.uploadFile({

url: url,

filePath: event.url,

name: 'uploadFile',

formData: {

user: 'test'

},

success (res) {

resolve(res.data.data)

let resp = JSON.parse(res.data)

if (resp && resp.code === 200) {

let obj = {};

obj.url = that.requestUrl.api.fileUrl + resp.data

obj.name = '图' + that.image.length + 1

that.image = []

that.image.push(obj)

}

}

});

})

},

// 删除图片

deleteImgs () {

let that = this

let params = {

pathList: this.image.map(\_ => \_.url).toString()

}

that.$http.post(that.requestUrl.api.deleteFile, params).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

that.image = []

console.log("删除图片")

}

})

},

// 文件超出大小限制

oversize () {

uni.showToast({

title: "图片最大不能超过2M",

icon: 'none'

})

},

},

}

import { mapMutations, mapState } from 'vuex';

export default {

components: {

},

data () {

return {

articleData: {},

}

},

computed: {

...mapState(['hasLogin', 'userInfo', 'advert', 'globalConfig'])

},

// 下拉刷新

onPullDownRefresh () {

setTimeout(() => {

uni.stopPullDownRefresh()

if (this.articleData.id !== null) {

this.findArticle()

}

}, 500);

},

onLoad (options) {

this.articleData.id = options.id ? options.id : ''

if (this.articleData.id !== null) {

this.findArticle()

}

if (this.hasLogin) {

if (this.userInfo.energy <= 0) {

uni.showModal({

title: '搜索次数不足',

content: '做任务/升级会员，获取搜索次数',

showCancel: false,

success: function (res) {

if (res.confirm) {

uni.reLaunch({ url: '/pages/my/index' });

}

}

});

return

} else {

this.consume()

}

}

},

onShow () {

/\* 建议放在onReady里执行，提前加载广告 \*/

let obj = {}

let advertId = null

// 插屏广告

// #ifdef MP-WEIXIN

obj = wx

advertId = this.advert.wxIndexInterstitialAd

// #endif

// #ifdef MP-QQ

obj = qq

advertId = this.advert.qqIndexInterstitialAd

// #endif

if (this.globalConfig.ADVERT\_SHOW === '0') {

// 显示广告

if (obj.createInterstitialAd && advertId) {

let interstitialAd = obj.createInterstitialAd({ adUnitId: advertId })

interstitialAd.onLoad(() => {

interstitialAd.show().catch((err) => {

console.error(err)

})

console.log('onLoad event emit')

})

interstitialAd.onError((err) => {

console.log('onError event emit', err)

})

interstitialAd.onClose((res) => {

console.log('onClose event emit', res)

})

}

}

},

methods: {

...mapMutations(['login']),

// 消耗体力

consume () {

let that = this

if (this.globalConfig && this.globalConfig.IS\_CONSUME\_TIMES === '0') {

let params = { type: 'article' }

that.$http.post(that.requestUrl.api.consume, params).then(res => {

console.log("消耗体力")

})

}

},

// 获取文章

findArticle () {

let that = this;

that.$http.get(that.requestUrl.api.findArticle + '?id=' + that.articleData.id).then(res => {

if (res.statusCode === 200 && res.data.code === 200) {

that.articleData = res.data.data

that.articleData.isAdvert = that.articleData.isAdvert.toString()

console.log("that.articleData.isAdvert ：" + that.articleData.isAdvert)

that.articleData.content = res.data.data.content.replace(/\<img/gi, '<img style="width:100%;height:auto;display:block; margin: 10px 0;"')

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

}

})

},

}

}

import { mapState } from 'vuex';

import articleItem from '@/components/article/item.vue'

export default {

components: {

articleItem

},

data () {

return {

currentTab: 0,

tabSelectedList: [],

loading: false, // 加载中

triggered: false, // 设置当前下拉刷新状态，true 表示下拉刷新已经被触发，false 表示下拉刷新未被触发

}

},

computed: {

...mapState(['hasLogin', 'userInfo', 'advert', 'globalConfig']),

},

onLoad (option) {

if (this.globalConfig.PLATE\_LIST && this.globalConfig.PLATE\_LIST && this.globalConfig.PLATE\_LIST.includes('article')) {

this.getArticleGroup()

}

},

onShow () {

/\* 建议放在onReady里执行，提前加载广告 \*/

let obj = {}

let advertId = null

// 插屏广告

// #ifdef MP-WEIXIN

obj = wx

advertId = this.advert.wxIndexInterstitialAd

// #endif

// #ifdef MP-QQ

obj = qq

advertId = this.advert.qqIndexInterstitialAd

// #endif

if (this.globalConfig.ADVERT\_SHOW === '0') {

// 显示广告

if (obj.createInterstitialAd && advertId) {

let interstitialAd = obj.createInterstitialAd({ adUnitId: advertId })

interstitialAd.onLoad(() => {

interstitialAd.show().catch((err) => {

console.error(err)

})

console.log('onLoad event emit')

})

interstitialAd.onError((err) => {

console.log('onError event emit', err)

})

interstitialAd.onClose((res) => {

console.log('onClose event emit', res)

})

}

}

},

// 下拉刷新

onPullDownRefresh () {

setTimeout(() => {

uni.stopPullDownRefresh()

}, 500);

},

methods: {

// 获取文章分类列表

getArticleGroup () {

let that = this

let params = {}

params = {

pageIndex: 1,

pageSize: 999,

state: 0,

orderBy: 'is\_top ASC'

}

if (this.globalConfig.IS\_COMMON\_DATA === '0') {

params.appId = that.requestUrl.api.commonAppId

} else if (this.globalConfig.IS\_COMMON\_DATA === '1') {

params.appId = that.requestUrl.api.appId

}

that.tabSelectedList = []

that.loading = true;

that.$http.post(that.requestUrl.api.getArticleGroup, params).then(res => {

that.loading = false;

if (res.statusCode === 200 && res.data.code === 200 && res.data.data.list.length > 0) {

let resp = res.data.data.list

if (resp.length > 0) {

resp.forEach((v, i) => {

let obj = {};

obj.name = v.title

obj.noDataShow = false

obj.isGetData = false

obj.id = v.id;

obj.title = v.title;

obj.name = v.title; // 设置标题

obj.pageIndex = 1;

obj.pageSize = 10;

obj.dataList = [];

if (i < 6) {

that.tabSelectedList.push(obj)

} else {

that.tabUnSelectedList.push(obj)

}

})

that.getArticleList()

}

}

}).catch(reason => {

that.loading = false;

console.log('Failed: ' + reason);

});

},

// 触底

onLower () {

if (this.tabSelectedList[this.currentTab].pageIndex

&& this.tabSelectedList[this.currentTab].totalPages

&& this.tabSelectedList[this.currentTab].pageIndex > this.tabSelectedList[this.currentTab].totalPages) {

return

}

this.getArticleList();

},

// 获取文章列表

getArticleList () {

let that = this

let params = {}

let tabItem = this.tabSelectedList[this.currentTab];

params = {

pageIndex: tabItem.pageIndex,

pageSize: tabItem.pageSize,

state: 0,

fId: tabItem.id,

orderBy: 'is\_top ASC, create\_time DESC'

}

if (this.globalConfig.IS\_COMMON\_DATA === '0') {

params.appId = that.requestUrl.api.commonAppId

} else if (this.globalConfig.IS\_COMMON\_DATA === '1') {

params.appId = that.requestUrl.api.appId

}

that.loading = true;

that.$http.post(that.requestUrl.api.getArticleList, params).then(res => {

that.loading = false;

if (res.statusCode === 200 && res.data.code === 200) {

let resp = res.data && res.data.data && res.data.data.list ? res.data.data.list : []

if (res.data.data.list.length > 0) {

resp.forEach(v => {

var images = []

if (v.images.length > 0) {

images = v.images.split(";")

}

v.images = images

})

}

tabItem.pageIndex += 1

tabItem.dataList = tabItem.dataList.concat(resp)

tabItem.isGetData = true;

tabItem.totalPages = res.data.data.pages

tabItem.totalSize = res.data.data.total

tabItem.noDataShow = tabItem.dataList.length === 0 ? true : false

if (tabItem.totalSize === 10 || (tabItem.pageIndex > tabItem.totalPages && tabItem.totalPages > 1)) {

// 没有更多数据

tabItem.noMoreDate = true

}

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

}

}).catch(reason => {

that.loading = false;

console.log('Failed: ' + reason);

});

},

// 下拉刷新

onRefresh () {

console.log("onRefresh")

if (this.triggered) return

//界面下拉触发，triggered可能不是true，要设为true

this.triggered = true;

this.tabSelectedList[this.currentTab].dataList = []

this.tabSelectedList[this.currentTab].pageIndex = 1

this.tabSelectedList[this.currentTab].noDataShow = false

this.getArticleList();

setTimeout(() => {

//触发onRestore，并关闭刷新图标

this.triggered = false;

}, 1000)

},

navTo (url) {

if (this.hasLogin) {

uni.navigateTo({ url: url })

} else {

uni.navigateTo({ url: '/pages/login/authorize' })

}

},

}

}

import { mapMutations, mapState } from 'vuex';

import staticGoods from '@/components/goods/static-goods.vue'

import basicGoods from '@/components/goods/basic-goods.vue'

import advert from '@/components/advert/advert.vue'

export default {

components: {

basicGoods,

staticGoods,

advert

},

data () {

return {

src: 'https://ss3.bdstatic.com/70cFv8Sh\_Q1YnxGkpoWK1HF6hhy/it/u=3032562195,2010430756&fm=26&gp=0.jpg',

activityList: [], // 活动列表

activityIdList: [], // 活动id列表

ordrList: [], // 订单列表

loading: false, // 加载中

pageSetting: {

pageIndex: 1, // 页码

pageSize: 10, // 每页页数

totalPages: 0, // 总页数

totalSize: 0, // 总数

},

}

},

computed: {

...mapState(['hasLogin', 'userInfo', 'globalConfig', 'advert']),

},

onLoad (option) {

},

onShow () {

if (this.globalConfig.BARGAIN\_SHOW === '0') {

this.init()

}

/\* 建议放在onReady里执行，提前加载广告 \*/

let obj = {}

let advertId = null

// 插屏广告

// #ifdef MP-WEIXIN

obj = wx

advertId = this.advert.wxIndexInterstitialAd

// #endif

// #ifdef MP-QQ

obj = qq

advertId = this.advert.qqIndexInterstitialAd

// #endif

if (this.globalConfig.ADVERT\_SHOW === '0') {

// 显示广告

if (obj.createInterstitialAd && advertId) {

let interstitialAd = obj.createInterstitialAd({ adUnitId: advertId })

interstitialAd.onLoad(() => {

interstitialAd.show().catch((err) => {

console.error(err)

})

console.log('onLoad event emit')

})

interstitialAd.onError((err) => {

console.log('onError event emit', err)

})

interstitialAd.onClose((res) => {

console.log('onClose event emit', res)

})

}

}

},

// 下拉刷新

onPullDownRefresh () {

setTimeout(() => {

uni.stopPullDownRefresh()

this.init()

}, 500);

},

methods: {

// 初始化

init () {

this.activityList = [] // 活动列表

this.ordrList = [] // 订单列表

this.pageSetting.pageIndex = 1

if (this.hasLogin) {

this.getOrderList();

} else {

this.getActivityList()

}

},

// 获取订单列表

getOrderList () {

let that = this

// 状态：0-砍价中 1-砍价成功 2-砍价失败 3-已兑换

let params = { pageIndex: 1, pageSize: this.globalConfig.ACTIVITY\_ORDER\_MAX, state: 0, userId: this.userInfo.userId }

that.loading = true;

that.activityIdList = [];

that.$http.post(that.requestUrl.api.getOrderList, params).then(res => {

that.loading = false;

if (res.statusCode === 200 && res.data.code === 200) {

that.ordrList = res.data.data.list

if (that.ordrList.length > 0) {

// 倒计时 处理

that.ordrList.forEach(v => {

let c\_t = new Date().getTime(); // 此时时间戳

let f\_t = v.createTime + v.duration \* 60 \* 60 \* 1000; // 结束时间戳

if (f\_t > c\_t) {

v.timestamp = f\_t - c\_t;

} else {

v.timestamp = 0

}

that.activityIdList.push(v.activityId)

})

}

that.getActivityList()

}

}).catch(reason => {

that.loading = false;

console.log('Failed: ' + reason);

});

},

// 获取活动列表

getActivityList () {

let that = this

let params = {}

params = {

pageIndex: this.pageSetting.pageIndex,

pageSize: this.pageSetting.pageSize,

appId: this.requestUrl.api.appId,

state: 0,

}

that.loading = true;

that.$http.post(that.requestUrl.api.getActivityList, params).then(res => {

that.loading = false;

if (res.statusCode === 200 && res.data.code === 200) {

if (res.data.data.list.length > 0) {

res.data.data.list.forEach(v => {

if (that.activityIdList.indexOf(v.id) === -1) {

that.activityList.push(v)

}

})

}

that.pageSetting.pageIndex += 1

that.pageSetting.totalPages = res.data.data.pages

that.pageSetting.totalSize = res.data.data.total

}

}).catch(reason => {

that.loading = false;

console.log('Failed: ' + reason);

});

},

// 创建订单

handleActivityBtn (p) {

let that = this

if (this.hasLogin) {

if (p.isAdvert === 1) {

// 不需要广告

this.insertOrder(p)

} else {

let obj = {}

let advertId = null

// #ifdef MP-WEIXIN

obj = wx

advertId = this.advert.wxActivityOrderRewardAd

// #endif

// #ifdef MP-QQ

obj = qq

advertId = this.advert.qqActivityOrderRewardAd

// #endif

if (this.globalConfig.ADVERT\_SHOW === '0') {

// 在页面中定义激励视频广告

let videoAd = null

// 在页面onLoad回调事件中创建激励视频广告实例

if (obj.createRewardedVideoAd && advertId) {

videoAd = obj.createRewardedVideoAd({

adUnitId: advertId

})

videoAd.onLoad(() => { })

videoAd.onError((err) => { })

videoAd.onClose((res) => {

if (res && res.isEnded || res === undefined) {

//正常播放结束,可以下发奖励

that.insertOrder(p)

} else {

that.loading = false;

//不下发奖励

obj.showToast({ title: '任务未完成', icon: 'none' })

}

})

}

// 用户触发广告后，显示激励视频广告

if (videoAd) {

videoAd.show().catch(() => {

// 失败重试

videoAd.load()

.then(() => videoAd.show())

.catch(err => {

that.loading = false;

console.log('激励视频 广告显示失败')

})

})

}

}

}

} else {

uni.navigateTo({ url: '/pages/login/authorize' })

}

},

// 生成订单

insertOrder (p) {

if (this.hasLogin) {

if (this.userInfo.active !== 0) {

uni.showToast({ title: '账号已被封号，无法操作', icon: 'none' })

return

}

let that = this

let params = {

activityId: p.id

}

that.loading = true;

that.$http.post(that.requestUrl.api.insertOrder, params).then(res => {

that.loading = false;

if (res.statusCode === 200 && res.data.code === 200) {

that.navTo('/pages/activity/detail?orderNo=' + res.data.data)

} else {

uni.showToast({ title: res.data.data, icon: 'none' })

}

}).catch(reason => {

that.loading = false;

console.log('Failed: ' + reason);

});

} else {

uni.navigateTo({ url: '/pages/login/authorize' })

}

}

}

}