

```
<template>

<view class="identify-dishes">

  <image :src="src" mode="aspectFill" class="full-screen-image"></image>

  <view class="dishes">

    <view class="dishes_loading" v-if="loading">

      <view >

        <u-loading mode="flower" size="42"></u-loading>

        <text> {{ loadingText }}...</text>

      </view>

    </view>

    <view v-else>

      <view v-if="goodList.length" >

        <view class="title">

          <text style="color: #333333; font-weight: 600">本餐食物</text>

        </view>

        <scroll-view scroll-y class="content" :scroll-x="false">

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

            <view class="item-left">

              <image

                class="image"

              />

              <view class="dishes-text">

                <view class="dishes-name">{{ good.name }}</view>

                <text>热量 {{ good.energy }} 千卡</text>

              </view>

            </view>

            <view class="item-right">

              <view class="custom-input">

                <u-input

                  :auto-height="true"

                  type="number"

                  placeholder=""

                  v-model="good.weight"

                  :border="false"

                  @blur="(e) => handleInputChange(e, index, good)"

                  height="60"

                  :customStyle="{

                    textAlign: 'center',

                    fontSize: '30rpx',

                    color: '#333',

                    background: '#EDF3FF',

                    borderRadius: '16rpx',

                    width: '150rpx',

                  }"

                ></u-input>

                &nbsp;  克

              </view>

              <u-icon

                size="40"

                @click="handleDelete(index)"

              />

            </view>

          </view>

        </scroll-view>

      </view>

    </view>

  </view>

</template>
```

```

        style=" padding-right: 22rpx;"
      ></u-icon>
    </view>
  </view>
</scroll-view>
<view class="save-btn">
  <u-button
    shape="circle"
    size="default"
    :custom-style="{
      background: '#1A65ED',
      borderRadius: '44rpx',
      color: '#fff',
      margin:'0 40rpx'
    }"
    @click="handleSave"
    type="primary"
  >保存
  </u-button>
</view>
</view>
<view v-else style="display: flex;justify-content: center;align-items: center; height: 50vh;">
  <view class="empty">
    <image
      class="empty-image"
    />
  </view>
  无法识别,<br />
  请
  <text class="redirect" @click="handlePhone">重新上传</text> 图片或
  <text class="redirect" @click="goToSearch">手动搜索 </text>添加
</view>
</view>
</view>
</view>
</view>
</view>
</view>
</view>
</template>
<script>
import { apiAll } from "@/utils/aiEncapsulation/util";
import { callCozeAgent } from "@/utils/aiEncapsulation/dialogue";
import { retrieveAgent } from "@/utils/aiEncapsulation/retrieve";
import { listMessageAgent } from "@/utils/aiEncapsulation/listMessage";
import { RunAgent } from "@/utils/aiEncapsulation/run";
import { image as uploadImageApi } from "@/api/upload";
import * as MealApi from "@/api/meal";
import CustomLoading from '@/components/custom-loading/custom-loading.vue';
export default {
  components(){
    CustomLoading
  }
}
```

```
},
data() {
  return {
    src: "",
    srcUrl:"",
    goodList: [],
    mealId: 1,
    meal_date: "",
    loading: true,
    energy:0,
    loadingText:'正在上传图片中',
  };
},
watch: {
  goodList(newValue, oldValue) {
    console.log('goodList 发生了变化', newValue, oldValue);
  }
},
async onLoad(options) {
  const parseJSON = JSON.parse(options.toJSON || "{}");
  this.mealId = parseJSON.mealId;
  this.meal_date = parseJSON.selectDate;
  this.src=parseJSON.src
  const imagePath = parseJSON.src;
  try {
    const imgRes = await this.uploadImage(imagePath);
    this.srcUrl=imgRes.preview_url;
    await this.runWorkFlow();
  } catch (error) {
    uni.showToast({
      title: "图片处理失败",
      icon: "none",
    });
  }
},
methods: {
  async runWorkFlow(){
    const response = await RunAgent(
      {
        image:this.srcUrl,
        type:'这是什么菜'
      }
    )
    if(response.code == 0){
      const good = JSON.parse(response?.data)?.output || [];
      this.goodList=[...good];
    }
    this.loading=false;
  },
  async uploadImage(filePath) {
```

```
this.loadingText='正在上传图片中'

try {
  if (!filePath || typeof filePath !== "string") {
    throw new Error("文件路径无效");
  }
  const files = [
    {
      path: filePath,
      name: "file",
    },
  ];
  const result = await uploadImageApi(files, true);
  if (!result || !result.length) {
    throw new Error("上传失败");
  }
  return result[0];
} catch (error) {
  console.error("上传图片失败:", error);
  throw error;
},

async pollCallCozeAgent(conversation_id, chat_id) {
  const intervalTime = 1000;
  const maxAttempts = 1000000;
  let attempts = 0;
  while (attempts < maxAttempts) {
    try {
      const response = await retrieveAgent(conversation_id, chat_id);
      if (response.data.status === "completed") {
        console.log("轮询结束，接口返回 completed", response);
        return response;
      }
      if (response.data.status === "failed") {
        console.log("轮询结束，接口返回 failed", response);
        return {};
      }
      await new Promise((resolve) => setTimeout(resolve, intervalTime));
    } catch (error) {
      console.error("轮询过程中出错:", error);
      break;
    }
    attempts++;
  }
  console.error("达到最大尝试次数，未检测到接口返回 completed");
  return null;
},

async sendMessage() {
  const messageContent = [
    {
      type: "image",
```

```

        file_url: this.srcUrl,
    },
    {
        type: "text",
        text: "这是什么菜",
    },
    {
        type: "text",
        text: '123',
    },
];
try {
    const response = await callCozeAgent(
        JSON.stringify(messageContent),
        (update) => {
            console.log("update", update);
        },
        null,
        true
    );
    const conversation_id = response.data.conversation_id;
    const chat_id = response.data.id;
    this.loadingText='正在识别菜品中'
    this.pollCallCozeAgent(conversation_id, chat_id).then((res) => {
        listMessageAgent(conversation_id, chat_id).then((response) => {
            console.log('sasa',JSON.parse(response?.data[1]?.content)?.output)
            if(response.code == 0){
                const good = JSON.parse(response?.data[1]?.content)?.output || [];
                this.goodList=[...good];
            }
            this.loading=false;
        });
    });
} catch (error) {
    console.error("调用 Coze API 失败 : ", error);
    uni.showToast({
        title: "发送失败",
        icon: "none",
        duration: 2000,
    });
}
},
removeItemAtIndex(array, index) {
    if (index >= 0 && index < array.length) {
        array.splice(index, 1);
    }
},
handleDelete(index) {
    this.removeItemAtIndex(this.goodList, index);
},

```

```

handleSave() {
  const dishes = this.goodList.map(item => {
    const {
      id: dishes_id,
      weight: dishes_weight,
      name: dishes_name
    } = item;
    return {
      dishes_id: Number(dishes_id),
      dishes_weight: Number(dishes_weight || 0) ,
      dishes_name
    };
  });
  const params = {
    form: {
      meal_times: this.mealId,
      meal_date: this.meal_date,
      dishes,
    },
  };
  MealApi.mealAdd(params).then((res) => {
    this.timer = setTimeout(() => {
      uni.showToast({
        title: res.message,
        icon: "none",
      });
      uni.navigateBack({
        delta: 1
      });
    }, 1000);
  });
},
handleInputChange(e, index, good) {
  good.weight = e;
  const params={
    id:Number(good.id),
    weight:e||0
  }
  MealApi.detailWithWeight(params).then(res=>{

```

```

<template>
<view class="container">
<view class="search">
<u-search
  clearabled
  placeholder="请输入食物名称"
  shape="round"
  animation="true"
  :show-action="false"
  v-model="searchFood"
  @search="handleSearchFood"

```

```

    ></u-search>
</view>
<commodity
  ref="mescrollItem"
  :list="list"
  :setting="setting"
  @addFood="handleAddFood"
  :searchFood="searchFood"
/>
<view class="goods-carts" :style="{ zIndex: foodVisiable && !showEditFoodModal ? 99999 : 999 }">
  <view class="add-meal">
    <view class="image-with-badge" @click="handleAddDetail">
      <image
        :src="'${ossUrl}static/${
          badge
            ? 'shop'
            : 'shop_disabled'
          }.svg'"
        style="width: 80rpx; height: 80rpx"
      />
      <u-badge
        :is-dot="false"
        type="error"
        :absolute="true"
        :offset="[0,0]"
        :count="badge"
      ></u-badge>
    </view>
    <view class="meal-title">{{ mealMap[mealId] }}</view>
  </view>
  <u-button shape="circle" size="large" type="primary" @click="handleSubmit()" :disabled="disabledBtn"
    :custom-style="{
      padding: '22rpx 118rpx',
      fontSize: '30rpx',
      height: '86rpx',
      background: 'linear-gradient(90deg, #389EF7 0%, #1A65ED 100%)',
      border: 'none'
    }"
  >完成</u-button>
</view>
<u-popup
  v-model="foodVisiable"
  mode="bottom"
  length="48%"
  border-radius="16"
  :safe-area-inset-bottom="true"
>
  <view class="goods-sum">
    <view class="goods-sum-title">
      共 {{ badge }} 条记录,总计{{ energy }}千卡</view>

```

```
>
<scroll-view scroll-y="true" style="height: 50%">
  <view
    v-for="(item, index) in $refs.mescrollItem?.selectedFoodList || []"
    :key="index"
  >
    <view class="goods-item">
      <view class="goods-item-left">
        <image
          class="image"
          :src="item?.file_url || `${ossUrl}default_dish.png`"
        ></image>
      </view>
      <view class="goods-item-right">
        <view class="goods-name" @click="handleShowSubCate(item)">
          <text class="twoline-hide">{{ item.dishes_name }}</text>
          <view class="online-hide">
            <template v-if="item?.source == 3">
              <text class="price-x">{{ item.energy }} 千卡</text> /
              <text >{{ item.dishes_number }}{{ item.unit_name }}</text>
            </template>
            <template v-else>
              <text class="price-x">{{ item.energy }} 千卡</text> /
              <text >{{ item.weight }} 克</text>
            </template>
          </view>
        </view>
      </view>
      <view class="goods-item-desc" @click="handleDelete(index, item)">
        <u-icon
          size="32"
          :name="`${ossUrl}delete.svg`"
        ></u-icon>
      </view>
    </view>
  </view>
</scroll-view>
</view>
</u-popup>
<edit-food
  :value="showEditFoodModal"
  :info="goodInfo"
  @close-popup="handleClosePopup"
  :goodWeight="goodWeight"
  :noApi="true"
  :showKitchen="false"
/>
<view v-if="showClockinOverlay" class="clockin-overlay">
  <view class="modal-mask">
    <view class="content-wrapper">
```

```
<view class="modal-content">  
  <view class="title">打卡成功</view>  
  
  <view class="score">  
    <text class="plus">+</text>  
    <text class="num">1</text>  
    <text class="unit">积分</text>  
  </view>  
  
  <view class="desc">  
    <view>成功记录一次饮食，</view>  
    <view>要多努力哦 ~ </view>  
  </view>  
  
  <view class="confirm-btn" @click="roger()">  
    知道了  
  </view>  
</view>  
</view>  
</view>  
</template>  
<script>  
  
import MescrollCompMixin from "@uni_modules/mescroll-uni/components/mescroll-uni/mixins/mescroll-comp";  
import SettingKeyEnum from "@common/enum/setting/Key";  
  
import { PageCategoryStyleEnum } from "@common/enum/store/page/category";  
import Search from "@components/search";  
import Primary from "./components/primary";  
import Secondary from "./components/secondary";  
import Commodity from "./components/commodity";  
  
import * as MealApi from "@api/meal";  
import { checkLogin } from "@core/app";  
import EditFood from ".edit-food.vue";  
import config from '@config.js'  
import storage from "@utils/storage";  
  
let lastRefreshTime;  
  
export default {  
  components: {  
    Search,  
    Primary,  
    Secondary,  
    Commodity,  
    EditFood  
  },  
  mixins: [MescrollCompMixin],  
  data() {  
    return {  
      ossUrl: config.ossUrl,  
      PageCategoryStyleEnum,  
      list: [],  
      setting: {},  
      isLoading: true,  

```

```

    badge: 0,
    energy: 0,
    selectedGoodList: [],
    mealId: 1,
    mealMap: {
      1: "早餐及早加餐",
      2: "午餐及午加餐",
      3: "晚餐及晚加餐",
    },
    searchFood: "",
    selectedFoodList: [],
    meal_date: "",
    timer: null,
    foodVisiable: false,
    showEditFoodModal: false,
    disabledBtn:false,
    showClockinOverlay: false
  };
},
onLoad(options) {
  const parseJSON = JSON.parse(storage.get('addMeal') || "{}");
  console.log('parseJSON',parseJSON)
  this.onRefreshPage();
  this.mealId = parseJSON.mealId;
  this.meal_date = parseJSON.selectDate;
},
onShow() {
  const isLogin = checkLogin();
  if (isLogin) {
    this.onRefreshPage();
    this.$refs.mescrollItem.onRefreshList();
    const parseJSON = JSON.parse(storage.get('addMeal') || "{}");
    this.mealId = parseJSON.mealId;
    this.meal_date = parseJSON.selectDate;
  }
},
onUnload() {
  clearTimeout(this.timer);
  try {
    uni.$emit('wl-kitchen-disconnect-all');
  } catch (err) {
    console.error(err);
  }
},
methods: {
  handleShowSubCate(item) {
    this.showEditFoodModal = true;
    const isCustom= item.dishes_source == 3;
    item.custom_weight=item.weight;
    this.goodInfo=item;
  }
}

```

```

    this.goodWeight=isCustom ? item.dishes_number :item.weight;
  },
  handleClosePopup(goodInfo){
    this.showEditFoodModal = false;
    this.goodWeight=null;
    this.$refs.mescrollItem.handleEditInfo(goodInfo)
  },
  handleAddDetail() {
    if (!this.badge) return;
    this.foodVisiable = !this.foodVisiable;
  },
  async handleSubmit() {
    let resType = false;
    const getdata ={form: { meal_times: this.mealId, meal_date: this.meal_date }}
    MealApi.scoreDiet(getdata).then((result) => {
      resType = result.data.exists
    })
    if (this.disabledBtn) return;
    this.disabledBtn = true;
    try {
      const selectedFoodList = this.$refs.mescrollItem?.selectedFoodList;
      selectedFoodList.forEach(item=>{
        if(item.source == 3){
          item.weight = item.custom_weight;
        }
      })
      const dishes=selectedFoodList.map(item=>{
        return {
          dishes_id: item.dishes_id,
          dishes_number: item.dishes_number,
          dishes_name: item.dishes_name,
          dishes_weight: item.dishes_source == 3 ? item.custom_weight : item.dishes_weight,

```

```

<template>

```

```

<view class="detail">

```

```

  <view class="card" style="margin: 20rpx 0;">

```

```

    <view class="card-title" style="align-items: center;">

```

```

      <view>

```

```

        <image :src=""${ossUrl}static/home/${imageMap[mealId]}.svg" style="width:80rpx;height:80rpx;vertical-align: middle;margin-right: 10rpx;" />

```

```

        {{ mealMap[mealId] }}

```

```

      </view>

```

```

      <text class="sub-title">{{ weight_sum }}克</text>

```

```

    </view>

```

```

    <view class="card-content" v-if="goodList?.length">

```

```

      <u-swipe-action

```

```

        :key="good.goods_no"

```

```

        v-for="(good, index) in goodList"

```

```

        :options="options"

```

```

        @open="openSwipeAction(index)"

```

```

        @click="()=>clickSwipeAction(index)"

```

```

        :show="good.show"

```

```
btnWidth="120"
>
<view class="goods-item" v-if="goodList?.length">
  <image class="image" :src="good.file_url || `${ossUrl}default_dish.png`" />
  <view class="goods-item-right" @click="handleShowEditPop(good)" >
    <view class="goods-name">
      <text class="twoline-hide">{{ good.dishes_name || '菜品名称'}}</text>
      <view class="item-prices oneline-hide">
        <text class="price-x">
          <text v-if="good.dishes_source == 3">{{ good.dishes_number || 0 }}{{ good.unit_name }}</text>
          <text v-else>{{ good.dishes_weight || 0 }}克</text>
        </text>
      </view>
    </view>
    <view class="desc-right" >
      <text class="price-x">{{ good.energy || 0 }}千卡</text>
      <image
        :src="`${ossUrl}static/right.svg`"
        style="width: 16rpx; height: 30rpx"
      />
    </view>
  </view>
  <empty v-else :tips="'亲，暂无分类'" />
</u-swipe-action>
</view>
<template v-else>
  <view class="empty">
    <image
      :src="`${ossUrl}empty.svg`"
      mode="scaleToFill"
      style="width: 100rpx; height: 100rpx; margin-bottom: 20rpx;"
    />
    <text>暂无数据</text>
  </view>
</template>
</view>
<view >
  <today-energy :todayEnergy="todayEnergy" :tipsList="tipsList" :suggestList="suggestList" />
</view>
<view class="logo">
</view>
<view class="static-btn">
  <button class="custom-button" @click="handleAddFood()">
    <image :src="`${ossUrl}select-add.svg`" class="custom-icon" mode="scaleToFill"/>
    选择添加</button>
  <button
    v-if="!isType8"
    class="custom-button"
    @click="handlePhone()"
```

```
      style="background-color: #f57440"
    >
    <image :src="'${ossUrl}phone-add.svg'" class="custom-icon" mode="scaleToFill"/>
      拍照添加
    </button>
  </view>

  <edit-food :value="showEditFoodModal" :goodWeight="goodWeight" @close-popup="handleClosePopup" :info="goodInfo"/>
</view>
</template>
<script>
import Empty from '@components/empty'
import EditFood from './edit-food.vue'
import * as MealApi from '@api/meal'
import config from '@config.js'
import TodayEnergy from './today-energy.vue'
import envConfig from '@env.js'

export default {
  components: {
    Empty,
    EditFood,
    TodayEnergy
  },
  data() {
    return {
      goodList: [
      ],
      goodWeight:0,
      goodInfo:{},
      elementList: [{}],
      options: [
        {
          text: "删除",
          style: {
            backgroundColor: "#FF3939",
          },
        },
      ],
      showEditFoodModal: false,
      currentGoodId:null,
      mealId:null,
      supply:{},
      eneryRemmond:{},
      energy:{},
      mealMap:{
        1:'早餐及早加餐',
        2:'午餐及午加餐',
        3:'晚餐及晚加餐',
      },
      weight_sum:0,
      statusMap:{
```

```

1:'down',
2:'normal',
3:'up',
},
imageMap:{
  1:'breakfast',
  2:'lunch',
  3:'dinner',
},
imagePath:"",
echartOption:{
  showBox:false,
  color:['#FF9A9A','#FFCD7E','#8AB4FF'],
  activeRadius:0,
  activeOpacity:0,
  rotateLock: false,
  dataLabel: false,
  enableScroll: false,
  legend: {
    show: false,
  },
  extra: {
    ring: {
      ringWidth: 8,
      activeOpacity: 0,
      offsetAngle: 0,
      labelWidth: 5,
    }
  }
},
chartData:{},
ossUrl:config.ossUrl,
todayEnergy:{},
suggestion_show:false,
suggestList:[],
tipsList:[],
isType8: false,
};
},
onLoad(options) {
  try {
    const cfg = (envConfig && typeof envConfig.getCurrentConfig === 'function') ? envConfig.getCurrentConfig() : null
    this.isType8 = Number(cfg?.type || 0) === 8
  } catch (err) {
    this.isType8 = false
  }
  const parseJSON=JSON.parse(options.toJSON || "{}");
  this.mealId = parseJSON.mealId;
  this.meal_date=parseJSON.selectDate;
},

```

```
onShow() {
  this.getMealDetail( this.mealId ,this.meal_date);
},
onShareAppMessage() {
  return {
    desc: '您的专属 AI 营养顾问',
    path: '/pages/index/index'
  }
},
onShareTimeline() {
  return {
    query: "
  }
},
methods: {
  async getMealDetail(id,date) {
    const meal_times=id
    const meal_date=date;
    await MealApi.mealInfo({
      meal_date,
      meal_times,
    }).then(res=>{
      this.goodList=res.data?.list;
      this.supply=res.data?.supply || {
        ratio:{
          carbohydrate:0,
          fat:0,
          protein:0
        },
        table:{
          carbohydrate:{ratio: 0,reality: "0",status: "2"},
          fat:{ratio:0,reality: "0",status: "2"},
          protein:{ratio: 0,reality: "0",status: "2"},
        }
      };
      this.energy=res.data?.energy || {};
      this.echartData=res.data?.energy?.reality || 0;
      this.weight_sum=res.data?.weight_sum || 0;
```

```
<template>
<view>
  <custom-navbar class="customnavbar" title="支付订单" :goback="goback" :showBack="true" :showHome="false" />
  <view v-if="!isLoading" class="container" :style="appThemeStyle">
    <view class="order-info">
      <view v-if="order.showExpiration" class="order-countdown">
        <text class="m-r-6">剩余时间</text>
        <count-down :date="order.expirationTime" separator="zh" theme="text" />
      </view>
      <view class="order-amount">
        <text class="unit">¥ </text>
        <text class="amount">{{ order.pay_price }}</text>
```

```

    </view>
  </view>
  <view class="payment-method">
    <view v-for="(item, index) in methods" :key="index" class="pay-item dis-flex flex-x-between" @click="handleSelectPayType(index)">
      <view class="item-left dis-flex flex-y-center">
        <view class="item-left_icon" :class="[item.method]">
          <text class="iconfont" :class="[PayMethodIconEnum[item.method]]"></text>
        </view>
        <view class="item-left_text">
          <text>{{ PayMethodEnum[item.method].name }}</text>
        </view>
        <view v-if="item.method === PayMethodEnum.BALANCE.value" class="user-balance">
          <text>(可用 ¥ {{ personal.balance }}元)</text>
        </view>
      </view>
      <view class="item-right col-m" v-if="curPaymentItem && curPaymentItem.method == item.method">
        <text class="iconfont icon-check"></text>
      </view>
    </view>
  </view>
  <view class="footer-fixed">
    <view class="btn-wrapper">
      <view class="btn-item btn-item-main" :class="{ disabled }" @click="handleSubmit()">确认支付</view>
    </view>
  </view>
  <u-modal v-if="tempUnifyData" v-model="showConfirmModal" title="支付确认" show-cancel-button confirm-text="已完成支付"
    :confirm-color="appTheme.mainBg" negative-top="100" :asyncClose="true"
    @confirm="onTradeQuery(tempUnifyData.outTradeNo, tempUnifyData.method)">
    <view class="modal-content">
      <text>请在{{ PayMethodClientNameEnum[tempUnifyData.method] }}内完成支付，如果您已经支付成功，请点击“已完成支付”按钮</text>
    </view>
  </u-modal>
</view>
</view>
</template>
<script>
import CustomNavbar from '@components/custom-navbar/index.vue';
import storage from '@utils/storage'
import { inArray, urlEncode } from '@utils/util'
import { Alipay, Wechat } from '@core/payment'
import Countdown from '@components/countdown'
import { PayMethodEnum } from '@common/enum/payment'
import { PayStatusEnum } from '@common/enum/order'
import * as CashierApi from '@api/cashier'
const PayMethodIconEnum = {
  [PayMethodEnum.WECHAT.value]: 'icon-wechat-pay',
  [PayMethodEnum.ALIPAY.value]: 'icon-alipay',
  [PayMethodEnum.BALANCE.value]: 'icon-balance-pay'
}
const PayMethodClientNameEnum = {

```

```

[PayMethodEnum.WECHAT.value]: '微信',
[PayMethodEnum.ALIPAY.value]: '支付宝'
}

const getTempUnifyData = orderKey => {
  const tempUnifyData = storage.get('tempUnifyData_' + orderKey)
  if (tempUnifyData) {
    storage.remove('tempUnifyData_' + orderKey)
    return tempUnifyData
  }
  return null
}

export default {
  components: {
    Countdown,
    CustomNavbar
  },
  data() {
    return {
      isLoading: true,
      disabled: false,
      PayMethodEnum,
      PayMethodIconEnum,
      PayMethodClientNameEnum,
      curPaymentItem: null,
      orderId: null,
      order: {},
      personal: { balance: '0.00' },
      methods: [],
      showConfirmModal: false,
      tempUnifyData: { outTradeNo: '', method: '' },
      goback: 1,
      returnUrl: 'pages/order/index'
    }
  },
  onLoad(options) {
    console.log(options);
    this.orderId = options.orderId
    if(options.goback){
      this.goback = options.goback
    }
    if(options.returnUrl){
      this.returnUrl = options.returnUrl
    }
  },
  onShow() {
    this.$nextTick(() => {
      this.getCashierInfo()
    })
  },
  onUnload() {

```

```

if(this.goback ==2){
  uni.switchTab({
    url: '/pages/order/index',
  });
}
},
methods: {
  getCashierInfo() {
    const app = this
    app.isLoading = true
    CashierApi.orderInfo(app.orderId, { client: app.platform })
    .then(result => {
      app.order = result.data.order
      app.personal = result.data.personal
      app.methods = result.data.paymentMethods
      app.isLoading = false
      app.setDefaultPayType()
      app.checkOrderPayStatus()
      this.performance()
    })
  },
  setDefaultPayType() {
    const app = this
    if (app.disabled) return
    if (app.curPaymentItem) return
    const defaultIndex = app.methods.findIndex(item => item.is_default == true)
    defaultIndex > -1 && app.handleSelectPayType(defaultIndex)
  },
  checkOrderPayStatus() {
    const app = this
    if (app.order.pay_status == PayStatusEnum.SUCCESS.value) {
      app.$toast('恭喜您，订单已付款成功')
      app.onSuccessNav()
    }
  },
  handleSelectPayType(index) {
    this.curPaymentItem = this.methods[index]
  },
  performance() {
    const app = this
    if (app.order.pay_status == PayStatusEnum.PENDING.value) {
      const performanceData = getTempUnifyData(app.orderId)
      if (performanceData) {
        app.tempUnifyData = performanceData
        app.showConfirmModal = true
      }
    }
  },
  handleSubmit() {
    const app = this

```

```

    if (!app.curPaymentItem) {
      app.$toast('您还没有选择支付方式')
      return
    }
    if (app.disabled) return
    app.disabled = true
    CashierApi.cashier_order_pay(app.orderId, {
      method: app.curPaymentItem.method,
      client: app.platform,
      extra: app.getExtraAsUnify(app.curPaymentItem.method)
    })
    .then(result =>
      app.onSubmitCallback(result)
    )
    .finally(err => setTimeout(() => app.disabled = false, 10))
  },
  getExtraAsUnify(method) {
    if (method === PayMethodEnum.ALIPAY.value) {
      return Alipay.extraAsUnify()
    }
    if (method === PayMethodEnum.WECHAT.value) {
      return Wechat.extraAsUnify()
    }
    return {}
  },
  onSubmitCallback(result) {
    const app = this
    const method = app.curPaymentItem.method
    const paymentData = result.data.payment
    if (method === PayMethodEnum.BALANCE.value) {
      app.onShowSuccess(result)
    }
    if (method === PayMethodEnum.ALIPAY.value) {
      console.log('paymentData', paymentData)
    }
  }
}
<template>
  <view class="container">
    <view class="address-card">
      <view class="address-selector">
        <text class="address-text">{{ quer_url.name}}</text>
      </view>
    </view>
    <commodity ref="mescrollItem" :list="list" :quer_url="quer_url" :setting="setting" :is_order="is_order" @cartUpdate="handleCartUpdate" @showDetail="showItemDetail" />
    <view class="goods-carts" :style="{ zIndex: 99999 }">
      <view class="add-meal">
        <view class="image-with-badge" @click="handleAddDetail">
          <image
            cartTotalCount > 0
            ? 'shop'
            : 'shop_disabled'
          >
            .svg
          </image>
        </view>
      </view>
    </view>
  </view>

```

```
        style="width: 80rpx; height: 80rpx"
    />
</u-badge>
</view>
<view class="meal-title"><view class="cash_labe">¥ </view>{{ cartTotalAmount }}</view>
</view>
<u-button shape="circle" size="large" type="primary" @click="handleSubmit('add')
:custom-style="{
padding:'24rpx 118rpx',
fontSize: '30rpx',
height: '86rpx',
background: 'linear-gradient(90deg, #389EF7 0%, #1A65ED 100%)',
border: 'none'
}"
:disabled="cartTotalCount === 0"
>去下单</u-button>
</view>
<u-popup
v-model="foodVisiable"
mode="bottom"
length="60%"
border-radius="16"
:safe-area-inset-bottom="true"
>
<view class="goods-sum">
<view class="goods-sum-title">
<view class="goods-sum-title-text">已选择菜品</view>
<view class="clear-cart" @click="clearCart">
<uni-icons type="trash" size="20" color="#666666"></uni-icons>
清空
</view>
</view>
<scroll-view scroll-y="true" style="height: 50%" class="cart-item-scroll">
<view v-if="cartItems.length === 0" class="empty-cart">
<text>购物车为空</text>
</view>
<view v-else class="cart-item" v-for="(item, index) in cartItems" :key="item.ai_dishes_id">
<image :src="item.displaySrc || item.goods_image" class="cart-item-image" @error="onCartItemError(index)" />
<view class="cart-item-info">
<text class="cart-item-name">{{ item.goods_name }}</text>
<text class="cart-item-kq">{{item.energy}}千卡 / {{ item.weight }}克</text>
<text v-if="hasBookableLimit(item)" class="cart-item-limit">可订{{ getItemBookableNum(item) }}份</text>
<view class="cart-item-price"><view class="cash_labe">¥ </view>{{ item.goods_price_min}}</view>
</view>
```

```
<view class="cart-item-controls">
  <view class="quantity-btn" @click.stop="decreaseCartItem(item)">
    <uni-icons type="minus" size="26" color="#1a65ed"></uni-icons>
  </view>
  <text class="quantity-text">{{ item.quantity }}</text>
  <view class="quantity-btn" :class="{ disabled: isIncreaseDisabled(item) }" @click.stop="increaseCartItem(item)">
    <uni-icons type="plus-filled" size="26" :color="isIncreaseDisabled(item) ? '#c7cfdd' : '#1a65ed'"></uni-icons>
  </view>
</view>
</u-popup>
</scroll-view>
</view>
</u-popup>
<u-popup
  v-model="showDetailPopup"
  mode="bottom"
  length="55%"
  border-radius="16"
  :safe-area-inset-bottom="true"
>
  <view class="detail-popup" v-if="selectedItem">
    <view class="details-box">
      <image :src="selectedItem.displaySrc || selectedItem.file_path" class="detail-hero" @error="onDetailImgError" />
      <view class="details-r-info">
        <view class="detail-row">
          <view class="detail-name">{{ selectedItem.name }}</view>
        </view>
        <view class="detail-meta">
          <text class="detail-sub">{{ selectedItem.energy }} 千卡 / {{ selectedItem.weight }} 克</text>
          <view v-if="hasBookableLimit(selectedItem)" class="detail-limit">可订 {{ getItemBookableNum(selectedItem) }} 份</view>
        </view>
        <view class="detail-action-row">
          <view class="detail-price"><view class="cash_label">¥ </view>{{selectedItem.price }}</view>
          <view class="detail-qty" v-if="is_order == 1">
            <view class="quantity-btn" @click.stop="decreaseDetailItem(selectedItem)">
              <uni-icons type="minus" size="26" color="#1a65ed"></uni-icons>
            </view>
            <text class="quantity-text">{{ getItemQuantity(selectedItem.ai_dishes_id) }}</text>
            <view class="quantity-btn" :class="{ disabled: isIncreaseDisabled(selectedItem) }" @click.stop="increaseDetailItem(selectedItem)">
              <uni-icons type="plus-filled" size="26" :color="isIncreaseDisabled(selectedItem) ? '#c7cfdd' : '#1a65ed'"></uni-icons>
            </view>
          </view>
        </view>
      </view>
      <view class="detail-body">
        <view class="label-tag">
          <u-tag
            :text="tag.name"
            v-for="tag in selectedItem.tag"
          >
```

```

      :key="tag.id"
      shape="circle"
      class="tag tag-size"
    />
  </view>
<view class="nutrients">
  <view class="nutrients-center">
    <view class="nutrients-name">能量(千卡)</view>
    <view>{{ selectedItem.energy }}</view>
  </view>
  <view class="nutrients-center">
    <view class="nutrients-name">碳水(克)</view>
    <view>{{ selectedItem.carbohydrate }}</view>
  </view>
  <view class="nutrients-center">
    <view class="nutrients-name">蛋白质(克)</view>
    <view>{{ selectedItem.protein }}</view>
  </view>
  <view class="nutrients-center">
    <view class="nutrients-name">脂肪(克)</view>
    <view>{{ selectedItem.fat }}</view>
  </view>
</view>
</view>
</u-popup>
</view>
</template>
<script>
import MescrollCompMixin from '@uni_modules/mescroll-uni/components/mescroll-uni/mixins/mescroll-comp'
import { PageCategoryStyleEnum } from '@common/enum/store/page/category'
import * as CategoryApi from '@api/category'
import Commodity from './components/commodity'
export default {
  components: {
    Commodity
  },
  mixins: [MescrollCompMixin],
  data() {
    return {
      PageCategoryStyleEnum,
      list: [],
      setting: {},
      isLoading: true,
      badge: 0,
      foodVisiable: false,
      cartItems: [],
      cartTotalCount: 0,
      cartTotalAmount: 0,
      showDetailPopup: false,

```

```

selectedItem: null,

quer_url:{
  restaurant_id: "",
  meal_date:"",
  meal_times: "",
  name:""
},
is_order: 1
}
},
onLoad(options) {
  if(options){
    this.quer_url.restaurant_id = options.restaurant_id
    this.quer_url.meal_date = options.meal_date
    this.quer_url.meal_times = options.meal_times
    this.quer_url.name = options.restaurant_name
    this.is_order = options.is_order
  }
},
onShow() {
  this.onRefreshPage()
},
methods: {

<template>
<view class="container">
  <custom-navbar title="确认下单" :showBack="true" :showHome="false" />
  <view class="address-card">
    <view class="address-selector">
      <text class="address-text">{{ restaurant_name }}</text>
    </view>
  </view>
  <view class="card-theme">
    <view class="card card-top">
      <view class="card-title">取餐信息</view>
      <view class="cell">
        <text class="cell-label">姓名</text>
        <text class="cell-value">{{ user_name }}</text>
      </view>
      <view class="cell">
        <text class="cell-label">手机号</text>
        <view class="cell-input" style="align-items: center;">
          <input class="phone-input" type="number" v-model.trim="phone" placeholder="请输入手机号" />
          <uni-icons type="right" size="15" style="margin-right: -10rpx;"></uni-icons>
        </view>
      </view>
      <view class="cell">
        <text class="cell-label">取餐时间</text>
        <picker mode="selector" :range="takeTimeOptions" :value="takeTimeIndex" @change="onTakeTimeChange">
          <view class="cell-select">
            <text> {{ currentTakeTime || '请选择' }}</text>
          </view>
        </picker>
      </view>
    </view>
  </view>

```

```

        <uni-icons type="down" size="16" color="#999"></uni-icons>
    </view>
</picker>
</view>
<view class="cell">
    <text class="cell-label">取餐方式</text>
    <picker
        v-if="take_methods.length"
        mode="selector"
        :range="takeMethodOptions"
        :value="takeMethodIndex"
        @change="onTakeMethodChange"
    >
        <view class="cell-select">
            <text>{{ currentTakeMethodName || '请选择' }}</text>
            <uni-icons type="down" size="16" color="#999"></uni-icons>
        </view>
    </picker>
    <text v-else class="cell-value">{{ currentTakeMethodName }}</text>
</view>
<view class="cell" v-if="isLockerMethod">
    <text class="cell-label">取餐柜</text>
    <picker
        v-if="lockerOptions.length"
        mode="selector"
        :range="lockerOptions"
        :value="lockerIndex"
        @change="onLockerChange"
    >
        <view class="cell-select">
            <text>{{ currentLockerName || '请选择' }}</text>
            <uni-icons type="down" size="16" color="#999"></uni-icons>
        </view>
    </picker>
    <text v-else class="cell-value">暂无可用取餐柜</text>
</view>
<view class="cell">
    <text class="cell-label">{{ isLockerMethod ? '取餐柜地址' : '自提地址' }}</text>
    <text class="cell-value">{{ isLockerMethod ? currentLockerAddress : restaurant_location }}</text>
</view>
</view>
<view class="card card-bot card-top">
    <view class="card-title">订单明细</view>
    <view class="order-item" v-for="(it, idx) in items" :key="idx">
        <image class="item-img" :src="it.file_path" mode="aspectFill" @error="onItemImgError(idx)" />
        <view class="item-info">
            <view class="item-row">
                <text class="item-name">{{ it.name }}</text>
                <view class="item-price"><view class="cash_labe">¥</view>{{ it.price }}</view>
            </view>
        </view>
    </view>

```

```
<view class="item-row">
  <text class="item-qty">x{{ it.quantity }}</text>
  <text class="item-price"></text>
</view>
</view>
</view>
<view class="order-item fee-item" v-if="showPackageFee">
  <image class="item-img" :src="currentPackageFeeItem.file_path" mode="aspectFill" @error="onPackageFeeImgError" />
  <view class="item-info">
    <view class="item-row">
      <text class="item-name">{{ currentPackageFeeItem.name }}</text>
      <view class="item-price"><view class="cash_labe">¥</view>{{ currentPackageFeeItem.price }}</view>
    </view>
    <view class="item-row">
      <text class="item-qty">x1</text>
      <text class="item-price"></text>
    </view>
  </view>
</view>
</view>
<view class="order-total">
  <text></text>
  <view class="order-sum">
    <text class="order-text">合计</text>
    <view class="total"><view class="cash_labe">¥</view>{{ displayTotalPrice }}</view>
  </view>
</view>
</view>
<view class="card card-top" @click="openRemark">
  <view class="card-title" style="border-bottom: none;">
    <view style="display: flex; justify-content: space-between; width: 100%;">
      <view class="remark-label">备注</view>
      <view class="remark-content">
        <view class="remark-placeholder" v-if="!remark">口味、偏好等要求</view>
        <view class="remark-text" v-else>{{ remark }}</view>
        <uni-icons type="right" size="15"></uni-icons>
      </view>
    </view>
  </view>
</view>
</view>
<view class="footer-bar">
  <view class="footer-price"><view class="cash_labe">¥</view>{{ displayTotalPrice }}</view>
  <view class="footer-btn" @click="submitOrder">提交订单</view>
</view>
</view>
<u-popup v-model="remarkVisible" width="90%" mode="center" length="40%" border-radius="16" :safe-area-inset-bottom="true">
  <view class="remark-popup">
    <view class="popup-header">
      <text class="popup-title">订单备注</text>
    </view>
    <textarea class="popup-textarea" cursor-spacing="60" v-model="remarkInput" placeholder="请输入备注" />
  </view>
</u-popup>
```

```

<view class="popup-actions">
  <view class="btn confirm" @click="confirmRemark">确定</view>
</view>
</view>
</u-popup>
</view>
</template>
<script>
import CustomNavbar from '@components/custom-navbar/index.vue';
import * as CategoryApi from '@api/category'
export default {
  name: "",
  components: {
    CustomNavbar,
  },
  computed: {
    currentTakeTime() {
      return this.formatTakeTime(this.takeTimeList[this.takeTimeIndex])
    },
    takeTimeOptions() {
      if (!Array.isArray(this.takeTimeList)) return []
      return this.takeTimeList.map(t => this.formatTakeTime(t))
    },
    takeMethodOptions() {
      if (!Array.isArray(this.take_methods)) return []
      return this.take_methods.map(item => item.name)
    },
    takeMethodIndex() {
      if (!Array.isArray(this.take_methods)) return 0
      const idx = this.take_methods.findIndex(item => Number(item.type) === Number(this.restaurant_method))
      return idx >= 0 ? idx : 0
    },
    currentTakeMethod() {
      if (!Array.isArray(this.take_methods) || !this.take_methods.length) return null
      const method = this.take_methods.find(item => Number(item.type) === Number(this.restaurant_method))
      return method || this.take_methods[0]
    },
    currentTakeMethodName() {
      if (this.currentTakeMethod && this.currentTakeMethod.name) return this.currentTakeMethod.name
      return Number(this.restaurant_method) === 1 ? '堂食' : '其他'
    },
    isLockerMethod() {
      return Number(this.currentTakeMethod && this.currentTakeMethod.type) === 4
    },
    lockerList() {
      const list = this.currentTakeMethod && this.currentTakeMethod.lockers
      return Array.isArray(list) ? list : []
    },
    lockerOptions() {
      return this.lockerList.map(item => item.name || item.code || item.sn || '取餐柜')
    }
  }
}

```

```

},
lockerIndex() {
  if (!this.lockerList.length) return 0
  const idx = this.lockerList.findIndex(item => String(item.code) === String(this.locker_code))
  return idx >= 0 ? idx : 0
},
currentLocker() {
  if (!this.lockerList.length) return null
  return this.lockerList[this.lockerIndex] || this.lockerList[0]
},
currentLockerName() {
  return this.currentLocker ? (this.currentLocker.name || this.currentLocker.code || this.currentLocker.sn) : ""
},
currentLockerAddress() {
  return this.currentLocker ? (this.currentLocker.address || "") : ""
},
currentPackageFeeItem() {
  return (this.currentTakeMethod && this.currentTakeMethod.package_fee_item) || null
},
showPackageFee() {
  return [2, 4].includes(Number(this.currentTakeMethod && this.currentTakeMethod.type)) && !!this.currentPackageFeeItem
},
displayTotalPrice() {
  const basePrice = Number(this.total_price) || 0
  const packageFee = this.showPackageFee ? (Number(this.currentPackageFeeItem.price) || 0) : 0
  return (basePrice + packageFee).toFixed(2)
}
},
data() {
  return {
    restaurant_name: "",
    user_name: "",
    phone: "",
    pickupTime: "",
    takeTimeList: [],
    takeTimeIndex: 0,
    items: [],
    take_methods: [],
    remark: "",
    remarkVisible: false,
    remarkInput: "",
    restaurant_method:"",
    locker_code: "",
    restaurant_location:"",
    total_price: '0.00',
    quer_url:{
      restaurant_id: "",
      meal_date:"",
      meal_times: "",
      name:"

```

```

    },
    is_order: 1
  },
  },
  onLoad(options) {
    this.quer_url.restaurant_id = options.restaurant_id
    this.quer_url.meal_date = options.meal_date
    this.quer_url.meal_times = options.meal_times
    this.quer_url.name = options.restaurant_name
    this.is_order = options.is_order
    this.quer_url.stall_type=options.stall_type
    this.quer_url.peitong_num=options.peitong_num
    this.quer_url.zhaodai_num=options.zhaodai_num
    CategoryApi.meal_checkout_order({
      restaurant_id:this.quer_url.restaurant_id,
      meal_date:this.quer_url.meal_date,
      meal_times:this.quer_url.meal_times,
    })
  }
}
</script>
<template>
  <a-card :bordered="false">
    <div class="table-operator">
      <a-row>
        <a-col :span="6">
          <a-button type="primary" icon="plus" @click="handleAdd">
            >新增菜品</a-button>
          </a-col>
          <a-col :span="8" :offset="10">
            <a-input-search
              style="max-width: 300px; min-width: 150px; float: right;"
              v-model="queryParam.search"
              placeholder="请输入菜品名称"
              @search="onSearch"
            />
          </a-col>
        </a-row>
      </div>
      <s-table
        ref="table"
        rowKey="id"
        :loading="isLoading"
        :columns="columns"
        :data="loadData"
        :pageSize="15"
        :scroll="{ x: 1450 }"
      >
        <span slot="dish_image" slot-scope="text">
          <a title="点击查看原图" :href="text" target="_blank" v-if="text">
            
          </a>
          
        </span>
      </s-table>
    </div>
  </a-card>
</template>

```

```
</span>

<div class="actions" slot="action" slot-scope="text, item">
  <a @click="handleEdit(item)">编辑</a>
  <a @click="handleLook(item)">查看</a>
  <template>
    <a @click="handleDelete(item.id)">删除</a>
  </template>
</div>
</s-table>
</a-card>
</template>
<script>
import * as Api from "@api/menu/index";
import { STable } from "@components";
export default {
  name: "food",
  components: {
    STable,
  },
  data() {
    return {
      queryParams: {},
      isLoading: false,
      columns: [
        {
          title: "菜品 ID",
          dataIndex: "id",
        },
        {
          title: "菜品图片",
          dataIndex: "file_url",
          scopedSlots: { customRender: "dish_image" },
        },
        {
          title: "菜品名称",
          dataIndex: "name",
          width: "150px",
          ellipsis: true,
          customRender: (text) => (
            <a-tooltip title={text}>
              <span class="ellipsis">{text}</span>
            </a-tooltip>
          ),
        },
        {
          title: "菜品分类",
          dataIndex: "category_name",
          width: "150px",
        },
      ],
    };
  },
  methods: {
    handleEdit(item) {
      this.$router.push({ path: "/menu/edit/" + item.id });
    },
    handleLook(item) {
      this.$router.push({ path: "/menu/look/" + item.id });
    },
    handleDelete(id) {
      this.$confirm("确定删除吗?", "提示", {
        type: "warning",
      }).then(() => {
        Api.deleteMenu({ id }).then(() => {
          this.$message.success("删除成功");
          this.refresh();
        });
      });
    },
  },
};
</script>
```

```

        title: "单位重量(g)",
        dataIndex: "weight",
        width: "100px",
    },
    {
        title: "能量(kcal)",
        dataIndex: "energy",
        width: "100px",
    },
    {
        title: "碳水化合物(g)",
        dataIndex: "carbohydrate",
        width: "100px",
    },
    {
        title: "蛋白质(g)",
        dataIndex: "protein",
        width: "100px",
    },
    {
        title: "脂肪(g)",
        dataIndex: "fat",
        width: "100px",
    },
    {
        title: "创建时间",
        dataIndex: "create_time",
        width: "200px",
    },
    {
        title: "创建人",
        dataIndex: "create_by",
        width: "150px",
    },
    {
        title: "操作",
        dataIndex: "action",
        width: "180px",
        fixed: "right",
        scopedSlots: { customRender: "action" },
    },
],
loadData: (param) => {
    return Api.list({
        ...param,
        ...this.queryParam,
    }).then((response) => {
        return response.data.list;
    });
},

```