

```
<template>

<div class="ai-dish-page" v-loading="loading">

  <div class="toolbar queryArea">

    <div class="selectInput" style="margin-right: 20px;">

      <el-input

        v-model.trim="searchName"

        class="search-input"

        clearable

        placeholder="请输入商品名称"

        @change="handleSearch"

        @clear="handleSearch"

        @keyup.enter.native="handleSearch">

        <el-button

          @click="handleSearch()"

          icon="el-icon-search"

          style="border-bottom: 0"

          slot="append">

        </el-button>

      </el-input>

    </div>

    <div class="toolbar-actions">

      <el-button

        v-if="batchMode"

        plain

        @click="cancelBatchMode"

      >

        取消

      </el-button>

      <el-button

        type="danger"

        @click="handleBatchButton"

      >

        {{ batchMode ? '删除选中' : '批量删除' }}

      </el-button>

    </div>

  </div>

  <div v-if="dishList.length" class="dish-grid">

    <div

      v-for="item in dishList"

      :key="item.id"

      class="dish-card"

      @click="openDetailDialog(item)"

    >

      <div class="dish-card__content">

        <div class="dish-card__header">

          <div class="dish-main">

            <div class="dish-name textHidden">{{ item.name }}</div>

            <div class="dish-meta">

              <span class="price">{{ formatAmount(item.amount) }}</span>

              <span class="divider"></span>

            </div>

          </div>

        </div>

      </div>

    </div>

  </div>

</div>
```

```
      <span class="portion">份</span>
    </div>
  </div>
</el-checkbox>
: value="isSelected(item.id)"
: disabled="!batchMode"
@ click.native.stop
@ change="toggleSelect(item.id, $event)"
/>
</div>
</div>
<div class="image-strip-wrap">
  <div class="image-strip">
    <div
      v-for="(image, imageIndex) in item.images"
      :key="image.id"
      class="dish-image-item"
    >
      <el-image
        :src="image.url"
        fit="cover"
        class="dish-image"
        @click.native.stop="openImagePreview(image.url)"
      >
        <div slot="error" class="image-slot">
          <i class="el-icon-picture-outline"></i>
        </div>
      </el-image>
      <i
        class="el-icon-close delete-image-btn"
        @click.stop="confirmDeleteImage(image.id)"
      />
    </div>
  </div>
</div>
</div>
</div>
</div>
<div v-else class="empty-wrap">
  <div class="empty-title">暂无数据</div>
  <div class="empty-desc">请调整搜索条件后重试</div>
</div>
<div class="pagination-wrap">
  <el-pagination
    @current-change="handleCurrentChangeFun"
    :hide-on-single-page="false"
    :current-page="currentPage"
    :page-size="pageSize"
    layout="total, jumper, ->, prev, pager, next"
    :total="tableTotal"
  ></el-pagination>
```

```
</div>

<el-dialog
  title="明细"
  :visible.sync="detailVisible"
  width="920px"
  custom-class="dish-detail-dialog"
  :close-on-click-modal="false"
>

  <div v-if="currentDish" class="detail-dialog">

    <div class="detail-dialog__top">

      <div class="detail-main">

        <div class="detail-title-row">

          <div class="detail-title">{{ currentDish.name }}</div>

          <span class="detail-tag">识别商品</span>

        </div>

        <div class="detail-meta-row">

          <span class="detail-price">{{ formatDetailAmount(currentDish.amount) }}</span>

          <span class="detail-count">学习了{{ currentDish.images.length }}张</span>

        </div>

      </div>

      <el-button
        type="danger"
        class="remove-all-btn"
        @click="confirmDeleteAllImages"
      >
        删除学习数据
      </el-button>

    </div>

    <div class="detail-dialog__content">

      <div v-if="currentDish.images.length" class="detail-image-grid">

        <div
          v-for="(image, imageIndex) in currentDish.images"
          :key="image.id"
          class="detail-image-item"
        >

          <el-image
            :src="image.url"
            fit="cover"
            class="detail-image"
            @click.native="openImagePreview(image.url)"
          >

            <div slot="error" class="image-slot">

              <i class="el-icon-picture-outline"></i>

            </div>

          </el-image>

          <i
            class="el-icon-close detail-delete-btn"
            @click.stop="confirmDeleteImage(image.id)"
          />

        </div>

      </div>

    </div>

  </div>

</div>
```

```
    </div>

    <div v-else class="detail-empty">暂无学习图片</div>
  </div>
</div>
</el-dialog>
<el-dialog
  :visible.sync="previewVisible"
  width="520px"
  custom-class="image-preview-dialog"
  :close-on-click-modal="true"
  append-to-body
>
  <div class="image-preview-wrap">
    <el-image
      v-if="previewImage"
      :src="previewImage"
      fit="contain"
      class="image-preview"
    >
      <div slot="error" class="image-slot">
        <i class="el-icon-picture-outline"></i>
      </div>
    </el-image>
  </div>
</el-dialog>
</div>
</template>
<script>
export default {
  name: "aidish",
  data() {
    return {
      searchName: "",
      keyword: "",
      batchMode: false,
      selectedIds: [],
      currentPage: 1,
      pageSize: 12,
      tableTotal: 0,
      dishList: [],
      detailVisible: false,
      currentDishId: null,
      previewVisible: false,
      previewImage: "",
      loading: false,
    };
  },
  computed: {
    currentDish() {
      return this.dishList.find((item) => item.id === this.currentDishId) || null;
    }
  }
}
```

```
    },
  },
  mounted() {
    this.getData();
  },
  methods: {
    normalizeDish(item) {
      const images = (item.images || []).map((image) => ({
        id: image.id,
        dishes_uuid: image.dishes_uuid,
        url: image.img_url,
        img_url: image.img_url,
        feature: image.feature || "",
        equipment_code: image.equipment_code || "",
        create_time: image.create_time || "",
      }));
      return {
        id: item.dishes_uuid,
        dishes_uuid: item.dishes_uuid,
        name: item.dishes_name || "--",
        content: "识别商品图片数据",
        amount: item.price,
        portion: item.image_count ? item.image_count + "张" : "0 张",
        code: images.length ? images[0].equipment_code || "--" : "--",
        image_count: item.image_count || images.length || 0,
        images,
      };
    },
    formatAmount(value) {
      if (value === null || value === undefined || value === "") {
        return "--";
      }
      return "¥" + Number(value || 0).toFixed(1);
    },
    formatDetailAmount(value) {
      if (value === null || value === undefined || value === "") {
        return "--";
      }
      const number = Number(value || 0);
      return "¥" + (Number.isInteger(number) ? number : number.toFixed(1));
    },
    getData() {
      this.loading = true;
      this.$axios.post("/p/bill/dishesRecognizeImgList", this.$qs({
        page: this.currentPage,
        page_size: this.pageSize,
        keyword: this.keyword,
      })).then((res) => {
        if (res.code === 0) {
          const data = res.data || {};
        }
      });
    }
  }
}
```

```
this.dishList = (data.data || []).map(this.normalizeDish);
this.tableTotal = Number(data.total || 0);
if (!this.dishList.length && this.tableTotal > 0 && this.currentPage > 1) {
  this.currentPage = this.currentPage - 1;
  this.getData();
  return;
}
if (this.currentDishId && !this.currentDish) {
  this.detailVisible = false;
  this.currentDishId = null;
}
return;
}
this.$message.error(res.message || "获取列表失败");
}).catch(() => {
  this.$message.error("获取列表失败");
}).finally(() => {
  this.loading = false;
});
},
handleSearch() {
  this.keyword = this.searchName;
  this.currentPage = 1;
  this.clearSelection();
  this.getData();
},
handleCurrentChangeFun(val) {
  this.currentPage = val;
  this.getData();
},
handleBatchButton() {
  if (!this.batchMode) {
    this.batchMode = true;
    this.clearSelection();
    return;
  }
  if (!this.selectedIds.length) {
    this.$message.warning("请选择需要删除的菜品");
    return;
  }
  this.$confirm("确认删除选中的菜品吗？", "删除提醒", {
    confirmButtonText: "确定",
    cancelButtonText: "取消",
    type: "warning",
    center: true,
  }).then(() => {
    const imageIds = this.dishList
      .filter((item) => this.selectedIds.indexOf(item.id) > -1)
      .reduce((result, item) => {
        return result.concat((item.images || []).map((image) => image.id));
      }, []);
```

```
    }, []);
    if (!imageIds.length) {
      this.$message.warning("所选菜品暂无可删除图片");
      return;
    }
    this.deleteImagesByIds(imageIds, () => {
      this.batchMode = false;
      this.clearSelection();
    });
  }).catch(() => {});
},
cancelBatchMode() {
  this.batchMode = false;
  this.clearSelection();
},
toggleSelect(id, checked) {
  if (!this.batchMode) {
    return;
  }
  if (checked) {
    if (this.selectedIds.indexOf(id) === -1) {
      this.selectedIds.push(id);
    }
  } else {
    this.selectedIds = this.selectedIds.filter((item) => item !== id);
  }
},
isSelected(id) {
  return this.selectedIds.indexOf(id) > -1;
},
clearSelection() {
  this.selectedIds = [];
},
openDetailDialog(item) {
  this.currentDishId = item.id;
  this.detailVisible = true;
},
openImagePreview(url) {
  this.previewImage = url;
  this.previewVisible = true;
},
confirmDeleteImage(imageId) {
  if (!imageId) {
    this.$message.warning("图片数据不存在");
    return;
  }
  this.$confirm("确认删除这张学习图片吗？", "删除提醒", {
    confirmButtonText: "确定",
    cancelButtonText: "取消",
    type: "warning",
```

```

        center: true,
    }).then(() => {
        this.deleteImagesByIds([imageId]);
    }).catch(() => {});
},
confirmDeleteAllImages() {
    if (!this.currentDish || !this.currentDish.images.length) {
        this.$message.warning("当前没有可删除的学习图片");
        return;
    }
    this.$confirm("确认删除该菜品全部学习图片吗？", "删除提醒", {
        confirmButtonText: "确定",
        cancelButtonText: "取消",
        type: "warning",
        center: true,
    }).then(() => {
        this.deleteImagesByIds(this.currentDish.images.map((image) => image.id));
    }).catch(() => {});
},
deleteImagesByIds(ids, callback) {
    const params = new URLSearchParams();
    ids.forEach((id) => {
        params.append("ids[]", id);
    });
    this.loading = true;
    this.$axios.post("/p/bill/dishesRecognizeImgDelete", params).then((res) => {
        if (res.code === 0) {
            this.$message.success("删除成功");
            if (typeof callback === "function") {
                callback();
            }
            this.getData();
            return;
        }
        this.$message.error(res.message || "删除失败");
    }).catch(() => {
        this.$message.error("删除失败");
    }).finally(() => {
        this.loading = false;
    });
},
},
};
</script>
<style lang="scss" scoped>
.ai-dish-page {
    min-height: 100%;
    padding: 16px;
    background: #f7f7f7;
}

```

```
.toolbar {
  display: flex;
  align-items: center;
  justify-content: space-between;
  margin-bottom: 16px;
  gap: 16px;
}

.search-input {
  width: 240px;
}

.toolbar-actions {
  display: flex;
  align-items: center;
  gap: 12px;
}

.dish-grid {
  display: grid;
  grid-template-columns: repeat(4, minmax(0, 1fr));
  gap: 12px;
}

.dish-card {
  padding: 14px 14px 16px;
  background: #ffffff;
  border-radius: 8px;
  box-shadow: 0 2px 10px rgba(32, 35, 41, 0.05);
}

.dish-card__content {
  cursor: pointer;
}

.dish-card__header {
  display: flex;
  align-items: flex-start;
  justify-content: space-between;
  margin-bottom: 14px;
}

.dish-main,
.detail-main {
  flex: 1;
  min-width: 0;
  padding-right: 12px;
}

.dish-name {
  margin-bottom: 8px;
  font-size: 18px;
  line-height: 25px;
  color: #333333;
  font-weight: 600;
}

.dish-desc {
  margin-bottom: 10px;
```

```
font-size: 13px;
line-height: 18px;
color: #8c8c8c;
}

.dish-meta {
font-size: 16px;
line-height: 22px;
}

.price {
color: #ff5a36;
font-weight: 600;
}

.divider,
.portion {
color: #9d9d9d;
}

.image-strip-wrap {
max-width: 100%;
overflow-x: auto;
overflow-y: hidden;
padding-bottom: 4px;
}

.image-strip {
display: flex;
align-items: center;
gap: 10px;
min-width: max-content;
}

.dish-image-item {
position: relative;
flex: 0 0 auto;
width: 54px;
height: 54px;
padding-top: 8px;
}

.dish-image {
width: 54px;
height: 54px;
border-radius: 10px;
overflow: hidden;
background: #f2f2f2;
cursor: pointer;
}

.image-slot {
width: 100%;
height: 100%;
display: flex;
align-items: center;
justify-content: center;
color: #c0c4cc;
```

```
font-size: 18px;
}

.delete-image-btn {
  position: absolute;
  top: 0;
  right: -6px;
  width: 18px;
  height: 18px;
  border-radius: 50%;
  background: rgba(0, 0, 0, 0.75);
  color: #ffffff;
  font-size: 12px;
  line-height: 18px;
  text-align: center;
  cursor: pointer;
}

.delete-image-btn:hover,
.delete-image-btn:focus {
  background: rgba(0, 0, 0, 0.85);
  color: #ffffff;
}

.empty-wrap {
  padding: 80px 20px;
  text-align: center;
  background: #ffffff;
  border-radius: 8px;
  color: #999999;
}

.empty-title {
  margin-bottom: 8px;
  font-size: 18px;
  color: #666666;
}

.empty-desc {
  font-size: 13px;
}

.pagination-wrap {
  margin-top: 20px;
}

.detail-dialog {
  min-height: 420px;
}

.detail-dialog__top {
  display: flex;
  align-items: flex-start;
  justify-content: space-between;
  gap: 20px;
  margin-bottom: 10px;
}

.detail-title-row {
```

```
display: flex;
align-items: center;
gap: 14px;
margin-bottom: 18px;
}
.detail-title {
font-size: 18px;
line-height: 25px;
font-weight: 600;
color: #333333;
}
.detail-tag {
font-size: 14px;
color: #f6b23e;
}
.detail-meta-row {
display: flex;
align-items: center;
flex-wrap: wrap;
gap: 15px;
color: #b0b0b0;
font-size: 14px;
}
.detail-price {
color: #ff5a36;
font-size: 20px;
font-weight: 600;
}
.remove-all-btn {
flex: 0 0 auto;
}
.detail-dialog__content {
max-height: 420px;
padding: 10px 4px 0 0 ;
box-sizing: border-box;
overflow: hidden;
overflow-y: auto;
}
.detail-image-grid {
display: grid;
grid-template-columns: repeat(4, minmax(0, 1fr));
gap: 18px;
}
.detail-image-item {
position: relative;
width: 100%;
padding-top: 100%;
}
.detail-image {
position: absolute;
```

```
top: 0;
left: 0;
width: 100%;
height: 100%;
border-radius: 6px;
overflow: hidden;
cursor: pointer;
background: #f2f2f2;
}

.detail-delete-btn {
  position: absolute;
  top: -8px;
  right: -8px;
  width: 25px;
  height: 25px;
  border-radius: 50%;
  background: rgba(0, 0, 0, 0.75);
  color: #ffffff;
  font-size: 14px;
  line-height: 25px;
  text-align: center;
  cursor: pointer;
}

.detail-delete-btn:hover,
.detail-delete-btn:focus {
  background: rgba(0, 0, 0, 0.85);
  color: #ffffff;
}

.detail-empty {
  height: 220px;
  display: flex;
  align-items: center;
  justify-content: center;
  color: #999999;
  background: #fafafa;
  border-radius: 8px;
}

.image-preview-wrap {
  display: flex;
  align-items: center;
  justify-content: center;
  min-height: 420px;
}

.image-preview {
  width: 100%;
  height: 420px;
}

/deep/ .dish-card__header .el-checkbox {
  line-height: 1;
}
```

```
/deep/ .dish-card__header .el-checkbox .el-checkbox__inner {
  width: 16px;
  height: 16px;
}

/deep/ .dish-card__header .el-checkbox.is-disabled .el-checkbox__inner {
  background: #ffffff;
  border-color: #dcdfe6;
}

/deep/ .dish-card__header .el-checkbox__input.is-disabled + .el-checkbox__label {
  color: #c0c4cc;
}

/deep/ .dish-detail-dialog .el-dialog__body {
  padding-top: 22px;
}

/deep/ .image-preview-dialog .el-dialog__body {
  padding-top: 10px;
}

@media (max-width: 1400px) {
  .dish-grid {
    grid-template-columns: repeat(3, minmax(0, 1fr));
  }

  .detail-image-grid {
    grid-template-columns: repeat(3, minmax(0, 1fr));
  }
}

@media (max-width: 1100px) {
  .dish-grid {
    grid-template-columns: repeat(2, minmax(0, 1fr));
  }

  .detail-image-grid {
    grid-template-columns: repeat(2, minmax(0, 1fr));
  }
}

@media (max-width: 768px) {
  .ai-dish-page {
    padding: 12px;
  }

  .toolbar {
    flex-direction: column;
    align-items: stretch;
  }

  .search-input {
    width: 100%;
  }

  .toolbar-actions {
    justify-content: flex-end;
  }

  .dish-grid {
    grid-template-columns: 1fr;
  }
}
```

```
.detail-dialog__top {
  flex-direction: column;
  align-items: stretch;
}

.detail-image-grid {
  grid-template-columns: repeat(2, minmax(0, 1fr));
}
}

</style>

<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)"
    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>
</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);
  }
}
<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>
    </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>
</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;  
}
```

```
<template>  
  <div>  
    <div class=" queryArea">  
      <el-select  
        clearable  
        placeholder="请选择设备类型"  
        slot="prepend"  
        @change="getInitData"  
        v-model="device"  
        style="width:150px"  
      >  
        <el-option  
          v-for="(item, index) in deviceOpt"  
          :key="index"  
          :label="item.label"  
          :value="item.value"  
        ></el-option>  
      </el-select>  
    </div>  
    <div style="width: 100%">  
      <el-table :data="tableData">  
        <el-table-column  
          v-for="column in columns"  
          :key="column.prop"  
          :prop="column.prop"  
          :label="column.label"  
          :min-width="column.width"  
        >  
          <template #default="scope">  
            <span v-if="column.prop === 'dishes'">  
              <div>  
                <span v-if='scope.row[column.prop]["left_dishes"]["name"]>  
                  {{ scope.row[column.prop]["left_dishes"]["name"] }}  
                &nbsp; </span>  
                <span>  
                  {{ scope.row[column.prop]["right_dishes"]["name"] }}  
                </span>  
              <span v-else-if="column.prop === 'operation'">  
                <el-button  
                  @click="editDevice(scope.row)"  
                  class="blueColor"  
                  size="medium"  
                  type="text"  
                >编辑</el-button>  
            </span>  
          </template>  
        </el-table-column>  
      </el-table>  
    </div>  
  </div>  
</template>
```

```

        </el-button>
      </span>
      <span v-else>{{ scope.row[column.prop] }}</span>
    </template>
  </el-table-column>
</el-table>
<template>
  <div class="block">
    <el-pagination
      @current-change="handleCurrentChangeFun"
      :hide-on-single-page="false"
      :current-page="currentPage"
      layout="total, jumper, ->, prev, pager, next"
      :total="tableTotal"
    ></el-pagination>
  </div>
</template>
</div>
<el-dialog
  title="编辑"
  center
  :visible.sync="editDialog"
  width="35%"
  :before-close="handleClose"
>
  <el-form
    ref="formLabel"
    label-position="right"
    label-width="80px"
    :rules="rules"
    :model="editDeviceInfo"
  >
    <el-form-item label="设备名称" prop="name">
      <el-input
        :disabled="true"
        v-model="editDeviceInfo.name"
      ></el-input>
    </el-form-item>
    <el-form-item label="设备编号" prop="code">
      <el-input
        :disabled="true"
        v-model="editDeviceInfo.code"
      ></el-input>
    </el-form-item>
    <el-form-item
      :label="
        editDeviceInfo.type !== 4 ? '菜品查找' : '菜品查找-左'
      "
      :prop="editDeviceInfo.type !== 4 ? 'left_dishes_uuid' : ""
    >

```

```

        <el-select
          clearable
          placeholder="请选择"
          filterable
          v-model="editDeviceInfo.left_dishes_uuid"
        >
          <el-option
            v-for="(item, index) in dishesOpt"
            :key="index"
            :label="item.name"
            :value="item.uuid"
          ></el-option>
        </el-select>
      </el-form-item>
      <el-form-item
        label="菜品查找-右"
        prop="right_dishes_uuid"
        v-if="editDeviceInfo.type == 4"
      >
        <el-select
          clearable
          placeholder="请选择"
          filterable
          v-model="editDeviceInfo.right_dishes_uuid"
        >
          <el-option
            v-for="(item, index) in dishesOpt"
            :key="index"
            :label="item.name"
            :value="item.uuid"
          ></el-option>
        </el-select>
      </el-form-item>
    </el-form>
    <span slot="footer" class="dialog-footer">
      <el-button @click="handleClose">取 消</el-button>
      <el-button type="primary" @click="saveDevice" v-no-more-click
        >保 存</el-button>
    >
  </span>
</el-dialog>
</div>
</template>
<script>
export default {
  name: "weighingPlatform",
  data(vm) {
    return {
      device: 0,
      deviceOpt: [

```

```

        { label: "全部", value: 0 },
        { label: "称重台-400 型", value: 2 },
        { label: "称重台-450 型-单称", value: 3 },
        { label: "称重台-450 型-双称", value: 4 },
        { label: "电子价签", value: 7 },
        { label: '称重台-460 型',value: 10 }
    ],
    currentPage: 1,
    tableTotal: 0,
    tableData: [],
    columns: [
        { prop: "name", label: "设备名称",width: "150" },
        { prop: "code", label: "设备编号",width: "150" },
        { prop: "type_cn", label: "设备类型", width: "200"},
        { prop: "dishes", label: "菜品名称", width: "200" },
        { prop: "push_dishes_user", label: "排菜人员", width: "100" },
        { prop: "push_dishes_time", label: "排菜时间", width: "150" },
        { prop: "operation", label: "操作" },
    ],
    editDialog: false,
    editDeviceInfo: {},
    dishesOpt: [],
    rules: {
        name: { ...this.$rules.noEmpty, message: "设备名称不能为空" },
        code: { ...this.$rules.noEmpty, message: "设备编号不能为空" },
        left_dishes_uuid: {
            ...this.$rules.noEmpty,
            message: "菜品查找不能为空",
        },
    },
};

};

mounted() {
    this.getInitData();
    this.getDishes();
},

methods: {
    handleCurrentChangeFun(val) {
        this.currentPage = val;
        this.getInitData();
    },
    getInitData() {
        this.$axios
            .post(
                "/p/equipment/pushList",
                this.$qs({
                    page: this.currentPage,
                    type: this.device,
                })
            )
    }
}

```

```
.then((res) => {
  if (res.code === 0) {
    this.tableData = res.data.rows;
    this.tableTotal = res.data.total;
  } else {
    this.$utils.message({
      type: "error",
      message: res.message,
    });
  }
});

},

changeDevice() {},

editDevice(item) {
  this.editDialog = true;
  this.editDeviceInfo = item;
},

handleClose() {
  this.$refs["formLabel"].resetFields();
  this.editDialog = false;
  this.getInitData();
},

saveDevice() {
  this.$refs["formLabel"].validate(valid => {
    if(valid){
      if(!this.editDeviceInfo.left_dishes_uuid && !this.editDeviceInfo.right_dishes_uuid){
        this.$message({
          type: "warning",
          message: this.editDeviceInfo.type ==4 ? '菜品查找至少选择一个' : '请选择菜品查找',
        });
        return;
      }
      const params = {
        equipment_id: this.editDeviceInfo.id,
        left_dishes_uuid: this.editDeviceInfo.left_dishes_uuid,
        right_dishes_uuid: this.editDeviceInfo.right_dishes_uuid,
      };
      this.$axios
        .post("/p/equipment/pushDishes", this.$qs(params))
        .then((res) => {
          if (res.code === 0) {
            this.handleClose();
            this.$message({
              type: "success",
              message: res.message,
            });
          } else {
            this.$utils.message({
              type: "error",
              message: res.message,
            });
          }
        });
      }
    }
  });
}
```

```
        });
    }
    });
}
})
},
getDishes(e) {
    this.$axios
        .post(
            "/p/cater/search",
            this.$qs({
                keyword: "",
            })
        )
        .then((res) => {
            if (res.code === 0) {
                this.dishesOpt = res.data.list;
            } else {
                this.$utils.message({
                    type: "error",
                    message: res.message,
                });
            }
        });
    },
},
};
</script>
<style></style>
<template>
    <div class="wrap">
        <el-card class="card" shadow="hover">
            <div class="title" style="display: none">
                <div class="titleLogo"></div>
                <div class="titleText">基本信息</div>
            </div>
            <div
                class="cardMain"
                v-loading="loading"
                element-loading-text="信息提交中"
                element-loading-spinner="el-icon-loading"
            >
                <el-form
                    label-position="top"
                    ref="form"
                    :rules="rules"
                    :model="form"
                    label-width="80px"
                >
                    <div class="formLeft">
```

```
<el-form-item
  class="name"
  label=" 菜品名称"
  prop="name"
>
  <div style="display: flex; gap: 8px; align-items: center;">
    <el-input
      v-model="form.name"
      placeholder="请输入"
    ></el-input>
    <el-button
      type="primary"
      size="mini"
      :loading="aiLoading"
      @click="handleAiGenerate"
    >
      AI 一下
    </el-button>
    <el-tooltip
      class="item"
      effect="dark"
      placement="top"
      content="AI 生成的菜品与重量仅供参考，以实际为准"
    >
      <i class="el-icon-warning-outline" style="color:#909399; font-size:14px; cursor:pointer;"></i>
    </el-tooltip>
  </div>
</el-form-item>
<el-form-item label=" 菜品分类" prop="category">
  <el-select
    style="width: 100%"
    @change="selectCategory"
    v-model="form.category"
    clearable
    placeholder="请选择"
  >
    <el-option
      v-for="(item, index) in categoryData"
      :key="index"
      :label="item.label"
      :value="item.value"
    >
    </el-option>
  </el-select>
</el-form-item>
<el-form-item label=" 菜品细分类" prop="sub_category">
  <el-select
    style="width: 100%"
    v-model="form.sub_category"
    clearable
```