Vuetify datatable

**Install Dependencies :**

npm install vuetify

npm install @fortawesome/fontawesome-free

npm install @mdi/font

**Créer**



import "vuetify/styles";

import { createVuetify } from "vuetify";

import \* as components from "vuetify/components";

import \* as directives from "vuetify/directives";

const customeTheme = {

    dark: false,

    colors: {

        primary: "#673AB7",

        secondary: "#424242",

        accent: "#82B1FF",

        error: "#FF5252",

        info: "#2196F3",

        success: "#4CAF50",

        warning: "#FFC107",

        lightblue: "#14c6FF",

        yellow: "#FFCF00",

        pink: "#FF1976",

        orange: "#FF8657",

        magenta: "#C33AFC",

        darkblue: "#1E2D56",

        gray: "#909090",

        neutralgray: "#9BA6C1",

        green: "#2ED47A",

        red: "#FF5c4E",

        darkblueshade: "#308DC2",

        lightgray: "#BDBDBD",

        lightpink: "#FFCFE3",

        white: "#FFFFFF",

        muted: "#6c757d",

    },

};

const vuetify = createVuetify({

    components,

    directives,

    theme: {

        defaultTheme: "customeTheme",

        themes: {

            customeTheme,

        },

    },

});

export default vuetify;

**Modifier**

import axios from "./components/config/axios";

import { createApp } from 'vue';

import { createRouter, createWebHashHistory } from "vue-router";

import { routes } from "./routes";

import 'bootstrap/dist/css/bootstrap.min.css'

import "@mdi/font/css/materialdesignicons.css";

import "@fortawesome/fontawesome-free/css/all.css";

import App from './App.vue'

import vuetify from "./vuetify";

let app = createApp(App)

const router = createRouter({

    history: createWebHashHistory(),

    routes: routes,

})

app.use(router).use(vuetify);

app.mount("#app");

**Code datatable exemple :**



<template>

        <div v-if="isLoading">

            <h2> Loading .... </h2>

        </div>

  <div v-else>

  <v-data-table

    :headers="headers"

    :items="categories"

    :itemsPerPage="itemsPerPage"

    :sort-by="[{ key: 'nomcategorie', order: 'asc' }]"

    class="elevation-1"

  >

    <template v-slot:top>

      <v-toolbar flat>

        <v-toolbar-title>Catégories</v-toolbar-title>

        <v-divider class="mx-4" inset vertical></v-divider>

        <v-spacer></v-spacer>

        <router-link :to="{name: 'createCategory'}" class="btn btn-primary">New Category</router-link>

      </v-toolbar>

    </template>

    <template v-slot:item="{ item }">

      <tr>

        <td>{{ item.nomcategorie }}</td>

        <td><img :src="item.imagecategorie" :alt=item.nomcategorie  width="100"/></td>

        <td>

          <router-link :to="{name: 'editCategory', params: { id: item.id }}">

             <v-icon

       size="small"

       color="green"

      >

        mdi-pencil

      </v-icon>

      </router-link>

      <v-icon

        color="red"

        size="small"

        @click="deleteCategorie(item.id)"

      >

        mdi-delete

      </v-icon>

        </td>

      </tr>

    </template>

  </v-data-table>

  </div>

</template>

<script setup>

import { VDataTable } from 'vuetify/labs/VDataTable';

import axios from '../config/axios.js';

import { ref, onMounted } from 'vue';

const categories = ref([]);

const isLoading = ref(true);

const itemsPerPage = 5;

const headers = ref([

  {

    title: 'Nom',

    key: 'nomcategorie',

    sortable: true,

  },

  {

    title: 'Image',

    key: 'imagecategorie',

    sortable: false

  },

  { title: 'Actions',

  key: 'actions',

  sortable: false

 },

]);

const fetchCategories = async () => {

  try {

    const response = await axios.get('/api/categories/');

    isLoading.value = false;

    categories.value = response.data;

  } catch (error) {

    console.error('Error fetching categories:', error);

  }

};

const deleteCategorie = async (id) => {

 try {

    await axios.delete(`/api/categories/${id}`);

    const i = categories.value.findIndex(data => data.id === id);

    if (i !== -1) {

      categories.value.splice(i, 1);

    }

  } catch (error) {

    console.error('Error deleting category:', error);

  }

};

onMounted(() => {

  fetchCategories();

});

</script>

