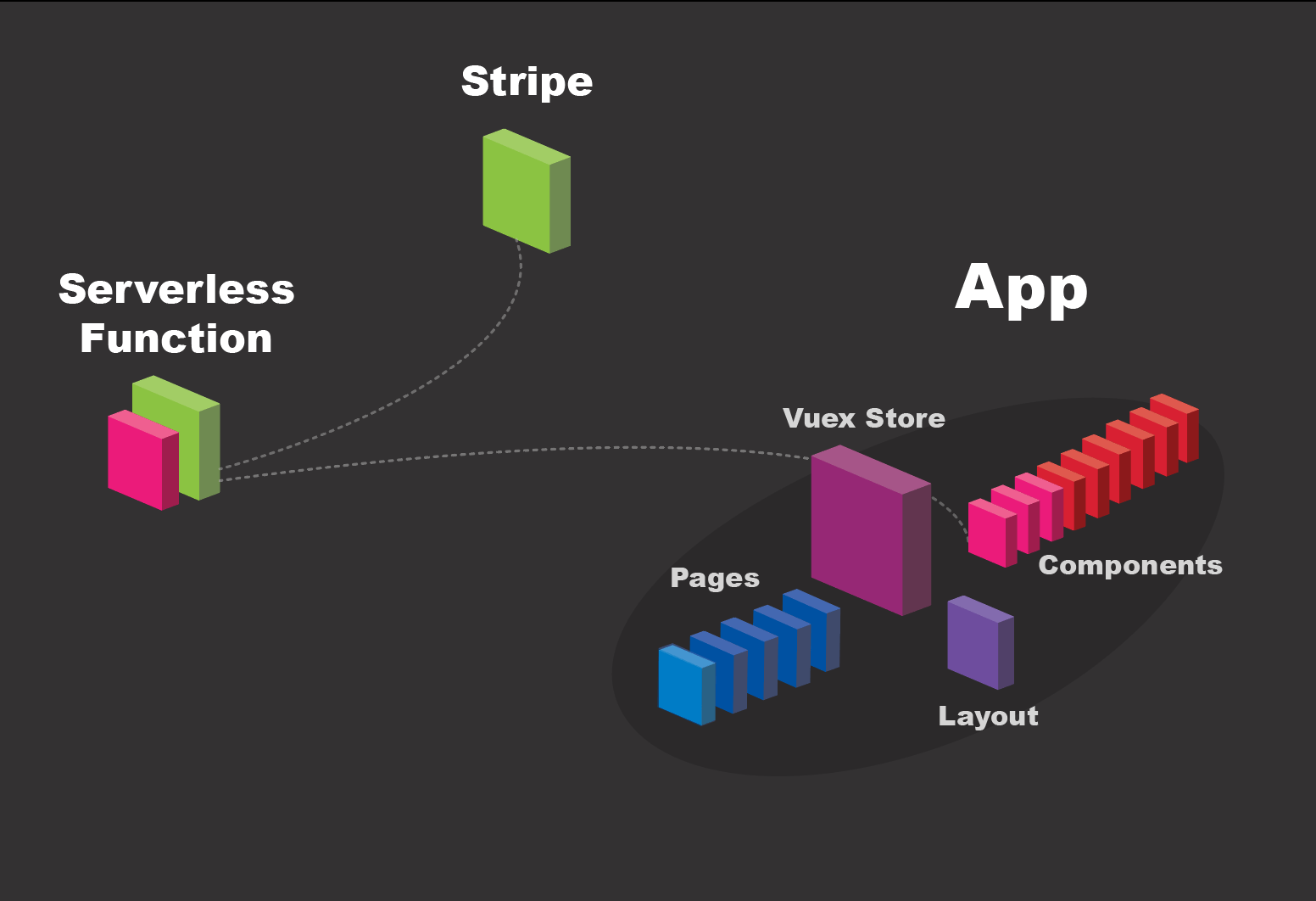
Atelier 6 : Shopping cart avec Vuex



Nous allons construire un store avec Vuex. Dans ce store mis en place avec Vuex, nous détenons un manifeste de toutes nos données produits utilisées pour remplir les pages avec des articles. Nous utiliserons également ces informations pour remplir un objet de panier (actuellement vide) où des articles peuvent être ajoutés pour l'achat. Nous utiliserons ces données sur une page appelée « Cart.vue » dans le répertoire des pages.

Nous allons également utiliser Stripe. Stripe propose un produit appelé Elements. Elements vous permet d'intégrer le système de paiement de Stripe dans votre propre formulaire de paiement et de le personnaliser comme votre propre site pour une expérience cohérente.

L’architecture globale est la suivante :

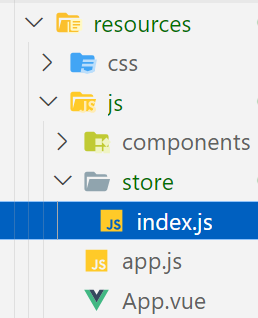


# Installation du package Vuex

npm install vuex@next

# Création du store

Créer store\index.js





import { createStore } from 'vuex'

export default createStore({

  state: {

    cart:[],

    cartTotal:0,

  },

  mutations: {

   async initialiseStore(state) {

         if(localStorage.getItem('cart')){

          state.cart = JSON.parse(localStorage.getItem('cart'))

         }

         if(localStorage.getItem('cartTotal')){

         state.cartTotal = parseFloat(localStorage.getItem('cartTotal'))

         }

         return true;

    },

    addRemoveCart(state,payload){

      //add or remove item

      payload.toAdd?

      state.cart.push(payload.product):

      state.cart = state.cart.filter(function( obj ) {

        return obj.id !== payload.product.id;

      });

      //calculating the total

      state.cartTotal = state.cart.reduce((accumulator, object) => {

        return parseFloat(accumulator) + parseFloat(object.price\*object.qty);

      }, 0);

       //saving in web storage

      localStorage.setItem('cartTotal',JSON.stringify(state.cartTotal));

      localStorage.setItem('cart',JSON.stringify(state.cart));

    },

    updateCart(state,payload){

         //update quantity

        state.cart.find(o => o.id ===  payload.product.id).qty = payload.product.qty;

       //calculating the total

        state.cartTotal = state.cart.reduce((accumulator, object) => {

          return parseFloat(accumulator) + parseFloat(object.price\*object.qty);

        }, 0);

      //saving in web storage

        localStorage.setItem('cartTotal',JSON.stringify(state.cartTotal));

        localStorage.setItem('cart',JSON.stringify(state.cart));

      },

  },

  actions: {},

  modules: {},

})

Il s'agit du magasin Vuex, nous avons un tableau de paniers dans l'état pour enregistrer les articles du panier et cartTotal pour conserver le total du panier.

Dans les mutations, nous avons deux fonctions, la première addRemoveCart pour ajouter ou supprimer des articles du panier et calculer le total et la seconde updateCart pour mettre à jour la quantité d'articles et calculer le total après la mise à jour.

Nous enregistrons les articles du panier dans le stockage Web à l'aide de l'objet « localStorage » afin que si l'utilisateur ferme le site Web, il ne perd pas le panier d'articles et nous puissions charger les articles par la fonction « initialiseStore » que nous appelons au démarrage de l'application.

# Paramétrage



import store from "./store"

….

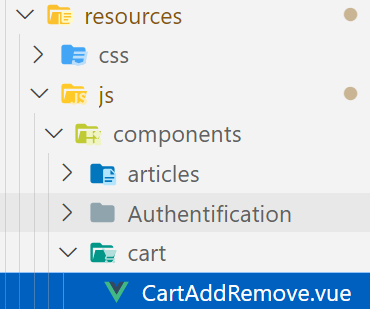
app.use(router).use(store);

# Add To Cart Button

Nous créerons un composant de bouton de panier pour ajouter et supprimer des articles en plus d'augmenter ou de diminuer la quantité. Nous devons installer le package « vue3-toastify » pour effectuer des notifications toast.

npm install vue3-toastify

Créer components\cart\CartAddRemove.vue



<template>

    <div v-if="product"  class="input-group plus-minus">

        <button class="btn btn-outline-secondary " :class="loading?'disabled':''" @click="addOrRemove(-1)" type="button" id="button-addon1">-</button>

            <input type="number" v-model="qty"  disabled class="form-control form-control-sm" placeholder="" aria-label="Example text with button addon" aria-describedby="button-addon1">

        <button class="btn btn-outline-secondary" :class="loading?'disabled':''"  @click="addOrRemove(1)" type="button" id="button-addon1">+</button>

    </div>

 </template>

 <script setup>

 import { toast } from 'vue3-toastify';

 import 'vue3-toastify/dist/index.css';

 import { onMounted , defineProps, ref} from 'vue';

 import store from "../../store";

 const props = defineProps({

  product: {},

  qty:0,

  loading:false

})

const product = ref(props.product);

           let qty = ref(props.qty);

           let  loading=ref(props.loading);

        const addOrRemove=(number)=>{ alert(qty.value)

           loading.value = true

            if(number === 1){

                if(qty.value < 10){

               qty.value++

                product.value.qty =qty.value

                store.commit('updateCart',{product:product.value})

                    toast.success('cart updated', {

                        autoClose: 1000,

                    });

                }else{

                    toast.warning('You reached the limit', {

                        autoClose: 3000,

                    });

                }

            }

            if(number === -1){

                if(qty.value > 1){

                   qty.value--

                    product.value.qty =qty.value

                    store.commit('updateCart',{product:product.value})

                    toast.success('cart updated', {

                            autoClose: 1000,

                    });

                }else{

                    toast.warning('You reached the limit', {

                        autoClose: 3000,

                    });

                }

            }

           loading.value = false

         }

     onMounted(() => {

        qty.value = product.value.qty

     })

 </script>

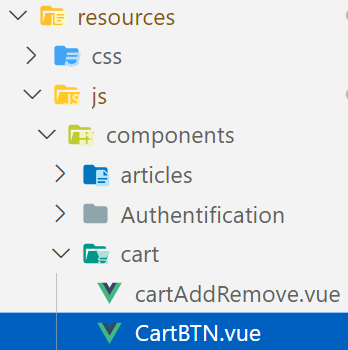
 <style>

.plus-minus input{

    max-width: 50px;

}

</style>





<https://icons.getbootstrap.com/>

<template>

    <button type="button" @click="addOrRemove()" class="cart-btn btn btn-sm btn-outline-secondary me-2">

    <span v-if="toAdd">

                    <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-cart-fill" viewBox="0 0 16 16">

                    <path d="M0 1.5A.5.5 0 0 1 .5 1H2a.5.5 0 0 1 .485.379L2.89 3H14.5a.5.5 0 0 1 .491.592l-1.5 8A.5.5 0 0 1 13 12H4a.5.5 0 0 1-.491-.408L2.01 3.607 1.61 2H.5a.5.5 0 0 1-.5-.5zM5 12a2 2 0 1 0 0 4 2 2 0 0 0 0-4zm7 0a2 2 0 1 0 0 4 2 2 0 0 0 0-4zm-7 1a1 1 0 1 1 0 2 1 1 0 0 1 0-2zm7 0a1 1 0 1 1 0 2 1 1 0 0 1 0-2z"/>

                    </svg>

    </span>

    <span v-if="!toAdd">

                      <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-cart-x-fill" viewBox="0 0 16 16">

                     <path d="M.5 1a.5.5 0 0 0 0 1h1.11l.401 1.607 1.498 7.985A.5.5 0 0 0 4 12h1a2 2 0 1 0 0 4 2 2 0 0 0 0-4h7a2 2 0 1 0 0 4 2 2 0 0 0 0-4h1a.5.5 0 0 0 .491-.408l1.5-8A.5.5 0 0 0 14.5 3H2.89l-.405-1.621A.5.5 0 0 0 2 1H.5zM6 14a1 1 0 1 1-2 0 1 1 0 0 1 2 0zm7 0a1 1 0 1 1-2 0 1 1 0 0 1 2 0zM7.354 5.646 8.5 6.793l1.146-1.147a.5.5 0 0 1 .708.708L9.207 7.5l1.147 1.146a.5.5 0 0 1-.708.708L8.5 8.207 7.354 9.354a.5.5 0 1 1-.708-.708L7.793 7.5 6.646 6.354a.5.5 0 1 1 .708-.708z"/>

                    </svg>

   </span>

    </button>

     <CartAddRemove v-if="!toAdd" :product="item"/>

 </template>

 <script setup>

 import CartAddRemove from './CartAddRemove.vue';

 import { toast } from 'vue3-toastify';

 import 'vue3-toastify/dist/index.css';

 import { onMounted , defineProps, ref} from 'vue';

 import store from "../../store";

 const props = defineProps({

  product: {}

})

const product = ref(props);

            let toAdd=ref(true);

            let item=ref([]);

        const addOrRemove=async ()=>{

                item.value.qty = 1

                store.commit('addRemoveCart',{product:item.value,toAdd:toAdd.value})

                let toasMSG;

                toAdd.value ? toasMSG = 'Added to cart'

                      :  toasMSG = 'Removed from cart'

                toast(toasMSG, {

                autoClose: 1000,

                });

                toAdd.value = !toAdd.value

         }

         onMounted(() => {

         console.log(store.state.cart)

         let cart = store.state.cart

         let obj = cart.find(o => o.id ===  product.value.id);

         if(obj) {

            toAdd.value=false

            item.value = obj

         }

         else{

            item.value = product.value

            toAdd.value=true

         }

     }

 )

 </script>

 <style>

 .cart-btn{

   width:40px;

   height: 38px;

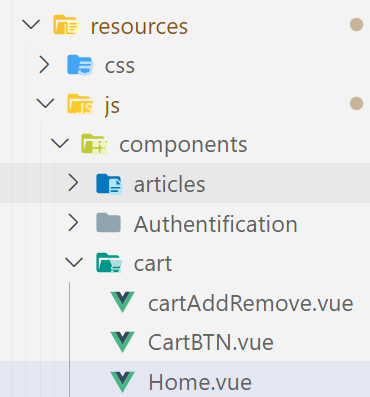
 }

.plus-minus input{

    max-width: 50px;

}

</style>



<template>

    <div class="container min-h-content py-5 text-center">

        <div class="row py-lg-5">

            <div class="row row-cols-1 row-cols-sm-2 row-cols-md-3 g-3">

                <div class="col" v-for="product in products"  :key="product.id">

                    <div class="card shadow-sm">

                        <img class="bd-placeholder-img card-img-top" width="100%"  :src="product.image" alt="">

                        <div class="card-body">

                        <p class="card-text">{{ product.name }}</p>

                        <div class="d-flex justify-content-between align-items-center">

                            <div class="btn-group">

                                <CartBTN :product="product"/>

                            </div>

                            <small class="text-muted"><i class="bi bi-currency-dollar"></i>{{ product.price }}</small>

                        </div>

                        </div>

                    </div>

                </div>

            </div>

        </div>

    </div>

</template>

<script setup>

import CartBTN from './CartBTN.vue'

          let  products=[

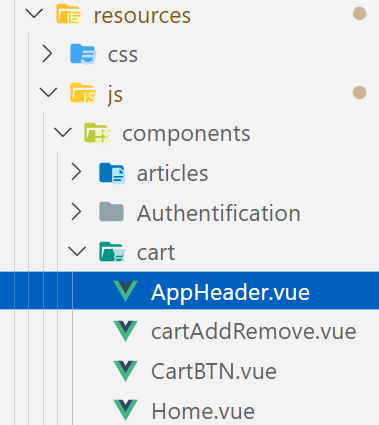
                {id:1, name:'First phone',image:'http://localhost:5173/img/phone1.jpg', price:100},

                {id:2, name:'Second phone',image:'http://localhost:5173/img/phone2.jpg', price:150},

                {id:3, name:'Third phone',image:'http://localhost:5173/img/phone3.jpg', price:180},

            ]

</script>



<template>

<ul class="navbar-nav ms-auto mb-2 mb-lg-0">

    <li class="nav-item me-3">

      <router-link class="nav-link" :class="$route.name == 'Cart'? 'active':''" aria-current="page" :to="{ name: 'Cart' }">

        <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-cart4" viewBox="0 0 16 16">

        <path d="M0 2.5A.5.5 0 0 1 .5 2H2a.5.5 0 0 1 .485.379L2.89 4H14.5a.5.5 0 0 1 .485.621l-1.5 6A.5.5 0 0 1 13 11H4a.5.5 0 0 1-.485-.379L1.61 3H.5a.5.5 0 0 1-.5-.5zM3.14 5l.5 2H5V5H3.14zM6 5v2h2V5H6zm3 0v2h2V5H9zm3 0v2h1.36l.5-2H12zm1.11 3H12v2h.61l.5-2zM11 8H9v2h2V8zM8 8H6v2h2V8zM5 8H3.89l.5 2H5V8zm0 5a1 1 0 1 0 0 2 1 1 0 0 0 0-2zm-2 1a2 2 0 1 1 4 0 2 2 0 0 1-4 0zm9-1a1 1 0 1 0 0 2 1 1 0 0 0 0-2zm-2 1a2 2 0 1 1 4 0 2 2 0 0 1-4 0z"/>

        </svg>

        <span v-if="$store.state.cart.length > 0" class="align-items-center justify-content-center translate-middle badge rounded-pill bg-secondary">

          {{ $store.state.cart.length }}

        </span>

      </router-link>

    </li>

</ul>

</template>

<script setup>

    import { onMounted } from 'vue';

    import store from "../../store";

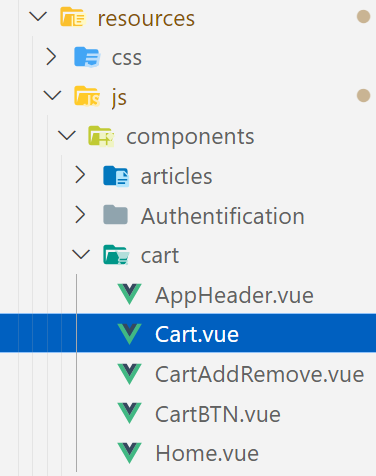
    onMounted(() => {

         store.commit('initialiseStore')

            }

);

</script>



<template>

    <section class="h-100 h-custom" >

    <div class="container py-5 h-100">

    <div class="row d-flex justify-content-center align-items-center h-100">

        <div class="col">

        <div class="card border-0">

            <div class="card-body p-4">

            <div class="row">

                <div class="col-lg-7">

                <h5 class="mb-3"><router-link :to="{name:'Home'}" class="text-body"><i

                        class="fas fa-long-arrow-alt-left me-2"></i>Continue shopping</router-link></h5>

                <hr>

                <div class="d-flex justify-content-between align-items-center mb-4">

                    <div>

                    <p class="mb-0">You have {{ $store.state.cart.length }} items in your cart</p>

                    </div>

                </div>

                <div v-for="item in $store.state.cart" class="card mb-3 shadow-sm border-0" :key="item.id">

                    <div class="card-body">

                    <div class="d-flex justify-content-between">

                        <div class="d-flex flex-row align-items-center">

                            <div>

                                <img

                                :src="item.image"

                                class="img-fluid rounded-3" alt="Shopping item" style="width: 65px;">

                            </div>

                            <div class="ms-3">

                                <p>{{ item.name }}</p>

                            </div>

                        </div>

                        <div class="d-flex flex-row align-items-center">

                            <div >

                                <CartAddRemove :product="item"/>

                            </div>

                        </div>

                        <div class="d-flex flex-row align-items-center">

                            <div >

                                <h5 class="mb-0"><i class="bi bi-currency-dollar"></i>{{ item.price\*item.qty }}</h5>

                                <small v-if="item.hasDiscount" class="text-muted text-decoration-line-through"><i class="bi bi-currency-dollar"></i>{{ item.price}}</small>

                            </div>

                            <a role="button" @click="removeItem(item)" class="ms-4" style="color: #cecece;">

                                <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-trash-fill" viewBox="0 0 16 16">

                                    <path d="M2.5 1a1 1 0 0 0-1 1v1a1 1 0 0 0 1 1H3v9a2 2 0 0 0 2 2h6a2 2 0 0 0 2-2V4h.5a1 1 0 0 0 1-1V2a1 1 0 0 0-1-1H10a1 1 0 0 0-1-1H7a1 1 0 0 0-1 1H2.5zm3 4a.5.5 0 0 1 .5.5v7a.5.5 0 0 1-1 0v-7a.5.5 0 0 1 .5-.5zM8 5a.5.5 0 0 1 .5.5v7a.5.5 0 0 1-1 0v-7A.5.5 0 0 1 8 5zm3 .5v7a.5.5 0 0 1-1 0v-7a.5.5 0 0 1 1 0z"/>

                                </svg>

                            </a>

                        </div>

                    </div>

                    </div>

                </div>

                </div>

                <div class="col-lg-5">

                <div class="card bg-primary text-white rounded-0 border-0">

                    <div class="card-body">

                    <div class="d-flex justify-content-between align-items-center mb-4">

                        <h5 class="mb-0">Cart details</h5>

                        <i class="bi bi-cart3 h1"></i>

                    </div>

                    <hr class="my-4">

                    <div class="d-flex justify-content-between">

                        <p class="mb-2">Subtotal</p>

                        <p class="mb-2"><i class="bi bi-currency-dollar"></i>{{ $store.state.cartTotal }}</p>

                    </div>

                    <div class="d-flex justify-content-between mb-4">

                        <p class="mb-2">Total</p>

                        <p class="mb-2"><i class="bi bi-currency-dollar"></i>{{ $store.state.cartTotal }}</p>

                    </div>

                    <button type="button" class="btn btn-info btn-block btn-lg">

                        Checkout

                    </button>

                    </div>

                </div>

                </div>

            </div>

            </div>

        </div>

        </div>

    </div>

    </div>

    </section>

    </template>

    <script setup>

    import store from "../../store"

    import CartAddRemove from './CartAddRemove.vue';

     const  removeItem=(item)=>{

            store.commit('addRemoveCart',{product:item,toAdd:false})

        }

    </script>



import Cart from './components/cart/Cart.vue';

import Home from './components/cart/Home.vue';

….

   {

        path:'/cart',

        name:'Cart',

        component:Cart

    },

    {

        path:'/shopping',

        name:'Home',

        component:Home

    },

# Chargement du cart

Pour charger les données de notre panier, nous démarrons l'application dont nous avons besoin pour ajouter la fonction « initialiseStore » à src\App.vue



 <router-link to="/logout" class="nav-item nav-link">  Logout </router-link>

                    <router-link to="/shopping" class="nav-item nav-link"> Shopping Cart </router-link>

                    <AppHeader />

…

<script setup>

    import AppHeader from './components/cart/AppHeader.vue';

    import { onMounted } from 'vue';

    import store from "./store"

    onMounted(() => {

         store.commit('initialiseStore')

            }

);

</script>