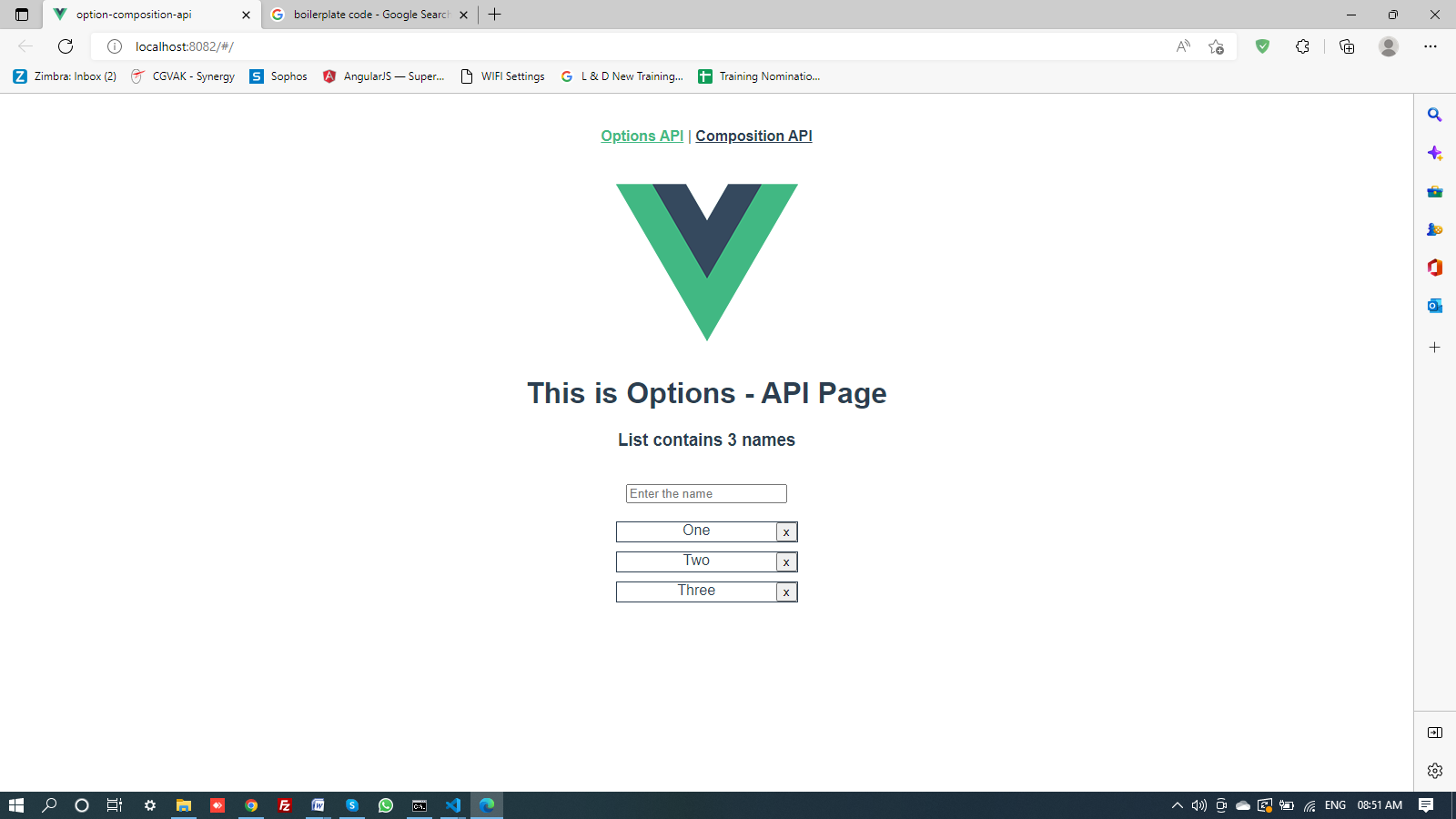
**Options and Composition API**

****

**App.vue**

<template>

<nav>

<router-link to="/">Options API</router-link> |

<router-link to="/composition">Composition API</router-link>

</nav>

<router-view/>

</template>

<style>

#app {

font-family: Avenir, Helvetica, Arial, sans-serif;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

text-align: center;

color: #2c3e50;

}

nav {

padding: 30px;

}

nav a {

font-weight: bold;

color: #2c3e50;

}

nav a.router-link-exact-active {

color: #42b983;

}

</style>

**Code in > router > index.js**

import { createRouter, createWebHashHistory } from 'vue-router'

import Option from '../views/Option.vue'

const routes = [

{

path: '/',

name: 'option',

component: Option

},

{

path: '/composition',

name: 'composition',

// route level code-splitting

// this generates a separate chunk (about.[hash].js) for this route

// which is lazy-loaded when the route is visited.

component: () => import(/\* webpackChunkName: "about" \*/ '../views/Composition.vue')

}

]

const router = createRouter({

history: createWebHashHistory(),

routes

})

export default router

**Code in > src > views > Options.vue**

<template>

<div class="option">

<img alt="Vue logo" src="../assets/logo.png" />

<h1>This is Options - API Page </h1>

<!-- <h3> List contains {{names.length}} names</h3> -->

<h3> List contains {{ nameCount }} names </h3>

</div> <br />

<div>

<input type="text" placeholder="Enter the name"

v-model="newName" @keyup.enter="addNewName()" />

</div>

<div>

<ul>

<li v-for="(name, index) in names" :key="name.id">

<span> {{name.pname}}</span>

<button @click="deleteName(index)"> x </button>

<!-- @click="deleteName(index)" -->

</li>

</ul>

</div>

</template>

<script>

export default{

data(){

return{

newName: '',

names: [

{

id: 1,

pname: 'One'

},

{

id: 2,

pname: 'Two'

},

{

id: 3,

pname: 'Three'

} ],

wordCollection:['failure','hatred','dislike','dishonest']

}

},

computed:{

nameCount(){

return this.names.length

}

},

methods: {

addNewName()

{

let temp = {

id: Date.now(),

pname: this.newName

}

this.names.push(temp)

this.newName=''

},

deleteName(index){

console.log(index)

this.names.splice(index, 1)

}

},

watch:{

newName(newVal){

// console.log(newVal)

if (this.wordCollection.includes(newVal.toLowerCase())){

this.newName=''

alert('You must Never say ' + newVal)

} } } } </script>

<style>

ul{

list-style: none;

padding: 0;

width: 200px;

margin: 20px auto 0;

}

li{

border: 1px solid;

display: flex;

margin-bottom: 10px;

}

li span{

flex-grow: 1;

}

</style>

**Code in > src > views > Composition.vue**

<template>

<div class="option">

<img alt="Vue logo" src="../assets/logo.png" />

<h1>This is Composition - API Page </h1>

<!-- <h3> List contains {{names.length}} names</h3> -->

<h3> List contains {{ nameCount }} names </h3>

</div> <br />

<!-- <div>

<input type="text" placeholder="Enter the name"

v-model="data.newName" @keyup.enter="addNewName()" />

</div> -->

<div>

<input type="text" placeholder="Enter the name"

v-model="newName" @keyup.enter="addNewName()" />

</div>

<div>

<ul>

<!-- <li v-for="(name, index) in data.names" :key="name.id"> -->

<li v-for="(name, index) in names" :key="name.id">

<span> {{name.pname}}</span>

<button @click="deleteName(index)"> x </button>

<!-- @click="deleteName(index)" -->

</li>

</ul>

</div>

</template>

<script setup>

import { ref, computed, watch } from 'vue'

//import { reactive, computed, watch } from 'vue'

// export default{

// setup(){

let newName = ref('')

//let names = ref([])

let names = ref([

{ id: 1,

pname: 'One' },

{ id: 2,

pname: 'Two' },

{ id: 3,

pname: 'Three'} ])

// let data = reactive({

// newName: '',

// names:[

// { id: 1,

// pname: 'One' },

// { id: 2,

// pname: 'Two' },

// { id: 3,

// pname: 'Three'}

// ]

// })

const wordCollection = ['failure','hatred','dislike','dishonest']

let nameCount = computed(() => {

return names.value.length

//return data.names.length

})

function addNewName()

{

let temp = {

id: Date.now(),

pname: newName.value

//pname: data.newName

}

names.value.push(temp)

newName.value =''

//data.names.push(temp)

//data.newName = ''

}

function deleteName(index){

console.log(index)

names.value.splice(index, 1)

//data.names.splice(index,1)

}

watch(newName, (newVal) => {

// watch(data, (newVal) => {

console.log(newVal)

if (wordCollection.includes(newVal.toLowerCase())){

newName.value = ''

alert('You must Never say ' + newVal )

// if (wordCollection.includes(newVal.newName.toLowerCase())){

// alert('You must Never say ' + data.newName )

// data.newName = ''

}

})

// return{

// newName, names,

// // data,

// nameCount,

// addNewName, deleteName

// }

// }

// }

</script>

<!-- // data(){

// return{

// newName: '',

// names: [

// {

// id: 1,

// pname: 'One'

// },

// {

// id: 2,

// pname: 'Two'

// },

// {

// id: 3,

// pname: 'Three'

// } ],

// wordCollection:['failure','hatred','dislike','dishonest']

// }

// },

// computed:{

// nameCount(){

// return this.names.length

// }

// },

// methods: {

// addNewName()

// {

// let temp = {

// id: Date.now(),

// pname: this.newName

// }

// this.names.push(temp)

// this.newName=''

// },

// deleteName(index){

// console.log(index)

// this.names.splice(index, 1)

// }

// },

// watch:{

// newName(newVal){

// // console.log(newVal)

// if (this.wordCollection.includes(newVal.toLowerCase())){

// this.newName=''

// alert('You must Never say ' + newVal)

// }

// }

// }

//} -->

<style>

ul{

list-style: none;

padding: 0;

width: 200px;

margin: 20px auto 0;

}

li{

border: 1px solid;

display: flex;

margin-bottom: 10px;

}

li span{

flex-grow: 1;

} </style>