**Step By Step**

1. Type on terminal.

npm install vue-router

npm install vuex

1. Chapter 231 – Registering Routes

Create router.js and type below.

import { createRouter, createWebHistory } from 'vue-router';

const router = createRouter({

    history: createWebHistory(),

    routes: [

        { path: '/', redirect: '/coaches' },

        { path: '/coaches', component: null },

        {

            path: '/coaches/id',

            component: null,

            children: [

                // /coaches/c1/contact

                { path: 'contact', component: null },

            ]

        },

        { path: '/register', component: null },

        { path: '/requests', component: null },

        { path: '/:notFound(.\*)', component: null },

    ]

});

export default router;

1. Chapter 232 – Adding Route Page Components
2. Add your own folder and sub folder depends on what you list. Better for you to divide it into some folder such as pages (divide it into some function like coaches and requests), NotFound.vue.
3. Import all the Vue pages and define it as components in Router.js like below.

import CoachDetail from './pages/coaches/CoachDetail.vue';

import CoachesList from './pages/coaches/CoachesList.vue';

import CoachRegistration from './pages/coaches/CoachRegistration.vue';

import ContactCoach from './pages/coaches/ContactCoach.vue';

import RequestsReceived from './pages/coaches/RequestsReceived.vue';

import NotFound from './pages/coaches/NotFound.vue';

const router = createRouter({

    history: createWebHistory(),

    routes: [

        { path: '/', redirect: '/coaches' },

        { path: '/coaches', component: CoachesList },

        {

            path: '/coaches/id',

            component: CoachDetail,

            children: [

                // /coaches/c1/contact

                { path: 'contact', component: ContactCoach },

            ]

        },

        { path: '/register', component: CoachRegistration },

        { path: '/requests', component: RequestsReceived },

        { path: '/:notFound(.\*)', component: NotFound },

    ]

});

1. In App.vue we can render our router-view to tell view router where to render these different components we registered. And import App.vue in main.js

App.vue

<template>

    <router-view></router-view>

</template>

main.js

import { createApp } from 'vue';

import router from './router.js';

import App from './App.vue';

const app = createApp(App);

app.use(router);

app.mount('#app');

1. Chapter 233 – Working On The Main Layout And Styling
2. Create two section Filter and List of Coaches in CoachesList.vue. Add css file into App.vue.

<template>

    <section>

        FILTER

    </section>

    <section>

        LIST OF COACHES

    </section>

</template>

1. Create folder components, layout, and TheHeader.vue. Create the header template in here and put the CSS.

<template>

    <header>

        <nav>

            <h1><router-link to="/">Find A Coach</router-link></h1>

            <ul>

                <li><router-link to="/coaches">All Coaches</router-link></li>

                <li><router-link to="/requests">Requests</router-link></li>

            </ul>

        </nav>

    </header>

</template>

1. Import TheHeader in App.vue.

<script>

import TheHeader from './components/layout/TheHeader.vue';

export default {

  components: {

    TheHeader

  }

}

</script>

1. Chapter 234 – Wiring Up Pages
2. Create RequestsReceived.vue.

<template>

    REQUESTS

</template>

1. Add some template in CoachesList.vue.

<template>

    <section>

        FILTER

    </section>

    <section>

        <div class="controls">

            <button>Refresh</button>

            <router-link to="/register">Register As Coach</router-link>

        </div>

        <ul>LIST OF COACHES</ul>

    </section>

</template>

1. Add template in CoachRegistration.vue.

<template>

    REGISTER

</template>

1. Add template in CoachDetail.vue.

<template>

    DETAILS FOR COACH

    <router-view></router-view>

    <router-link to="/coaches/c1/contact">Contact</router-link>

</template>

1. Chapter 235 – Adding Vuex And Coach Data
2. Create modules, coaches, action.js getters.js index.js and mutations.js. All of these modules are used to store get or list the data.

Index.js

import mutations from './mutations.js';

import actions from './actions.js';

import getters from './getters.js';

export default {

    namespaced: true,

    state() {

        return {

            coaches: [

                [

                    {

                      id: 'c1',

                      firstName: 'Maximilian',

                      lastName: 'Schwarzmüller',

                      areas: ['frontend', 'backend', 'career'],

                      description:

                        "I'm Maximilian and I've worked as a freelance web developer for years. Let me help you become a developer as well!",

                      hourlyRate: 30

                    },

                    {

                      id: 'c2',

                      firstName: 'Julie',

                      lastName: 'Jones',

                      areas: ['frontend', 'career'],

                      description:

                        'I am Julie and as a senior developer in a big tech company, I can help you get your first job or progress in your current role.',

                      hourlyRate: 30

                    }

                ]

            ]

        };

    },

    mutations,

    actions,

    getters,

};

Store/Index.js = main store for all modules

import { createStore } from 'vuex';

import coachesModule from './modules/coaches/index.js';

const store = createStore({

    modules: {

        coaches: coachesModule

    }

});

export default store;

1. Chapter 236 – Working On The Coaches List And List Items
2. Chapter 237 –Adding A Couple Of Base Components (Base Card, Base Button, Base Badge)
3. Chapter 238 – Building The Coach Detail Component
4. Chapter 239 – Filtering Coaches
5. Chapter 240 – Registering As A Coach: The Form
6. Chapter 241 – Adding Coaches To Vuex
7. Chapter 242 – Adding Form Validation
8. Chapter 243 – Working On The Contact Form
9. Chapter 244 – Storing Requests (Messages) With Vuex
10. Chapter 245 – Outputting Incoming Requests (Messages)
11. Chapter 246 – Filtering Requests For The Active Coach
12. Chapter 247 – Sending A PUT HTTP Request To Store Coach Data
13. Chapter 248 – Fetching Coach Data (GET HTTP Request)
14. Chapter 249 – Rendering A Loading Spinner
15. Chapter 250 – Adding HTTP Error Handling
16. Chapter 251 – Sending Coaching Requests HTTP Requests
17. Chapter 252 – Catching HTTP Response Data
18. Chapter 253 – Adding Route Transitions
19. Chapter 254 – The “Not Found” Page And Summary
20. Chapter 255 – Module Resources