Vue.Js Setup Part 3 :

**++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**links :**[**https://www.youtube.com/watch?v=MQZ4PQGw-sc&list=PL8p2I9GklV45qwTH-mdzllUuFRJO-euYn&index=22**](https://www.youtube.com/watch?v=MQZ4PQGw-sc&list=PL8p2I9GklV45qwTH-mdzllUuFRJO-euYn&index=22)

**# fetch data from API | axios >>>>**

**install axios and vue-axios**

**make component**

**use mounted life cycle hooks for call api**

**fetch api with axios**

**use v-for for show list of data**

**api url link :** [**http://dummy.restapiexample.com/api/v1/employees**](http://dummy.restapiexample.com/api/v1/employees)

**TodoList.vue >>>>**

<template>

<div>

<h3>Item List</h3>

<!-- Show loading message while data is being fetched -->

<div v-if="loading">

<p>Loading items, please wait...</p>

</div>

<!-- Display error if something goes wrong -->

<div v-if="error">

<p style="color: red;">{{ error }}</p>

</div>

<!-- Display the table once data is fetched -->

<table v-if="!loading && !error" class="item-table">

<thead>

<tr>

<th>ID</th>

<th>Title</th>

<th>Status</th>

</tr>

</thead>

<tbody>

<tr v-for="item in items" :key="item.id">

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

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

<td>{{ item.completed ? 'Completed' : 'Pending' }}</td>

</tr>

</tbody>

</table>

</div>

</template>

<script>

import axios from 'axios';

export default {

name: "HomePage",

data() {

return {

items: [], // to store fetched data

loading: false, // to manage loading state

error: null, // to handle errors

};

},

async mounted() {

this.loading = true; // Set loading to true when fetching starts

try {

const response = await axios.get('https://jsonplaceholder.typicode.com/todos');

this.items = response.data; // Assign the fetched data to items

} catch (error) {

this.error = 'Failed to fetch data. Please try again later.'; // Show error message on failure

} finally {

this.loading = false; // Set loading to false when fetching is done

}

}

};

</script>

<style scoped>

.item-table {

width: 100%;

border-collapse: collapse;

margin-top: 20px;

}

.item-table th, .item-table td {

border: 1px solid #ddd;

padding: 8px;

text-align: left;

}

.item-table th {

background-color: #f4f4f4;

font-weight: bold;

}

.item-table tr:nth-child(even) {

background-color: #f9f9f9;

}

.item-table tr:hover {

background-color: #f1f1f1;

}

h3 {

margin-bottom: 20px;

color: #4CAF50;

}

</style>

now we are use axios lib and fetch data & complete to fetch data & show on table and error & fetching list data .

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**add bootstrap >>>> ( link is ...** [**https://bootstrap-vue.org/docs/components**](https://bootstrap-vue.org/docs/components) **)**

**docs link >>>>** [**https://bootstrap-vue.org/**](https://bootstrap-vue.org/)

**install one package .....**

**# vue add bootstrap-vue ( not working in vue 3 )**

**Now i am using bootstrap via npm >>>>**

**Step 1: Create a New Vue 3 Project**

**First, create a new Vue 3 project if you don’t have one:**

# vue create my-vue-app

**Navigate to the project directory:**

# cd my-vue-app

**Step 2: Install BootstrapVue 3 and Bootstrap 5**

**Run the following command to install BootstrapVue 3 and Bootstrap 5:**

**# npm install bootstrap@5 bootstrap-vue-3**

**Step 3: Add Bootstrap and BootstrapVue to main.js**

**After installing the necessary packages, you need to import Bootstrap CSS, BootstrapVue 3, and optionally the BootstrapVue icons in your main.js file.**

**Edit src/main.js:**

import { createApp } from 'vue';

import App from './App.vue';

**// Import Bootstrap and BootstrapVue 3 CSS files**

**import 'bootstrap/dist/css/bootstrap.css';**

**import 'bootstrap-vue-3/dist/bootstrap-vue-3.css';**

**// Import BootstrapVue 3**

**import BootstrapVue3 from 'bootstrap-vue-3';**

const app = createApp(App);

**app.use(BootstrapVue3); // Use BootstrapVue 3 globally**

app.mount('#app');

**Use Bootstrap Components in Vue**

**Now, you can use BootstrapVue components in your Vue 3 app. For example, you can use a Bootstrap button or card component like this:**

**Home.vue >>>>**

<template>

<div class="container mt-4">

<b-button variant="primary" @click="sayHello">Click Me</b-button>

<b-card

title="BootstrapVue 3 Card"

class="mt-4"

>

<p class="card-text">This is a card component from BootstrapVue 3!</p>

</b-card>

</div>

</template>

<script>

export default {

name: 'HelloWorld',

methods: {

sayHello() {

alert('Hello from BootstrapVue 3!');

}

}

};

</script>

So use this process use bootstrap via npm component..

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Post API with form data >>>>>**

**first i got api & set in our page when submit data i am getting data in console ovserver.**

**Home.vue >>>>**

<template>

<div class="form-container">

<form @submit.prevent="submitPost" class="post-form">

<div class="form-group">

<label for="title">Title:</label>

<input v-model="postData.title" id="title" required />

</div>

<div class="form-group">

<label for="content">Content:</label>

<textarea v-model="postData.content" id="content" required></textarea>

</div>

<button type="submit" class="submit-btn">Submit</button>

</form>

<!-- Display success message or error message -->

<div v-if="successMessage" class="success-message">{{ successMessage }}</div>

<div v-if="errorMessage" class="error-message">{{ errorMessage }}</div>

</div>

</template>

<script>

import axios from 'axios';

export default {

name:"homePage",

data() {

return {

postData: {

title: '',

content: '',

},

successMessage: '',

errorMessage: '',

};

},

methods: {

async submitPost() {

try {

const response = await axios.post('https://jsonplaceholder.typicode.com/posts', this.postData);

this.successMessage = 'Post submitted successfully!';

this.errorMessage = '';

console.log('Post Response:', response.data);

} catch (error) {

this.errorMessage = 'Failed to submit post.';

this.successMessage = '';

console.error(error);

}

},

},

};

</script>

<style scoped>

/\* Center the form vertically and horizontally \*/

.form-container {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

text-align: center;

flex-direction: column;

}

/\* Style for the form \*/

.post-form {

background-color: #f9f9f9;

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

max-width: 400px;

width: 100%;

display: flex;

flex-direction: column;

}

/\* Style for each form group (label and input) \*/

.form-group {

margin-bottom: 20px;

text-align: left;

}

/\* Style for input fields and textarea \*/

input, textarea {

width: 100%;

padding: 10px;

font-size: 16px;

border: 1px solid #ccc;

border-radius: 4px;

margin-top: 5px;

}

/\* Style for the submit button \*/

.submit-btn {

padding: 10px;

font-size: 16px;

background-color: #28a745;

color: white;

border: none;

border-radius: 4px;

cursor: pointer;

transition: background-color 0.3s;

}

.submit-btn:hover {

background-color: #218838;

}

/\* Style for success and error messages \*/

.success-message {

color: green;

margin-top: 10px;

}

.error-message {

color: red;

margin-top: 10px;

}

</style>

so post is submitted using dummy api for testing ..

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Delete data with API | delete method >>>**

Delete option add one on list and while delete button so record is removed from list >>>

**Home.vue >>>>>**

<template>

<div class="user-list-container">

<h2>User List</h2>

<table class="user-table">

<thead>

<tr>

<th>ID</th>

<th>Name</th>

<th>Email</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

<tr v-for="user in users" :key="user.id">

<td>{{ user.id }}</td>

<td>{{ user.name }}</td>

<td>{{ user.email }}</td>

<td>

<button @click="confirmDelete(user.id)" class="delete-btn">Delete</button>

</td>

</tr>

</tbody>

</table>

<div v-if="errorMessage" class="error-message">{{ errorMessage }}</div>

</div>

</template>

<script>

import axios from 'axios';

export default {

name:"homePage",

data() {

return {

users: [],

errorMessage: '',

};

},

mounted() {

this.fetchUsers();

},

methods: {

// Fetch the user list from the dummy API

async fetchUsers() {

try {

const response = await axios.get('https://jsonplaceholder.typicode.com/users');

this.users = response.data;

} catch (error) {

this.errorMessage = 'Failed to load user list.';

console.error(error);

}

},

// Method to confirm deletion

confirmDelete(userId) {

const isConfirmed = window.confirm('Are you sure you want to delete this user?');

if (isConfirmed) {

this.deleteUser(userId);

}

},

// Delete a user and remove from the local list

async deleteUser(userId) {

try {

// Simulate DELETE request to the API

await axios.delete(`https://jsonplaceholder.typicode.com/users/${userId}`);

// Remove the user from the list after successful deletion

this.users = this.users.filter(user => user.id !== userId);

this.errorMessage = ''; // Reset error message if any

} catch (error) {

this.errorMessage = 'Failed to delete user.';

console.error(error);

}

},

},

};

</script>

<style scoped>

.user-list-container {

max-width: 800px;

margin: 50px auto;

text-align: center;

}

.user-table {

width: 100%;

border-collapse: collapse;

margin-bottom: 20px;

}

.user-table th, .user-table td {

border: 1px solid #ccc;

padding: 10px;

}

.user-table th {

background-color: #f9f9f9;

}

.delete-btn {

background-color: #ff4d4d;

color: white;

border: none;

padding: 8px 12px;

cursor: pointer;

border-radius: 4px;

}

.delete-btn:hover {

background-color: #ff1a1a;

}

.error-message {

color: red;

margin-top: 10px;

}

</style>

while click on delete button then function run delete users .

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Basic Routing | Vue router part 1 >>>>>**

install package for router in vue Js ...

# npm install vue-router

create two component about & home then load code ..

after install write some code in main.js file >>>>

**Main.js >>>**

// Import Vue and App component

import { createApp } from 'vue';

import App from './App.vue';

// Import Bootstrap and BootstrapVue 3 CSS files

import 'bootstrap/dist/css/bootstrap.css';

import 'bootstrap-vue-3/dist/bootstrap-vue-3.css';

// Import BootstrapVue 3

import BootstrapVue3 from 'bootstrap-vue-3';

**// Import Vue Router and necessary methods from vue-router**

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

**// Import the components for routing**

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

**import About from './components/About.vue';**

**// Define the routes**

**const routes = [**

**{ path: '/', component: Home },**

**{ path: '/about', component: About },**

**];**

**// Create the router instance**

**const router = createRouter({**

**history: createWebHistory(),**

**routes,**

**});**

// Create the Vue app instance

const app = createApp(App);

// Use BootstrapVue 3 globally

app.use(BootstrapVue3);

**// Use Vue Router globally**

**app.use(router);**

// Define the global "filter-like" function (for example: capitalize)

app.config.globalProperties.$filters = {

capitalize(val) {

if (!val) return '';

return val.toString().toUpperCase();

},

};

// Mount the app to the #app element

app.mount('#app');

-----------------------------------------------------------------------------------------------

add router-view in app.vue page >>>

when url hit then load component based on url its clear ...

<template>

<div id="app">

<h2>Vue Router how to handle</h2>

**<!-- this is add for link based component load then use router-view open and close both -->**

**<router-view></router-view>**

</div>

</template>

<script>

export default {

name: 'App',

components: {

}

};

</script>

<style>

#app {

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

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

text-align: center;

color: #2c3e50;

margin-top: 60px;

}

</style>

**Home.vue >>>>**

<template>

<div>

<h3>Home Page</h3>

</div>

</template>

<script>

export default {

name:"homePage",

}

</script>

**About.vue >>>>**

<template>

<div>

<h3>About Page</h3>

</div>

</template>

<script>

export default {

name:"aboutPage",

}

</script>

now its working vue routes load based over components..

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Add Routing Link | vue router part 2 >>>>>>**

**App.vue >>>>**

<template>

<div id="app">

<h2>Vue Router how to handle</h2>

<!-- Router link based on href like wise in vue js i am use router-link -->

<ul>

<ol><router-link to="/">Home</router-link></ol>

<ol><router-link to="/about">About</router-link></ol>

</ul>

<!-- this is add for link based component load then use router-view open and close both -->

<router-view></router-view>

</div>

</template>

<script>

export default {

name: 'App',

components: {

}

};

</script>

<style>

#app {

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

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

text-align: center;

color: #2c3e50;

margin-top: 60px;

}

</style>

Its working fine <router-link to="/">Home</router-link> works as href or any click link so its getting data to easy way.

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Pass data in routing params | Vue router part 3 >>>>

**Home.vue >>>>**

<template>

<div class="user-list-container">

<h2>Home List</h2>

**<p><router-link to="user/1">Saurabh</router-link></p>**

**<p><router-link to="user/2">Rishabh</router-link></p>**

**<p><router-link to="user/3">Vimal</router-link></p>**

**<p><router-link to="user/4">Mukesh</router-link></p>**

</div>

</template>

<script>

export default {

name:"homePage",

data() {

return {};

}

};

</script>

<style scoped>

</style>

**About.vue >>>>**

<template>

<div>

<h3>About Page</h3>

</div>

</template>

<script>

export default {

name:"aboutPage",

}

</script>

**Login.vue >>>>**

<template>

<div>

<h3>Login Page</h3>

</div>

</template>

<script>

export default {

name:"loginPage",

}

</script>

User.vue >>>>>

<template>

<div>

<h2>User View Page</h2>

**<h4>User Id is: {{ userId }}</h4>**

</div>

</template>

<script>

export default {

name: "userPage",

data() {

return {

**userId: this.$route.params.id // Corrected from param to params**

};

}

}

</script>

**App.vue >>>>**

<template>

<div id="app">

<h2>Pass data in routing params | Vue router part 3</h2>

<!-- Router link based on href like wise in vue js i am use router-link -->

<ul>

**<ol><router-link to="/">Home</router-link></ol>**

**<ol><router-link to="/about">About</router-link></ol>**

**<ol><router-link to="/login">Login</router-link></ol>**

**<ol><router-link to="/user/:id">User View</router-link></ol>**

</ul>

<!-- this is add for link based component load then use router-view open and close both -->

<router-view></router-view>

</div>

</template>

<script>

export default {

name: 'App',

components: {

}

};

</script>

<style>

#app {

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

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

text-align: center;

color: #2c3e50;

margin-top: 60px;

}

</style>

Main.vue >>>>

// Import Vue and App component

import { createApp } from 'vue';

import App from './App.vue';

// Import Bootstrap and BootstrapVue 3 CSS files

import 'bootstrap/dist/css/bootstrap.css';

import 'bootstrap-vue-3/dist/bootstrap-vue-3.css';

// Import BootstrapVue 3

import BootstrapVue3 from 'bootstrap-vue-3';

// Import Vue Router and necessary methods from vue-router

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

// Import the components for routing

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

import About from './components/About.vue';

**import Login from './components/Login.vue';**

**import User from './components/User.vue';**

// Define the routes

const routes = [

{ path: '/', component: Home },

{ path: '/about', component: About },

**{ path: '/login', component: Login },**

**{ path: '/user/:id', component: User },**

];

// Create the router instance

const router = createRouter({

history: createWebHistory(),

routes,

});

// Create the Vue app instance

const app = createApp(App);

// Use BootstrapVue 3 globally

app.use(BootstrapVue3);

// Use Vue Router globally

app.use(router);

// Define the global "filter-like" function (for example: capitalize)

app.config.globalProperties.$filters = {

capitalize(val) {

if (!val) return '';

return val.toString().toUpperCase();

},

};

// Mount the app to the #app element

app.mount('#app');

**update this code we can manage route param via api or web url ..**