**Life cycle hooks in apis**

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

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

import HomeView from '../views/HomeView.vue'

import LifeCycleCompostionAPI from '@/components/LifeCycleCompostionAPI.vue'

import LifeCyleOptionsAPI from '@/components/LifeCycleOptionsAPI.vue'

const routes = [

{

path: '/',

name: 'lcoapi',

component: LifeCyleOptionsAPI

},

{

path: '/lifecyclecapi',

name: 'lccapi',

component: LifeCycleCompostionAPI

// 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/AboutView.vue')

}

]

const router = createRouter({

history: createWebHistory(process.env.BASE\_URL),

routes

})

export default router

**App.js – Code**

<template>

<nav>

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

<router-link to="/lifecyclecapi">Life Cycle-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: left;

color: #2c3e50;

}

nav {

padding: 30px;

}

nav a {

font-weight: bold;

color: #2c3e50;

}

nav a.router-link-exact-active {

color: #42b983;

}

</style>

**Option API code – LifeCycleOptionsAPI.vue**

<template>

<div>

<h3> Life Cycle Hooks Demo - Options API </h3>

Count Value : {{count}}

</div>

<div>

<button @click.prevent="count++"> Click Me </button>

</div>

</template>

<script>

export default{

data(){

return{

count : 0

}

},

beforeCreate(){

console.log('Before create called');

},

created(){

console.log('Created');

},

beforeMount(){

console.log('Before Mounted');

} ,

mounted(){

console.log('Mounted');

},

beforeUpdate(){

console.log('Before updated')

},

updated(){

console.log('updated')

},

beforeUnmount(){

console.log('Before Unmount')

},

unmounted(){

console.log('unmounted')

}

}

</script>

**Composition API code – LifeCycleCompositionAPI.vue**

<template>

<div> <h2> Life Cycle Hooks - Composition API </h2></div>

<div> Count Value : {{ state.count }}

<br />

<button @click.prevent="state.count++">Click Me </button>

</div>

<div v-if="showComp">

<AboutView />

</div>

<div>

<button @click="showComp=!showComp" > Toggle Show Component </button>

</div>

</template>

<script>

import { onBeforeMount,

onBeforeUnmount, onBeforeUpdate,

onMounted, onRenderTracked, onRenderTriggered, onUnmounted,

onUpdated , reactive, ref } from 'vue'

import AboutView from '@/views/AboutView.vue';

export default{

setup() {

let showComp = ref(true);

const state = reactive({

count: 0

});

onBeforeMount(() => {

console.log("Before Mounted");

});

onMounted(() => {

console.log("On mounted");

});

onBeforeUpdate(() => {

console.log("On Before Updated");

});

onUpdated(() => {

console.log("On Updated");

});

onBeforeUnmount(() => {

console.log("On Before Unmounted");

});

onUnmounted(() => {

console.log("On Unmounted");

});

onRenderTracked((e) => {

console.log(e);

});

onRenderTriggered((e) => {

console.log(e);

});

return { state, showComp };

},

components: { AboutView }

}

</script>

<style scoped></style>