Section 1: Composition API and Testing Basics

Create posts.ts file inside store folder

import { reactive, readonly } from "vue";

// ref number, string

// computed

// reactive {}, Map, Set

interface PostsState {

  foo: string;

}

export class PostsStore {

  #state: PostsState;

  constructor() {

    this.#state = reactive<PostsState>({

      foo: "foo",

    });

  }

  getState() {

    return readonly(this.#state);

  }

  updateFoo(foo:string){

    this.#state.foo = foo

  }

}

const store = new PostsStore()

export function usePosts(){

    return store

}

Timeline in view folder

<script lang="ts" setup>

import { DateTime } from 'luxon';

import { ref, computed } from 'vue';

import { Post, today, thisWeek, thisMonth } from "../posts.ts";

import TimelineItem from '../components/26timelineItems';

import {usePosts} from '../store/posts.ts';

const postStore = usePosts();

const periods=["Today","This Week","This Month"] as const

type Period = typeof periods[number]

const selectedPeriod = ref<Period>("Today")

function selectPeriod(period:Period) {

  console.log("called");

  selectedPeriod.value = period

}

// const posts:Post[]=[today,thisWeek,thisMonth] //simple string

//For Date Formate

const posts:Post[]= computed(()=>{

  return [today,thisWeek,thisMonth]

.map(post=>{

  return{

    ...post,created:DateTime.fromISO(post.created)

  }

})

.filter(post=>{

  console.log("selectedPeriod",selectedPeriod);

  if (selectedPeriod.value === "Today") {

    return post.created>= DateTime.now().minus({day:1})

  }

  if (selectedPeriod.value === "This Week") {

    return post.created>= DateTime.now().minus({week:1})

  }

  return post

})

})

</script>

<template>

{{ postStore.getState().foo }}

<button @click="postStore.updateFoo('Testing')">Click </button>

<a v-for="period of periods"

:key = "period"

:class = "{'is-active':period == selectedPeriod}"

@click="selectPeriod(period)"

>

  {{ period }}

</a>

<TimelineItem v-for="post of posts"

:key = "post.id"

:post = "post"

/>

</template>

<style>

.is-active{

  font-size:20px;

  font-weight:bold;

  color:"green"

}

.panel{

  display:block;

  width:50%;

  margin:10px auto;

  padding:10px;

  box-shadow:2px 2px 8px grey

}

</style>

For PINIA

Needs to add