Bernardo, Jordan – August 17, 2022

CIT 510-IT41S2 - Integrative Programming and Technologies 2 Mr. Enrico Chavez

Latest Vue 3 features

a) Teleport

- Formerly known as Portal, Teleport enables us to select the parent element that a piece of HTML code should be placed beneath in order to be rendered.

b) Some of the components breaking changes

- The new "emit" option works similarly to existing properties if we want to define events in a component that this component will emit to its parent.

- Use the define AsyncComponent function to define an asynchronous component.

c) Fragments

- With Vue 3, we may now have several nodes in the root instead of having to wrap root elements in a single main element.

d) Filters were removed from Vue 3

- The usage of filters is no longer supported; instead, computed properties or methods can be used.

e) Experimental Suspense feature

- With the Suspense component in Vue 3, we can implement slow loading of components. While our user is waiting for the data, the <Suspense> component enables us to give some fallback information, such as a spinner or some text.
-

f) Experimental state-driven CSS variables

- Vue 3 allows us to link a value from the component state to any CSS property by using the v-bind() method in the style tag of a single file component. This enables us to modify the value of various CSS attributes on the fly.

g) Single file component <style scoped> changes

- If we utilize scoped styles in SFC, the CSS only applies to the currently selected component. Developers may use this functionality to give more standardized custom CSS in single file component scoped styles.

h) Multiple v-models

- Vue.js's v-model is used for two-way binding. Of course, the form elements are the most common way to use it. Unlike Vue 3, which enables us to send numerous v-

models to our components by naming them, Vue 2 only allows us to utilize one v-model per component.

i) Lifecycle naming changes

- The `destroyed` lifecycle was renamed to `unmounted`. The `beforeDestroy` was changed to `beforeUnmount`.

j) <template> render changes

- The `<template>` tag will only render its content in the new version if it has a specific directive; otherwise, it will be handled as a standard HTML element and will result in a standard `<template>` tag.

Vuex:

For Vue.js applications, Vuex is a state management pattern and library. With rules guaranteeing that the state may only be changed in a predictable way, it acts as a central repository for all the components of an application.

Pinia:

Pinia is a store library for Vue that enables state sharing between components and pages. If you're familiar with the Composition API, you might assume that sharing a global state is already possible by using a straightforward export const state = reactive({}). While this is true for single page apps, server-side rendering exposes your application to security flaws. Using the Composition API, Pinia provides a user-friendly, type-safe, and versatile Store for Vue that also supports DevTools.

Summarization:

Vuex assists us in managing shared state at the expense of additional ideas and boilerplate. A trade-off must be made between immediate and long-term production.

If you go into Vuex without having constructed a large-scale SPA before, it could seem intimidating and verbose. That's quite typical; if your project is straightforward, you'll probably be OK without Vuex. You could only require a basic store layout. However, if you are developing a medium-to-large SPA, it's likely that you have encountered circumstances that have made you consider how to manage state more effectively outside of your Vue components. In this case, Vuex will be the logical next step for you.

The support for Vuex 3 and 4 will continue. It's unlikely to see any new features added, though. Installation of Vuex and Pinia is possible in the same project. It could be a good alternative if you're moving an existing Vuex app to Pinia. However, we strongly suggest utilizing Pinia if you're starting a new project.

Honor Pledge:

"I affirm that I have not given or received any unauthorized help on this assignment and that this work is my own."

JORDAN BERNARDO