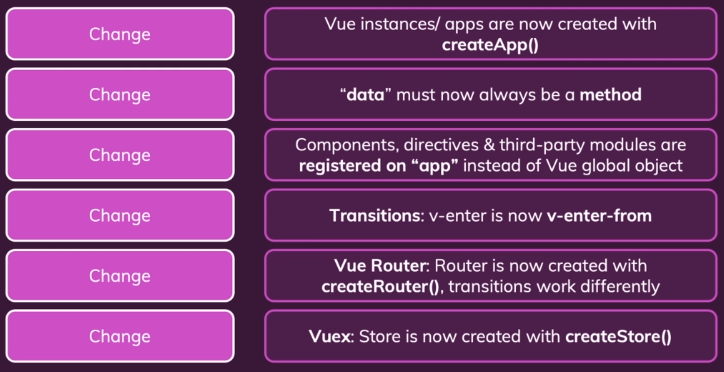
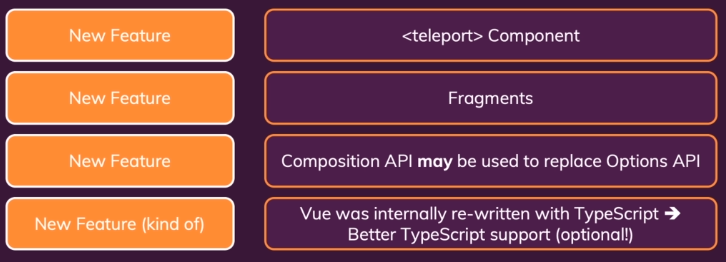
**Section 22 – Vue 2 To Vue 3 Migration**

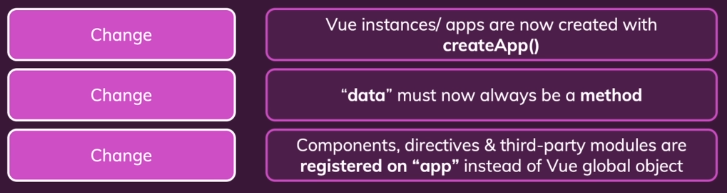
Chapter 317 – Vue 3 – Overview





Chapter 318 – Important Changes And Migration Steps

1. Basic App V2 And Basic App V3



Index.html V2

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>A Basic Vue App</title>

</head>

<body>

  <div id="app">

    <h2>{{ message }}</h2>

    <the-button @update="changeMessage"></the-button>

  </div>

  <script src="https://unpkg.com/vue"></script>

  <script src="app.js"></script>

</body>

</html>

Index.html V3

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>A Basic Vue App</title>

</head>

<body>

  <div id="app">

    <h2>{{ message }}</h2>

    <the-button @update="changeMessage"></the-button>

  </div>

  <script src="https://unpkg.com/vue@next"></script>

  <script src="app.js"></script>

</body>

</html>

App.js V2

Vue.component('the-button', {

  template: '<button @click="updateMessage">Click me</button>',

  methods: {

    updateMessage() {

      this.$emit('update');

    },

  },

});

new Vue({

  el: '#app',

  data: {

    message: 'This works in Vue 2!',

  },

  methods: {

    changeMessage() {

      this.message = 'Will it work in Vue 3?';

    },

  },

});

App.js V3

const app = Vue.createApp(

  {

    data() {

      return { message: 'This works in Vue 2!' };

    },

    methods: {

      changeMessage() {

        this.message = 'Will it work in Vue 3?';

      },

    },

});

app.component('the-button', {

  emits: ['update'],

  template: '<button @click="updateMessage">Click me</button>',

  methods: {

    updateMessage() {

      this.$emit('update');

    },

  },

});

app.mount('#app');

1. Composition API
2. Router V2 And Router V3



Main.js V2

import Vue from 'vue';

import VueRouter from 'vue-router';

import App from './App.vue';

import WelcomeScreen from './pages/WelcomeScreen.vue';

import UsersList from './pages/UsersList.vue';

Vue.use(VueRouter);

new Vue({

  router: new VueRouter({

    mode: 'history',

    routes: [

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

      { path: '/users', component: UsersList },

    ],

  }),

  render: (h) => h(App),

}).$mount('#app');

Main.js V3

import { createApp } from 'vue';

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

import App from './App.vue';

import WelcomeScreen from './pages/WelcomeScreen.vue';

import UsersList from './pages/UsersList.vue';

// Add router!

const router = createRouter({

    history: createWebHistory(),

    routes: [

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

        { path: '/users', component: UsersList },

    ],

});

const app = createApp(App);

app.use(router);

router.isReady().then(() => {

    app.mount('#app');

});

App.js V2

<template>

  <div>

    <the-header></the-header>

    <transition name="route" mode="out-in">

      <router-view></router-view>

    </transition>

  </div>

</template>

App.js V3

<template>

  <div>

    <the-header></the-header>

    <router-view v-slot="slotProps">

      <transition name="route" mode="out-in">

        <component :is="slotProps.Component"></component>

      </transition>

    </router-view>

  </div>

</template>

1. Teleport Fragments
2. Transitions V2 And Transitions V3



Transitions V2

new Vue({

  el: '#app',

  data: {

    isVisible: true

  },

  methods: {

    toggleBox() {

      this.isVisible = !this.isVisible;

    }

  },

});

Transitions V3

Vue.createApp({

  data() {

    return { isVisible: true };

  },

  methods: {

    toggleBox() {

      this.isVisible = !this.isVisible;

    }

  },

}).mount('#app');

Style.css V2

.box-fade-enter,

.box-fade-leave-to {

  opacity: 0;

  transform: translateY(-30px);

}

.box-fade-enter-active,

.box-fade-leave-active {

  transition: all 0.3s ease;

}

.box-fade-enter-to,

.box-fade-leave {

  opacity: 1;

  transform: translateY(0);

}

Style.css V3

.box-fade-enter-from,

.box-fade-leave-to {

  opacity: 0;

  transform: translateY(-30px);

}

.box-fade-enter-active,

.box-fade-leave-active {

  transition: all 0.3s ease;

}

.box-fade-enter-to,

.box-fade-leave-from {

  opacity: 1;

  transform: translateY(0);

}

1. Vuex V2 And Vuex V3



Main.js V2

import Vue from 'vue';

import Vuex from 'vuex';

import App from './App.vue';

Vue.use(Vuex);

const store = new Vuex.Store({

  state() {

    return {

      users: [],

    };

  },

  mutations: {

    addUser(state, payload) {

      const newUser = {

        id: new Date().toISOString(),

        name: payload.name,

      };

      state.users.push(newUser);

    },

  },

  actions: {

    addUser(context, payload) {

      context.commit('addUser', payload);

    },

  },

  getters: {

    users(state) {

      return state.users;

    },

  },

});

new Vue({

  store: store,

  render: (h) => h(App),

}).$mount('#app');

Main.js V3

import { createApp, createStore } from 'vue';

import App from './App.vue';

const store = createStore({

    state() {

      return {

        users: [],

      };

    },

    mutations: {

      addUser(state, payload) {

        const newUser = {

          id: new Date().toISOString(),

          name: payload.name,

        };

        state.users.push(newUser);

      },

    },

    actions: {

      addUser(context, payload) {

        context.commit('addUser', payload);

      },

    },

    getters: {

      users(state) {

        return state.users;

      },

    },

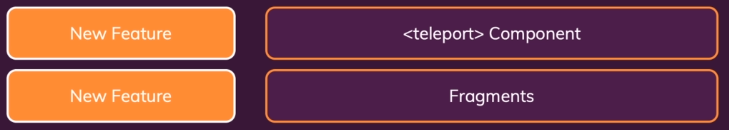
});

const app = createApp(App).mount('#app');

app.use(store);

app.mount('#app');

Chapter 319 – Vue 3 New Features



1. Using Teleport = ConfirmDialog.vue

<template>

  <teleport to="body">

    <div>

      <div class="backdrop"></div>

      <dialog open>

        <header>

          <h2>Are you sure?</h2>

        </header>

        <section>

          <p>Do you want to proceed with this operation?</p>

        </section>

        <menu>

          <button @click="cancel">No</button>

          <button @click="confirm">Yes</button>

        </menu>

      </dialog>

    </div>

  </teleport>

</template>

<script>

export default {

  emits: ["cancel", "confirm"],

  methods: {

    cancel() {

      this.$emit("cancel");

    },

    confirm() {

      this.$emit("confirm");

    },

  },

};

</script>

1. Fragments = AdminMode.vue

V2

<template>

<li>

  <confirm-dialog v-if="forceConfirm" @cancel="cancelChange" @confirm="confirmChange"></confirm-dialog>

    <h2>{{ modeName }}</h2>

    <button @click="deactivate">Deactivate</button>

  </li>

</template>

V3

<template>

  <confirm-dialog v-if="forceConfirm" @cancel="cancelChange" @confirm="confirmChange"></confirm-dialog>

  <li>

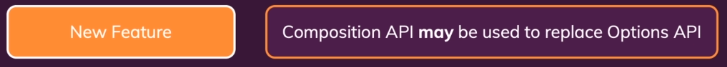
    <h2>{{ modeName }}</h2>

    <button @click="deactivate">Deactivate</button>

  </li>

</template>

Chapter 320 – The New Composition API



No Note

Chapter 321 – Further Migration Resources

No Note

Chapter 322 – About This Course And Vue 3

No Note

Chapter 323 – About This Course And Vue 3 (Written Summary)

No Note