Vue JS (v3), Vuetify, Vue Router, Pinia, Unit Testing BOOTCAMP

VUE-204

OVERVIEW

4 days for seasoned developers

Vue.js (Vue for short) is a lightweight frontend framework, offering easy two-way data binding, and a reactive UI, all from a tiny code footprint. Both Vue 3, along with the differences between Vue 2 and Vue 3, shall be covered. The focus will be on Vue 3, and Vue 2 will only be briefly covered.

Vue Router is a routing library for Vue-based Single Page Applications (SPA). Vue Router v4 will be covered. Pinia is a state management tools used in Vue applications.

TypeScript shall be used to build the running case study.

PREREQUISITES

- Basic HTML, CSS knowledge
- Basic knowledge of JavaScript – primitive data types, objects, arrays, functions.

  NOTE: Passing functions, array iterator methods, asynchronous programming, ES2015 features including rest, spread, arrow functions, modules, Promises and async..await will be covered in the training

RUNNING CASE STUDY – WORKSHOPS APP

At the end of this bootcamp, participants will build a workshops application using Vue. A backend server shall be provided. The app will list workshops, details of workshops, their list sessions, and allow authenticated users to post sessions, and also upvote/downvote on sessions.

LIST OF SOFTWARE TO BE INSTALLED BEFORE TRAINING BEGINS

1. Git CLI on participant systems and GitHub account should be created for every participant (to be created individually by participant). The **GitHub account should be a personal one** and not one associated with the company's GitHub account (I will not be able to add a company account as collaborator on my repositories, and hence shall not be able to share code).

   Git CLI download: https://git-scm.com/downloads

   GitHub link for account creation: https://github.com/join?source=header-home

   Once an account is created by everyone, **the list of GitHub user names needs to be shared with me** - I will add them to the GitHub repository before start of training.

2. Node.js needs to be installed on all systems – Mac OSX, Linux and Windows is supported. The 22.x.x (LTS version) may be installed. This will also install npm.

   Node.js https://nodejs.org/en/download/

3. Download and install Visual Studio Code (VSCode)

   https://code.visualstudio.com/download

   It is available for Windows, Mac OSX and popular Linux distributions.

4. For browser – latest version of one of Chrome or Firefox, **preferably Chrome**. Internet Explorer is not acceptable.

   Chrome: https://www.google.com/chrome/browser/desktop/index.html

   Firefox: https://www.mozilla.org/en-US/firefox/new/

5. **Additionally, it would be great if participants have as little restrictions (as permissible) on internet access during the session**

CHAPTERS AND TOPICS

Day 1

Select Topics in JavaScript (0.5 days / 1.5 days)

This section on JavaScript provides only a brief refresher, with no hands-on (instructor will code and demonstrate)

Overview of select features of JavaScript

- Functions as first-class citizens
- Functions as arguments, callbacks
- Array iterator methods
- Asynchronous programming
- Block-level scoping and the use of let, const
- Template strings
- Object and Array Destructuring
- Rest and spread operators (includes object spread)
- Arrow Functions
- Modules
- Promises
- async..await

Overview of TypeScript

This section on TypeScript provides only a brief refresher, with no hands-on (instructor will code and demonstrate)

- Installation and getting started
- The tsc compiler options and configuration using tsconfig.json file
- Primitive types and the any type
- Static type checking, type inference, type narrowing, type assertion
- Arrays
- Type Aliases
- Union types
- Defining function argument and return types
- Function signatures involving callback functions
- Using interface to define structure for an object (properties and methods)
- Implementing interfaces in classes
- Using Generics

Utility types – Omit, Partial, Pick

Introduction to Vue

What and Why

Installation - Gearing up for Vue application development

Scaffolding an application using Vue CLI

Understanding the project structure

Understanding the build process

More about the Vue CLI

How it works internally - basic introduction to Webpack /. Vite

Entrypoint, setting up bundles

Introduction to the running example: Workshops Application

Data binding, Conditionals, Looping and Lists

Mounting the app

Data binding

Attribute binding

Directives: v-show, v-if, v-else, v-else-if

Iteration using v-for, key

Binding HTML – v-html

Showing list of workshops

Binding classes

Vue internals and Vue 3 APIs

How the Vue compiler and reactivity works

Virtual DOM

Significance of key

Options and composition API

Vue 3 composition API – setup(), ref(), reactive(), computed(), watch(), lifecycle related methods vs Options API options (each composition API method will be covered in details later)

Day 2

Adding Interactivity

Methods

Binding to DOM events

Using computed properties

Forms and Inputs

Using v-model binding

Binding values to check boxes and radio buttons

Modifiers: .number, .lazy and more

Working with components

Creating components

Global registration

Local registration

Relationships between components

Composing components

Using props to pass data

$emit() vs function prop

$attrs

Prop validation

Defining a template component - inline strings, template in script tag, and using single file components

Working with custom events – Listening to events, modifying child props using .sync

Lifecycle hooks

Using watch

Computed properties vs watch

Day 3

Advanced Components and Routing

- Working with slots – named slots, scoped slots
- Async components
- Adding Axios
- Setting up routes
- Router links
- Adding routes with parameters
- Using redirection and wildcards

Extending Vue

- Reusing functionality using Composition API
- Custom directives – modifiers, values and args

State Management

- When are state management tools helpful?
- Introduction to Pinia
- Getting started and injecting the store into the app
- Working with Pinia
- State, Getters, Actions, Plugins

Day 4

Case study: Workshops Application development (Vue, Vue router, Vuetify, Pinia, Authentication and Authorization)

Vue 2 vs Vue 3

- How composition API is to be used for code reuse between components
- Problem with mixins
- Migrating an app from Vue 2 to Vue 3

Basic Introduction to Unit Testing

What is unit testing and why it is required

Configure Jest in application

Writing a test – describe(), it()/test(), beforeEach()/afterEach(), beforeAll(), afterAll()

Test coverage

Vue Test Utils

How Vue Testing Library helps

Component tests

render, screen, getBy*(), findBy*(), queryBy*() methods – when to use which methods

Jest DOM for DOM matchers, User event library

Testing services – using MSW for mocking API calls

Case study: Workshops App using Vue 3 and Unit Testing

Debugging

How to go about looking for errors

The Vue developer tools

Using the developer tools for Pinia

Deployment

Configuring the app for different environments

Deployment options

Case study: Workshops Application development (Deployment on AWS S3