Course Outline

Professional Vue.js

Course Summary

Description

In this 5-day course, students will start with the fundamental concepts and structures of Vue.js development, and quickly progress to learning about and solving problems with the most up-to-date techniques and tools in Vue.js development, including: Components, Routing, and Vuex.

Professional Vue.js is a complete introduction to modern front-end development. It covers the syntax, conventions, and best practices of modern JavaScript development while also teaching Vue.js. Through a series of hands-on exercises and demonstrations, students will learn how to develop and test complete Vue applications.

Objective

After taking this course, students will be able to:

- Install, configure, and use modern web tooling
- Learn to create test suites for Vue
- Write ES2018 code and compile it using Babel
- Understand what Vue.js is and what problem it solves
- Explore the basic architecture of a Vue.js application
- Gain a deep knowledge of Vue.js components
- Build a Single Page Application with Vue Router
- Use Vuex for maintaining state in a Vue.js application
- Use Local Storage with Vue
- Render Vue.js on the server
- Learn Vue.js best practices

Audience

This course is designed for experienced front-end web developers who wish to begin using Vue.

Topics

- Introduction and Vue Quickstart
- Development Ecosystem
- Intro to Node.js
- Node.js and npm
- Mixins
- Debugging JavaScript
- Tooling
- Getting Started With Vue.Js
- Basic Vue.Js Features
- All About Components
- Using Vuex
- AJAX and Vue
- Single-Page Applications
- Unit Testing and End-To-End Testing
- Organize + Automate + Deploy = Webpack
- Asynchronous JavaScript (Time permitting)
- Transitions and Animations (Optional)

Prerequisite

Before taking this course, students should have solid experience with HTML5, CSS3, and JavaScript

Duration

Five Days

Course Outline

Professional Vue.js

Course Outline

I. Introduction and Vue Quickstart
A. Vue Hello World
B. Using vue-cli
C. Lab: Modifying Components and Tests

II. Development Ecosystem
A. Code Editors and IDEs
B. Installing and Configuring Visual Studio Code

III. Intro to NodeJS
A. What is Node?
B. A history of Node
C. How to write and run a Node script

IV. Node.js and npm
Getting Started with Node.js and npm

V. Debugging JavaScript
A. Using Chrome's Developer Tools
B. Installing and Using the Vue DevTools
C. Debugging in Chrome

VI. Tooling
A. Why have tooling?
B. npm and yarn
C. package.json
D. eslint and jslint
E. transpilers
F. webpack

VII. Getting Started With Vue.Js
A. History Of Vue.Js
B. What Makes Vue Different?
   1. Vue Vs React
   2. Vue Vs Angular
C. Vue's Virtual DOM
D. Vue Versions
   1. What To Expect In Vue 3

VIII. Basic Vue.Js Features
A. Understanding the Vue Instance
   1. Instance properties and methods
B. Introduction to Vue's Directives
   1. Arguments
   2. Modifiers
   3. Shorthands
C. Vue Lifecycle
   1. created
   2. mounted
   3. updated
   4. destroyed
   5. Using Lifecycle Methods
D. Event Handling
   1. Inline Event Handling
   2. Event Handling with Methods
   3. Event Modifiers
E. Vue Template Syntax
   1. Using HTML
   2. Using JSX
   3. Using CreateElement
   4. Attributes
F. Loops and Lists
   1. Using key
   2. Using v-for
      a. v-for with Arrays
      b. v-for with Objects
   3. Mutating arrays and objects
      a. Array and Object Change Detection
      b. The mutations Vue Can't Detect
G. Conditional Rendering
   1. v-if vs. v-show
H. Using Filters
   1. Formatting Currencies with Filters
   2. Formatting Dates with Filters
I. Computed Properties
   1. Benefits of Computed Properties
   2. Filtering a List with a Computed Property
   3. Sorting a List with a Computed Property
   4. Using setters in computed properties
J. Creating custom watchers
   1. Async operations with watchers

Course Outline

Professional Vue.js

Course Outline(cont.)

K. Binding HTML Classes
   1. Displaying and Hiding an Element Conditionally
   2. Adding Styles Conditionally
   3. Binding Inline Styles
L. Outputting Raw HTML
M. Vue and Forms
   1. Input Bindings
      a. value binding
      b. 2-way binding
   2. Creating a Form with Checkboxes
   3. Creating a Form with Radio Buttons
   4. Creating a Form with a Select Element
   5. Input Modifiers

IX. All About Components

A. Creating a Component
B. Organizing and Reusing Components
C. Event Handling with Components
D. Passing Data
   1. Using Props
   2. Using Events
   3. Using v-model
E. Reading a Child's State
F. Using Components in Your Own Components
G. Content Distribution with Slots
H. Single File Components with Webpack
I. Loading Your Components Asynchronously

X. Mixins

A. Basics
B. Option Merging
C. Global Mixin
D. Custom Option Merge Strategies

XI. Using Vuex

A. Understanding the Flux Pattern
B. The Store
   1. Creating a Store
   2. store.state
   3. store.commit
C. Using Vuex State in Vue Components
D. Using mapState
E. Store getters
   1. Using Getters
   2. Mapping Getters
F. Creating Mutation Types

G. Synchronous Mutations
   1. Mutation handlers
   2. Passing data to store.commit
      a. Single value
      b. Object-style commit
   3. Committing Mutations in Components
H. Async Mutations with Actions
   1. Actions vs. Mutations
   2. Dispatching Actions
   3. Composing Actions
I. Splitting the Store with Modules
   1. Local State
   2. Namespacing
   3. Accessing Globals Assets from Namespaced Modules
   4. Registering Global Actions
   5. Binding Helpers
   6. Dynamic Registration
   7. Reusing Modules
J. Plugins

XII. AJAX and Vue

A. Sending Basic AJAX Requests with Axios
B. Validating User Data before Sending It
C. Recovering from an Error during a Request
D. Creating a REST Client (and Server!)
E. Implementing Infinite Scrolling
F. Processing a Request before Sending It Out

Course Outline

Professional Vue.js

Course Outline(cont.)

G. Preventing XSRF Attacks on Your App

XIII. *Single-Page Applications*

A. Creating an SPA with Vue-Router
B. Fetching Data before Switching Route
C. Managing Errors for Your Routes
D. Adding a Progress Bar to Load Pages
E. Using Params
1. Reacting to Params Changes
F. Having More Than One Router-View in Your Page
G. Nested Routes
H. Navigating Programmatically
I. Using Named Dynamic Routes
J. Adding Transitions between Your Routes
K. How to Redirect to Another Route
L. Passing Props to Route Components
M. Route History Manipulation

XIV. *Unit Testing and End-To-End Testing*

A. Unit Testing with Jest or Mocha
B. Vue Test Utils
C. Adding Karma to Your Workflow
D. Writing Testable Components
E. Testing Your Application State and Methods
F. Testing DOM Asynchronous Updates
G. End-to-end testing with TestCafe
H. Stubbing External API Calls with Sinon.JS
I. Measuring the Coverage of Your Code

XV. *Organize + Automate + Deploy = Webpack*

A. Extracting Logic from Your Components to Keep the Code Tidy
B. Organizing Your Dependencies with Webpack
C. Using External Components in Your Webpack Project
D. Developing with Continuous Feedback with Hot Reloading
E. Running a Code Linter While Developing
F. Releasing Your Components to the Public

XVI. *Asynchronous JavaScript (Time permitting)*

A. Multithreading
B. How the event loop works
C. Promises
D. Async and await

XVII. *Transitions and Animations (Optional)*

A. Integrating with Third-Party CSS Animation Libraries
B. Adding Your Own Transition Classes
C. Animating with JavaScript Instead of CSS
D. Transitioning on the Initial Render
E. Transitioning Between Elements
F. Letting an Element Leave Before the Enter Phase in a Transition
G. Adding Entering and Leaving Transitions for Elements of a List
H. Transitioning Elements That Move in a List
I. Animating the State of Your Components
J. Dynamic Transitions