

Developing Applications with Vue.js

Duration: 5 Days

Description

Vue is a progressive framework for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries.

Take your HTML, CSS, and JavaScript skills to the next level by learning one of the hottest JavaScript frameworks available today, Vue.js! Vue.js is a fun and easy to use JavaScript framework for building user interfaces. The core of the framework is lightweight and fast and can be used in projects of any size. In this course, you'll build two fun, challenging projects to learn Vue.js and apply everything instantly.

Delegates will learn

- the basics of Vue.js, including its use of templates to display data on a page
- to set up Vue projects from scratch, or use vue-cli to set up from a template
- to create instances, learn about single page apps, Custom directives, Mixins, Routing and more
- to effectively use and navigate the Vuex library
- the fundamental concepts of Vue.js and how to use it in applications
- to discover how Vue.js works with CSS to style your websites and applications
- to use render functions and JSX, rather than templates, to determine what Vue displays
- to control how code is executed and displayed with vue-router
- to build a fully functional app from scratch, and unit test your applications

Audience

This course is for web developers who have little or no prior experience with Vue.js. It mainly targets JavaScript enthusiasts who want to learn a modern and simple JavaScript framework.

Outline

Introduction to Vue.js

- What is Vue.js
- Why Vue.js
- Quick tour of Vue-cli
- Hello World with Vue.js
- Anatomy of Hello world vue app

Working with Template & viewmodel

- What is vue.js template
- What is viewmodel / vue instance?
- Template syntax
- Interpolation
- Directive
- Filters
- Shorthands
- Javascript expressions
- Computed properties
- Watchers
- Class and style bindings
- HTML classes
- Inline styles
- Conditional Rendering
- Iterative Rendering

Vue Instance Revisited

- Basic vue instance usage
- Multiple vue instances
- Vue instance Lifecycle
- Accessing data from outside vue instance
- Data and methods
- More about \$el, \$data and \$refs

Event handling

- Introduction
- Listening to events
- Method Event handlers
- Event modifiers

- Key modifiers

Working with forms & two way binding

- Introduction to two way binding via v-model
- Handling user input with HTML form controls
- Data binding
- .lazy, .number, .trim modifiers
- Submitting a form

Introduction to components

- What are components?
- Why components
- Hello World component
- Anatomy of a component
- Registering Components Locally and Globally
- Root Component
- Thinking in components
- Composing application as components

Passing data to components

- Props, parent to child communication
- Custom events, child to parent
- Non parent child communication event emitter style using .emit, .on

Content distribution with slots

- Compilation scope
- Single slot
- Named slot
- Scoped slot

Dynamic components

- Switching Multiple Components with Dynamic Components
- Dynamic Component Behavior
- Keeping Dynamic Components Alive
- Lifecycle hooks

Custom directives

- Directive hook functions
- Developing custom directive
- Passing values and arguments to custom directives

- Directive hook arguments

Mixins

- Creating and using mixins
- Option merging
- Global mixin

Filters

- Local filters
- Global filters
- Chaining filters

Remote communication with vuejs

- Using vue-resource to working with HTTP
- Fetch data
- Post data to server
- Request Response Interception
- Template URLs

Routing

- Setting up vue-router
- Loading routes
- Routing modes
- Navigation with router links
- Styling active links
- Imperative navigation
- Route Parameters
- Setting up child or nested routes
- Named routes
- Query parameters
- Named router views
- Wildcard routes
- Route guards

State Management

- What is need for state management
- What is vuex
- Centralized store
- Understanding Getters

- Using Getters
- Mapping Getters to Properties

Understanding Mutations

- Using Mutations
- Using Actions
- Mapping Actions to Methods

Unit Testing

- Setup and tooling
- Testing components

Prerequisites

The knowledge of basic HTML & CSS & Javascript skills