

Tworzenie aplikacji z użyciem VueJS

VUEJS

Czas trwania: 3 dni

Vue wyróżnia się spośród frameworków progresywnym podejściem - możemy zacząć od prostych widżetów a skończyć na złożonej aplikacji single page

Cele szkolenia

- Zapoznanie uczestników z dobrymi praktykami programowania w najnowszej wersji języka JavaScript oraz praktycznymi zagadnieniami związanymi z tworzeniem aplikacji internetowych w architekturze "Single-page application" przy użyciu frameworka Vue.js
- Konstruowanie czytelnych i wydajnych aplikacji JavaScript z wykorzystaniem aktualnych możliwości tego języka
- Używanie elementów programowania funkcyjnego i reaktywnego do rozwiązywania typowych problemów
- Prawidłowe i efektywne skonfigurowanie środowiska do pracy z zaawansowanymi projektami frontendowymi
- Stosowanie mechanizmów pozwalających kontrolować jakość kodu aplikacji JavaScript
- Stworzenie aplikacji opartej na komponentach
- Samodzielne tworzenie zaawansowanych aplikacji webowych

Zalety

- Szkolenie VueJS - tworzenie aplikacji webowych wyróżnia szeroki zakres przedstawionego materiału przy jednoczesnym skupieniu na aspekcie praktycznym
- Największy nacisk położony jest na techniki i narzędzia, które sprawdzają się w rzeczywistych projektach i przynoszą największy wzrost wydajności pracy
- Część warsztatowa szkolenia pozwala na osiągnięcie poziomu umiejętności pozwalających na samodzielne wykorzystanie zdobytej wiedzy od razu po zakończeniu kursu
- Kameralne grupy - szkolenia technologiczne prowadzimy w grupach liczących do 8 osób. Pozwala to na indywidualne podejście oraz aktywizację każdego uczestnika
- Praktyka przed teorią - wszystkie szkolenia technologiczne prowadzone są w formie warsztatowej. Konieczna teoria jest wyjaśniana na przykładzie praktycznych zadań
- Konkretność umiejętności - w ramach każdego szkolenia rozwijamy praktyczne umiejętności związane z daną technologią i tematyką
- Nauka z praktykami - wszyscy trenerzy na co dzień pracują w projektach, gwarantuje to dostęp do eksperckiej wiedzy i praktycznego know-how



Dla kogo?

- Szkolenie VueJS - tworzenie aplikacji webowych przeznaczone jest dla programistów posiadających podstawową wiedzę w zakresie JavaScript, którzy chcą poszerzyć swoje kompetencje związane z tworzeniem zaawansowanych aplikacji internetowych działających po stronie przeglądarki

Wymagania

- Od uczestników wymagana jest przynajmniej podstawowa umiejętność programowania w języku JavaScript



Program

1. Wprowadzenie
 - a. Vue Framework, czy biblioteka?
 - b. Architektura aplikacji Vue
 - c. Virtual DOM
 - d. Podejście przyrostowe
 - e. Proste aplikacje z Vue.js
2. Narzędzia ekosystemu Vue
 - a. NodeJS i NPM
 - b. Webpack, vue-loader
 - c. Vue-CLI
 - d. TypeScript
 - e. Vue DevTools
 - f. Pliki .vue
 - g. Preprocesory CSS
3. Wiązania danych i widoku
 - a. Obiekt data i model reaktywny
 - b. Bindowanie tekstu i atrybutów
 - c. Atrybuty style i class
 - d. Dyrektywy v-bind, v-cloak, v-text i v-html
4. Manipulacja strukturą dokumentu
 - a. Ukrywanie z v-show, v-hide
 - b. Renderowanie warunkowe z v-if, v-else
 - c. Iteracje z v-for, zastosowanie atrybutu key
 - d. Dostęp do indeksu tablicy i klucza obiektu
 - e. Generowanie zakresów
 - f. Użycie v-for z tagiem template
5. Praca z obiektami i tablicami
 - a. Mechanizm wykrywania zmian Vue - jak działa?
 - b. Rozszerzanie modelu o nowe dane
 - c. Manipulacja tablicami
 - d. Pułapki i dobre praktyki
6. Obsługa zdarzeń i formularze
 - a. Dyrektywy v-on, v-once i wiązanie metod
 - b. Obiekt \$event, informacje o zdarzeniu
 - c. Modyfikatory zdarzeń
 - d. Obsługa różnych typów pól formularzy
 - e. Dyrektywa v-model
 - f. Zaawansowane możliwości v-model
7. Mechanizmy reaktywne i obserwowanie zmian
 - a. Obserwowanie danych z Watchers
 - Opcje immediate i deep
 - b. Computed - optymalizacja dynamicznych wartości



-Gettery i settery a computed

8. Komponenty

- a. Rejestrowanie komponentu - lokalne i globalne
- b. Własności komponentu - Props
 - Walidacja właściwości
 - Właściwości opcjonalne i wymagane
- c. Cykl życia komponentu
- d. Emitowanie własnych typów zdarzeń
- e. Dziedziczenie atrybutów i zdarzeń
- f. Rozszerzanie komponentów
- g. Komunikacja z v-model
- h. Obiekty referencji DOM
- i. Asynchroniczne i leniwe ładowanie

9. Projekcja treści

- a. Mechanizm Slots
- b. Wiele slotów, sloty nazwane
- c. Opcjonalne sloty, treść domyślna, ukrywanie slotów
- d. Bindowanie danych do slotów
- e. Zaawansowane przykłady użycia slotów

10. Metody komunikacji HTTP

- a. Fetch API - użycie i polyfill
- b. Vue-Resource - konfiguracja i przykład użycia
- c. Axios - użycie, konfiguracja i interceptor

11. Routing

- a. Koncepcja Single Page Application
- b. Hash routing, a browser history API
- c. Konfiguracja Vue-Router
 - Ścieżki
 - Nazwane ścieżki
 - Przekierowania
 - Obsługa błędów
- d. Przekazywanie parametrów
- e. Obiekty \$router i \$route
- f. Tworzenie odnośników
- g. Oznaczanie bieżącej pozycji
- h. Nawigacja programistyczna, push i replace
- i. Dynamiczne adresy
- j. Router hooks, router guards - kontrola dostępu
- k. Zaawansowane możliwości i przykłady

12. Zarządzanie stanem aplikacji

- a. Wprowadzenie do Vuex
- b. Store, Akcje, Mutacje, Stan i Gettery
- c. Zasady działania, dobre praktyki
- d. Integracja z komponentami \$store i helper



- e. Moduły i namespaces
- f. Przykłady i dobre praktyki
- 13. Animacje aplikacji opartych o Vue.js
- 14. Informacje dodatkowe
 - a. Vue Styleguide
 - b. Vue i TypeScript
 - c. SSR w Vue
- 15. TypeScript
 - a. Kiedy i dlaczego używać TypeScript
 - b. Vue CLI i TypeScript
 - c. Konfiguracja kompilatora
 - d. Komponenty jako klasy
 - e. Dekoratory komponentu
 - f. Props jako dekorator, dobre praktyki
 - g. Watch i Computed w TypeScript
 - h. Vue Stora i module jako klasy
 - i. Dekoratory dla Vuex Store
 - j. Definiowanie brakujących typów
- 16. Deployment
 - a. Budowanie i optymalizacja vue
 - b. Działanie routera na zdalnym serwerze
 - c. Optymalizacje i dobre praktyki

