

## Program szkolenia:

# Vue.js + Vuex + ES6

### Informacje:

<b>Nazwa:</b>	<b>Vue.js + Vuex + ES6</b>
<b>Kod:</b>	<b>Vue-vue</b>
<b>Kategoria:</b>	Vue.js
<b>Odbiorcy:</b>	developerzy, architekci
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	20% wykłady / 80% warsztaty

---

Szkolenie w formie warsztatowej z dużym naciskiem na pisanie własnego kodu. W trakcie warsztatów skupiamy się na:

- wydzielenie warstw aplikacji
- rozdzielanie odpowiedzialności dotyczącej: zarządzania komponentami, zarządzania stanem aplikacji, obliczeń i selektorów
- dzielenia aplikacji na reużywalne komponenty: komponenty bezstanowe, kontenery
- wydzielenia odpowiedzialności komponentów oraz projektowania zależności pomiędzy nimi
- rozróżniania i dzielenia stanu aplikacji ze względu na jeden z 5 podstawowych typów
- skalowalna struktura projektu
- skalowalny routing
- projektowania skalowalnej architektury stanu aplikacji

### Zalety szkolenia:

- dobrych praktyki dotyczących pisania eleganckiego, czystego kodu w tym
- tworzenia skalowalnych aplikacji klienckich
- dzielenia aplikacji na reużywalne komponenty

## Szczegółowy program:

### 1. Vue.js

#### 1.1. architecture overview, essentials

##### 1.1.1. Components basics

##### 1.1.1.1. data and methods

#### 1.2. Templates, templates syntax

##### 1.2.1. interpolations

##### 1.2.2. built-in directives

##### 1.2.3. conditional rendering

##### 1.2.4. list rendering

##### 1.2.5. browser events handling, events modifiers

##### 1.2.6. form inputs binding

#### 1.3. In-depth components:

##### 1.3.1. computed props and watchers

##### 1.3.2. global vs local registration

##### 1.3.3. lifecycle and Lifecycle Hooks

##### 1.3.4. props

##### 1.3.4.1. literal vs dynamic

##### 1.3.4.2. one-way data flow

##### 1.3.4.3. validation

##### 1.3.5. custom events, modifiers

##### 1.3.6. components composition

##### 1.3.7. cross components communication: props, events, slots

##### 1.3.8. state basics and state management

1.4. Components Unit Testing

1.5. Custom Directives

1.5.1. hook functions

1.5.2. hook arguments

1.6. Filters

1.7. Plugins

1.8. Unit Testing

1.9. Routing with vue-router

1.9.1. static and dynamic route matching

1.9.2. imperative and declarative navigation

1.9.3. props passing

1.9.4. guards

1.9.5. data fetching

1.9.6. lazy loading

## 2. Vuex

2.1. architecture overview

2.2. building blocks

2.2.1. State

2.2.2. Getters

2.2.3. Mutations

2.2.4. Actions

2.2.5. Modules

2.3. devtools-extension, Hot Module Reload

2.4. plugins

2.5. form handling

2.6. Unit Testing

### 3. Vue.js + Vuex

3.1. App modeling and design

3.2. Real apps

3.2.1. real projects challenges and problems

3.2.2. non trivial examples

3.2.3. best practices

3.2.4. style guides (project structure, conventions, dev environment and tools)

3.3. best additional external resources

### 4. Modeling App State

4.1. Components, Higher Order Components

4.2. separation of concerns

4.3. design principles

4.4. data architecture

4.5. unidirectional data flow

### 5. ES6

5.1. scopes

5.2. closures

5.3. contexts

5.4. arrow functions

5.5. classes

5.6. enhanced object literals

5.7. template strings

5.8. destructuring

5.9. default + rest + spread

5.10. let + const

5.11. iterators + for..of

5.12. generators

5.13. modules

5.14. module loaders