

# Tytuł szkolenia: Budowanie i testowanie aplikacji w oparciu o Angular

## Kod szkolenia: J-ANGULAR-T

### Wprowadzenie

**Angular** to najpopularniejsza obecnie **platforma programistyczna (framework)** do tworzenia aplikacji internetowych, działających po stronie przeglądarki w modelu SPA (Single-Page Application). Framework zyskał popularność wśród programistów, gdyż wielokrotnie skrócił pracę nad projektami, szczególnie w pracy zespołowej. Wersja pierwsza Angulara nie była jednak wolna od wad: z perspektywy czasu widać problemy z wydajnością, brak kompatybilności z ECMAScript 6 oraz mnogość wizji prawidłowego budowania aplikacji.

Angular został napisany praktycznie od podstaw gdyż bazuje na ES6. Autorzy platformy odpuścili takie koncepcje jak: kontrolery, \$scope, angular.module, ng-model. Za to zachęcają do używania TypeScriptu. W Angular możemy programować obiektowo, używać silnego typowania, adnotacji, klas i innych.

W ofercie Vavatech dostępne jest również [trzy dniowe szkolenie z Angular, które pomija temat testów aplikacji](#)

### Adresaci szkolenia

Szkolenie dedykowane jest dla osób znających podstawy **JavaScript**, **HTML** oraz **CSS**, które chcą poznać najnowsze narzędzie do budowy nowoczesnych **aplikacji webowych** działających po stronie przeglądarki - Angular.

### Cel szkolenia

Poznanie tajników **języka JavaScript** oraz wykorzystanie zdobytej wiedzy na wielu praktycznych przykładach.

### Czas i forma szkolenia

- 35 godzin (5 dni x 7 godzin), w tym wykłady i warsztaty praktyczne.

### Plan szkolenia

#### Jak działa Angular

- a. Komponenty
- b. Dekoratory komponentów
- c. Model aplikacji

#### Budowanie aplikacji - webpack

- a. Konfiguracja
- b. Kompilacja plików TypeScript
- c. Kompilacja plików SASS

#### Komponenty wbudowane

- a. NgIf
- b. NgSwitch
- c. NgStyle
- d. NgClass
- e. NgFor
- f. NgNonBindable

#### Własne komponenty i dyrektywy

- a. Wejście i wyjście
- b. Wstrzykiwanie zależności
- c. Style
- d. Zdarzenia: OnInit, OnChanges, DoCheck
- e. Zdarzenia: AfterContentInit, AfterViewInit

- f. Zdarzenia: AfterContentChecked, AfterViewChecked

### **Utworzenie prostej aplikacji w Angular**

- a. Utworzenie komponentu
- b. Zasilenie komponentu danymi
- c. Praca z kolekcją
- d. Powielanie wierszy w widoku
- e. Dodanie nowego produktu

### **TypeScript**

- a. Wprowadzenie do TypeScript.
- b. Silne typowanie
- c. Klasy
- d. Interfejsy
- e. Import oraz eksport modułów
- f. Adnotacje

### **Formularze w Angularze**

- a. Jak używać Controls i Control Groups
- b. Utworzenie pierwszego formularza
- c. Użycie FormBuilder
- d. Walidacja formularza
- e. Nasłuchiwanie na zmianach
- f. ngModel

### **Struktura danych ( Observables )**

- a. Wprowadzenie do RxJS
- b. Tworzenie obiektu Observable
- c. Łączenie zdarzeń z wielu różnych obiektów Observable
- d. Przetwarzanie danych obiektów Observable

### **Komunikacja z serwerem**

- a. HTTP
- b. angular/http API

### **Routing**

- a. Jak działa ruting w aplikacji Angular
- b. Utworzenie prostego rutingu
- c. Strategie rutingu
- d. Parametry rutingu
- e. Zagnieżdżanie rutingu

### **Konwersja aplikacji Angular 1 do aktualnej wersji Angulara**

- a. Jak używać komponentów angular 1.x w aktualnej wersji Angulara
- b. Tworzenie hybrydy

### **Utworzenie aplikacji (SPA)**

- a. Jak powinna wyglądać architektura aplikacji
- b. Pobranie danych z serwera
- c. Wyświetlenie listy produktów
- d. Usuwanie produktu
- e. Dodawanie produktu
- f. Edycja produktu

### **Testy "End to End":**

- a. Instalacja i konfiguracja
- b. Wprowadzenie do narzędzia Protractor
- c. Testowanie formularza
- d. Testowanie dodawania oraz usuwania produktu
- e. Debugowanie testów
- f. Logowanie błędów
- g. Dodatkowe narzędzia wspomagające pracę

### **Testy "Unit":**

- a. Instalacja i konfiguracja
- b. Wprowadzenie do narzędzia Karma
- c. Mockowanie API
- d. Testowanie komponentu
- e. Testowanie serwisu
- f. Testowanie filtra

**Debugowanie aplikacji:**

**Przygotowanie wersji produkcyjnej aplikacji:**

- a. Minimalizacja i konkatyzacja
- b. Automatyzacja zadań