

# NGRX

## Angular + TypeScript + Redux + RxJS

*Program szkolenia, Ducin IT Consulting*

Czas trwania: 3-5 dni

Formuła: 33% wykłady, 66% ćwiczenia

Szkolenie ma na celu zrozumienie architektury NGRX, celów jej stosowania, problemów, jakie rozwiązuje - a także efektywnej implementacji w aplikacjach Angularowych. Trener stawia przed uczestnikami problemy do rozwiązania polegające na dodaniu nowej funkcjonalności lub rozszerzeniu istniejącej (dopasowując je do poziomu grupy, aby były trudne, ale jednocześnie w zakresie możliwości). Omawiane są alternatywne rozwiązania (prywatny stan komponentów Angularowych, współdzielony stan w serwisach - oraz scentralizowany stan w NGRX) - i wybrane z nich - implementowane.

Szkolenie kładzie nacisk na dogłębne zrozumienie całej platformy. NGRX bazuje na aż czterech technologiach: frameworku Angular, języku TypeScript, bibliotece do zarządzania stanem Redux oraz bibliotece strumieni reaktywnych - RxJS. Uczestnicy wyrównują poziom wiedzy z zakresu ww. technologii (program indywidualnie dostosowywany pod każdą grupę). Omawiane są często popełniane błędy, a także rozwiązania, które ograniczają elastyczność i rozszerzalność aplikacji w przyszłości.

Szkolenie uwzględnia także ćwiczenia projektowania architektury/rozwiązań. Mają one na celu precyzyjną analizę problemu z jakimi uczestnicy będą mieli do czynienia w przyszłości oraz wybór optymalnego rozwiązania, zanim rozpocznie się implementacja.

## **Zalety Szkolenia:**

- zrozumienie i spójne połączenie filozofii reduxa oraz strumieni reaktywnych
- przykłady o złożoności odpowiadającej realnym problemom
- dobre praktyki oraz rozwiązania często spotykanych problemów

# Program szkolenia:

## 1. Building blocks

- 1.1. Actions, Action Types
- 1.2. Action Creators
- 1.3. Reducers
- 1.4. State
- 1.5. Store
- 1.6. Selectors
- 1.7. Effects
- 1.8. Metareducers

## 2. Tooling

- 2.1. Setup, Automation
- 2.2. DevTools - setup, demo, usecases

## 3. State Management

- 3.1. Private Component State
- 3.2. Shared Service State
- 3.3. Shared Centralized Store State

## 4. Architecture

- 4.1. Application Design and Architecture
- 4.2. Angular Component Architecture
- 4.3. Reactive State Changes
- 4.4. Event Sourcing
- 4.5. Redux vs NGRX
- 4.6. Pros and cons of NGRX

## 5. Functional Programming

5.1. Immutability

5.2. Pure Functions

5.3. Memoization

## 6. Usecases

6.1. Finding RxJS operators that fit certain UI

6.2. Implementation exercises

6.3. Design exercises