

## Program szkolenia:

# Architektura modularnego monolitu w .NET Core

### Informacje:

<b>Nazwa:</b>	<b>Architektura modularnego monolitu w .NET Core</b>
<b>Kod:</b>	<b>NET-arch-monolith</b>
<b>Kategoria:</b>	Architektura .NET
<b>Odbiorcy:</b>	developerzy, architekci
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	30% wykłady / 70% warsztaty

---

Szkolenie Modularny Monolit dedykowane jest dla osób i zespołów, które poszukują technik i wzorców wspomagających logiczny podział monolitycznej solucji na niezależne od siebie moduły, zapewniając kompromis pomiędzy zaletami rozwijania pojedynczej aplikacji, a modularności i autonomiczności niezależnych usług w podejściu mikroserwisowym.

Podczas szkolenia uczestnicy poznają techniki i wyzwania związane z podziałem monolitu na niezależne części (tzw. vertical slice) oraz skupią się na przykładowej implementacji modułu w dedykowanym projekcie.

### Zalety szkolenia:

- Pragmatyczne podejście do modularyzacji
- Przygotowanie do wydzielania microservices
- Zdrowe granice modułów

## Szczegółowy program:

### 1. Teoria

- 1.1. Podział systemów w kontekście modularności, a rozproszonej
- 1.2. "Klasyczny" vs. modularny monolit
- 1.3. Zalety implementacji modularnego monolitu

### 2. Architektura aplikacji

- 2.1. Czysta architektura
- 2.2. Podstawowe building blocks z Domain-Driven Design
- 2.3. CQS/CQRS

### 3. Implementacja

- 3.1. Przegląd aplikacji, strukturyzacja pików/katalogów/projektów w solucji.
- 3.2. Implementacja nowego modułu jako vertical slice

### 4. Komunikacja pomiędzy modułami

- 4.1. Sposoby komunikacji pomiędzy modułami
- 4.2. Kontrakty lokalne vs. kontrakty współdzielone
- 4.3. Implementacja mechanizmu komunikacji modułowej

### 5. Integracja pomiędzy modułami

- 5.1. Synchroniczna integracja (transakcyjność zapisu).
- 5.2. Asynchroniczna integracja między modułami. Omówienie wad oraz zalet (fire and forget vs async/await).
- 5.3. Implementacja mechanizmu integracji modułowej
- 5.4. Współdzielona infrastruktura i obsługa błędów

### 6. Rozproszone procesy biznesowe

- 6.1. Omówienie istniejących rozwiązań
- 6.2. Praktyczne zastosowanie wzorca Saga

## 7. Testowanie

7.1. Testy jednostkowe

7.2. Testy integracyjne

7.3. Testy end-to-end

7.4. Testy kontraktów

7.5. Testy wydajnościowe

## 8. Strategie wdrożeniowe

8.1. Omówienie mechanizmów udostępnionych przez framework

8.2. Dynamiczne wdrożenie modułów

## 9. Opcjonalny Dzień 4 - Przykład wydzielenia mikroserwisu z modułu

9.1. Wyzwania architektury mikroserwisowej

9.2. Zamiana komunikacji modułowej na komunikację HTTP

9.3. Zamiana integracji modułowej na komunikację z wykorzystaniem brokera wiadomości

9.4. Dostosowanie infrastruktury monolitu, do obsługi dwóch typów komunikacji tj. wewnątrz procesowej i zewnętrznej