

SZKOLENIE ŚREDNIO ZAAWANSOWANE

Micronaut framework

MICRONAUT

Czas trwania: 3 dni (24h)

Micronaut to nowoczesny, oparty o JVM framework, umożliwiający tworzenie modułowych, łatwo testowalnych rozwiązań, opartych o architekturę mikroserwisów (cloud-native)

Cele szkolenia

- Wprowadzenie i gruntowne poznanie możliwości frameworku
- Zrozumienie idei programowania z użyciem interfejsów, wstrzykiwania zależności oraz aspektów
- Poznanie najważniejszych modułów Micronaut framework

Zalety

- Warsztatowy charakter zajęć (realizacja jednego, dużego projektu)
- Gruntowne wprowadzenie do tematu, umożliwiające samodzielną pracę z omawianymi technologiami
- Program ukierunkowany na praktyczne zdobywanie wiedzy, umożliwiającej budowanie rozwiązań opartych o architekturę mikroserwisów
- 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 adresowane do programistów języka Java, chcących tworzyć skalowalne i łatwe w utrzymaniu aplikacje, z wykorzystaniem najnowszej wersji Micronaut framework

Wymagania

- Umiejętność programowania w języku Java na poziomie średnio zaawansowanym
- Elementarna wiedza z zakresu technologii wchodzących w skład JEE
- Podstawowa znajomość SQL i relacyjnych baz danych

Program

1. Wprowadzenie do Micronaut framework



- a. Założenia, możliwości, zastosowanie
 - b. Architektura mikroserwisów
 - c. Przygotowanie środowiska deweloperskiego
 - d. Tworzenie i konfigurowanie projektu
2. Praca z danymi
 - a. Konfiguracja dostępu do zewnętrznych źródeł danych
 - b. Integracja z JPA / Hibernate i zarządzanie transakcjami
 - c. Wykorzystanie baz SQL i NoSQL
 - d. Migracja danych oparta o rozwiązania Flyway i/lub Liquibase
 - e. Użycie pamięci cache
3. Usługi sieciowe
 - a. Projektowanie REST API
 - b. Budowanie usług opartych o standard JAX-RS
 - c. Publikowanie dokumentacji z OpenAPI/Swagger
 - d. Konsumowanie zewnętrznych usług opartych o REST oraz GraphQL
4. Usługi sterowane zdarzeniami
 - a. Programowanie reaktywne
 - b. Reactive Streams oraz biblioteki Reactor i RxJava
 - c. Integracja z popularnymi brokerami - Kafka, RabbitMQ
 - d. Strumieniowanie danych
5. Bezpieczeństwo i monitorowanie
 - a. Uwierzytelnianie i kontrola dostępu
 - b. Śledzenie i agregacja logów
 - c. Tracing
 - d. Publikowanie metryk z wykorzystaniem Micrometer
6. Wdrażanie aplikacji / Cloud Native
 - a. Czysta architektura i podział aplikacji na reużywalne moduły
 - b. Konteneryzacja
 - c. Rejestr usług / Service discovery
 - d. Wdrażanie w środowisku zarządzanym na przykładzie Kubernetes

