

SZKOLENIE ŚREDNIO ZAAWANSOWANE

Helidon framework

HELIDON

Czas trwania: 5 dni (40h)

Helidon to framework do tworzenia aplikacji Java w architekturze mikroservisowej, które działają na tradycyjnym JVM lub jako rozwiązania natywne

Cele szkolenia

- Wprowadzenie i gruntowne poznanie możliwości frameworka Helidon
- Zrozumienie idei programowania z użyciem interfejsów, wstrzykiwania zależności oraz aspektów
- Poznanie najważniejszych modułów Helidon 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 mikroservisów
- Praktyka przed teorią - wszystkie szkolenia technologiczne prowadzone są w formie warsztatowej. Konieczna teoria jest wyjaśniana na przykładzie praktycznych zadań
- Konkretna umiejętność - 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 Helidon framework adresowane jest do programistów języka Java, chcących tworzyć skalowalne i łatwe w utrzymaniu aplikacje z wykorzystaniem najnowszej wersji Helidon framework

Wymagania

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

Program

1. Wprowadzenie do Helidon
 - a. Założenia, możliwości, architektura, zastosowanie
 - b. Programowanie reaktywne
 - c. Eclipse MicroProfile
 - d. Architektura mikroservisów
 - e. Cloud Native Computing
 - f. GraalVM
 - g. Helidon SE vs Helidon MP
2. Wstrzykiwanie zależności w mikroservisach



- a. Wzorzec projektowy Dependency Injection
- b. Specyfikacja Context and Dependency Injection
- c. Wstrzykiwanie statycznej konfiguracji z użyciem Microprofile Config
3. Tworzenie usług sieciowych
 - a. Tworzenie usługi REST z użyciem JAX-RS oraz Jersey
 - b. Konsumowanie usługi REST
 - c. Konfiguracja CORS
 - d. Dokumentowanie REST API z użyciem OpenAPI
 - e. Reaktywne usługi sieciowe
4. Utrwalanie danych
 - a. Praca z bazami relacyjnymi z użyciem Java Persistence API
 - b. Walidacja danych z użyciem specyfikacji Bean Validation
 - c. Operacje transakcyjne
5. Obsługa zdarzeń i wiadomości
 - a. Reactive Messaging
 - b. Reactive Streams
6. Bezpieczeństwo
 - a. Uwierzytelnienie i autoryzacja z użyciem JSON Web Token
 - b. Zabezpieczanie zasobów JAX-RS
 - c. Zabezpieczanie wbudowanych końcówek Helidon
7. Monitorowanie usług
 - a. Status usługi za pośrednictwem Microprofile HealthCheck
 - b. Dane telemetryczne z użyciem Microprofile Metrics
 - c. Śledzenie ruchu z zastosowaniem Microprofile OpenTracing
8. Budowanie i wdrażanie aplikacji
 - a. Testy jednostkowe
 - b. Testy integracyjne
 - c. Budowanie aplikacji z użyciem Maven
 - d. Tworzenie natywnej wersji aplikacji z użyciem GraalVM
 - e. Konteneryzacja

