

Tytuł szkolenia: Programowanie reaktywne w Javie z Reactorem i Springem

Kod szkolenia: programowanie-reaktywne-Java

Wprowadzenie

Pierwsze dwa dni szkolenia zdominowane są przez zadania praktyczne (60%), niemniej jednak już na samym początku duży nacisk kładzie się na teorię. Dzień trzeci jest w całości warsztatowy – tu wykorzystujemy wiedzę zdobytą podczas poprzednich dni szkolenia.

Adresaci szkolenia

Szkolenie przeznaczone jest głównie dla dwóch kategorii uczestników:

- osoby, które mają doświadczenie w programowaniu w Javie i Springu i chcą w praktyczny sposób nauczyć się programowania reaktywnego z wykorzystaniem ww. narzędzi,
- osoby, które mają już praktyczne doświadczenie w programowaniu reaktywnym, ale czują potrzebę lepszego poznania teorii.

Wymagania wstępne:

Znajomość języka Java oraz frameworka Spring na poziomie średniozaawansowanym.

Uwaga:

Przedstawiony ramowy program szkolenia może zostać dopasowany do indywidualnych potrzeb Klienta.

Cel szkolenia

Czas i forma szkolenia

- 21 godzin (3 dni x 7 godzin), w tym wykłady i warsztaty praktyczne.

Plan szkolenia

1. Wprowadzenie do programowania reaktywnego
 - Definicja
 - Powody wprowadzenia podejścia reaktywnego
 - Marnotrawstwo zasobów
 - Problemy ze skalowalnością
 - Demo - porównanie wydajności aplikacji w sposób niereaktywny i reaktywny
 - Kiedy **nie** stosować podejścia reaktywnego
2. Strumienie reaktywne jako specyfikacja programowania reaktywnego
 - Komponenty (kod, testy, specyfikacja tekstowa)
 - Interfejsy
 - **Publisher**- nadawca elementów
 - **Subscriber**- odbiorca elementów
 - **Subscription**- połączenie między nadawcą a odbiorcą

- Przebieg subskrypcji (początek, elementy żądania, koniec)
- Ciśnienie wsteczne i nieograniczona subskrypcja
- Przegląd zaawansowanych reguł subskrypcji

3. Ramy reaktora

- Ogólny opis
- **Mono** oraz **Flux** jako klasy podstawowe
- Marmurowe diagramy
- **Flux**- strumień wielu elementów.
 - Tworzenie z istniejących elementów
 - Tworzenie programowe
- **Mono**- strumień pojedynczego elementu
- Subskrypcja strumieni
 - Poprzez wywołania zwrotne
 - przez **BaseSubscriber**.
- Anulowanie subskrypcji

4. Operatorzy

- Jak działają operatorzy
 - Podłączanie do łańcucha subskrypcji
 - W górę, w dół
- Mapowanie
 - synchroniczny (**map()**)
 - Asynchroniczne (**flatMap()**, **concatMap()**)
- Filtrowanie (**filter()**, **take()**, **skip()**)
- Zmniejszanie (**reduce()**)
- Łączenie strumieni
 - Tego samego typu (**merge()**, **concat()**)
 - Różne rodzaje (**zip()**)
- Zależności czasowe
 - Elementy opóźniające (**delayElements()**)
 - Wyzwalanie jednego strumienia przez inny (**then()**)
- Tworzenie własnych operatorów (**transform()**)
- Jak wybrać operatora, którego potrzebujesz – praktyczne wskazówki

5. Obsługa błędów

- Sygnalizacja błędu zamiast zgłaszania wyjątku
- Wymiana sygnału błędu
 - Używanie elementu
 - Korzystanie z sekwencji
- Ponowne mapowanie błędów
- Dodatkowe działanie w przypadku wystąpienia błędu
- Ponowna subskrypcja strumienia

6. Testowanie

- Przedstawiamy klasy pomocnicze testów Reactor
- **StepVerifier**- weryfikacja danych emitowanych przez strumień
 - kreacja
 - Łańcuch twierdzeń
 - Przeprowadzanie testu
 - Wirtualny minutnik
- **TestPublisher**- możliwość manipulowania emisją danych do strumienia
 - kreacja
 - Emitowanie danych

7. Debugowanie

- Analiza stosu wywołań reaktora
- Włączanie trybu debugowania
 - Jako globalny hak
 - Jako agent Java
- Punkty kontrolne
- Rejestrowanie zachowania strumienia przy użyciu `log()` operatora.

8. Wielowątkowość

- `Scheduler` jako abstrakcja wyboru nici
- Predefiniowane i niestandardowe harmonogramy
- Zmiana wątku publikacji elementu - `publishOn()` operator.
- Zmiana wątku tworzenia elementu - `subscribeOn()` operator.
- Równoległe przetwarzanie danych w strumieniu

9. Typowe pułapki programowania reaktywnego

- Brak abonamentu
- Zbyt chętnie operacje
- Operacje blokujące w wątkach nieblokujących
- Nadmiar operatorów
- Ręczne tworzenie strumieni

10. Framework Spring - podejście reaktywne

- Reaktywny stos sprężynowy - opis ogólny
- Reaktywny serwer HTTP - WebFlux
 - Podejście adnotacyjne
 - Podejście funkcjonalne
 - Technika SSE (Server-Sent-Events).
 - WebSockets
- Reaktywny klient HTTP - WebClient
 - Stworzenie i wykorzystanie
 - Połączenie kilku operacji

11. Warsztaty - implementacja gry MasterMind w podejściu reaktywnym z wykorzystaniem Springa

- Przedstawienie wymagań, ogólny zarys architektury i dostarczonych narzędzi
- Wykonanie części serwerowej
- Implementacja części klienckiej