

Program szkolenia:

Testy kontraktowe w .NET z Pact.NET

Informacje:

Nazwa:	Testy kontraktowe w .NET z Pact.NET
Kod:	NET-arch-ctnet
Kategoria:	Architektura .NET
Czas trwania:	3 dni
Forma:	30% wykłady / 70% warsztaty

Celem szkolenia jest nauka praktycznego zastosowania testów kontraktowych i narzędzia Pact Broker do zapewnienia maksymalnej pewności, zredukowania kosztów wdrożeń oraz drogich testów manualnych i end-to-end w środowisku mikroserwisowym.

Szkolenie porusza istotne aspekty, takie jak: kto powinien pisać testy kontraktowe, kto jest za nie odpowiedzialny, jak w praktyce wygląda Consumer-Driven Contract Testing i gdzie w strategii testowania umieścić testy kontraktowe, aby dostarczały maksimum wartości.

Na szkoleniu dowiesz się i przećwiczysz konkretne techniki wprowadzania łamiących zmian w API i schematach eventów

Zalety szkolenia:

- Zrozumiesz wzorce i antywzorce pisania testów kontraktowych
- Poznasz narzędzie Pact Broker, dzięki któremu odpowiedź na pytanie „czy mogę wdrożyć kod?” stanie się zautomatyzowanym zapytaniem
- Dowiesz się, gdzie uwzględnić testy kontraktowe w swojej strategii testowania
- Poznasz konkretne strategie wprowadzania łamiących zmian w API

Szczegółowy program:

1. Testy kontraktowe w testowaniu mikroserwisów

1.1. Czym jest architektura mikroserwisów

1.1.1. Odpowiedzialność serwisów i całego systemu

1.1.2. Poziomy testów a model C4

1.2. Złamanie kontraktu – zyski i koszty rozpraszania

1.3. Odpowiedzialność zespołów za serwisy i jakość / testy

1.4. Wzorce integracji i kontraktów

1.4.1. Komunikacja synchroniczna i asynchroniczna

1.4.2. Rodzaje relacji między producentem a konsumentem (Open Host, Conformist, Customer–Supplier)

1.4.3. Wzorce komunikatów: komendy, zdarzenia i kwerendy

2. Testy kontraktowe a podejście shift-left

2.1. Miejsce testów kontraktowych w piramidzie testowania

2.2. Szybka pętla zwrotna

2.3. Uzasadnienie i koszt stosowania testów kontraktowych

3. Narzędzia do testów kontraktowych w .NET

3.1. Pact.NET – testy kontraktowe dla REST i komunikatów z kolejek

3.2. Alternatywy i uzupełnienia: JSON Schema, stuby, mocki, Docker Compose

4. Wdrożenie testów kontraktowych w projekcie

4.1. Strategia pierwszego minimalnego testu kontraktowego

4.2. Weryfikacja kontraktu po stronie konsumenta i dostawcy

5. Publikowanie kontraktów i wyników weryfikacji

5.1. Pact Broker – publikacja kontraktów i rezultatów testów

5.2. Statusy kontraktów: „In Progress” i „Pending”

6. Rozbudowane testy kontraktowe

6.1. Testowanie komunikacji asynchronicznej (kolejki)

6.2. Test Fixtures i utrzymywalność testów

6.3. Provider States i matchery – zarządzanie stanem i parametrami

7. Włączenie testów kontraktowych do CI/CD

7.1. Automatyczna publikacja i weryfikacja kontraktów w pipeline

7.2. Raportowanie wdrożeń na środowiska

7.3. Testy kontraktowe jako część Code Review

8. Bezpieczeństwo wdrożeń – „Can I Deploy”

8.1. „Can I Merge” – weryfikacja PR-ów pod kątem kontraktów

8.2. „Can I Deploy” – zgodność kontraktów w danym środowisku

9. Wzorce i antywzorce w testowaniu kontraktów

9.1. Testowanie stanowych procesów

9.2. Dobór reprezentatywnych scenariuszy

9.3. Unikanie błędów w pipeline’ach innych zespołów

9.4. Zasada: „Nie obiecuj i nie oczekuj nic ponad to, co w kontrakcie”