

# *o książce*

---

Celem tej książki jest nauczenie, jak z powodzeniem tworzyć aplikacje za pomocą architektury mikroserwisowej.

Nie tylko omawia ona zalety architektury mikroserwisowej, ale także opisuje jej wady. Dowiemy się, kiedy powinniśmy rozważyć użycie architektury monolitycznej i kiedy warto skorzystać z mikroserwisów.

## **Kto powinien przeczytać tę książkę**

Ta książka koncentruje się na architekturze i rozwoju. Jest przeznaczona dla wszystkich osób odpowiedzialnych za tworzenie i dostarczanie oprogramowania, takich jak programiści, architekci, dyrektorzy (CTO) lub wiceprezesi (VP) ds. technicznych.

Książka skupia się na przedstawieniu wzorców architektury mikroserwisowej i innych koncepcji. Moim celem było sprawić, aby ten materiał stał się dostępny i przystępny niezależnie od używanego stosu technologicznego. Trzeba tylko znać podstawy architektury i projektowania aplikacji korporacyjnych. W szczególności należy rozumieć pojęcia, takie jak „architektura trójwarstwowa”, „projektowanie aplikacji internetowych”, „relacyjne bazy danych”, „komunikacja międzyprocesowa za pomocą przesyłania komunikatów i REST” oraz „podstawy bezpieczeństwa aplikacji”. Przykłady kodu używają Java i frameworka Spring. Aby uzyskać jak najwięcej z nich, warto zaznajomić się z frameworkiem Spring.

## **Mapa drogowa**

Ta książka składa się z 13 rozdziałów:

- Rozdział 1 opisuje objawy monolitycznego piekła. Występują one wówczas, gdy monolityczna aplikacja przerasta swoją architekturę. Doradza także, jak od tego uciec, przyjmując architekturę mikroserwisową. Zawiera również przegląd języka

wzorców architektury mikroserwisowej, będącego motywem przewodnim znacznej części książki.

- Rozdział 2 wyjaśnia, dlaczego architektura oprogramowania jest ważna, i opisuje wzorce, których można użyć do rozłożenia aplikacji na zbiór usług. Tłumaczy także, jak pokonać różne przeszkody, które zwykle napotykamy po drodze.
- Rozdział 3 opisuje różne wzorce niezawodnej komunikacji międzyprocesowej w architekturze mikroserwisowej. Objaśnia, dlaczego asynchroniczna komunikacja oparta na komunikatach jest często najlepszym wyborem.
- Rozdział 4 wyjaśnia, w jaki sposób zachować spójność danych w różnych usługach za pomocą wzorca sagi. Saga to sekwencja lokalnych transakcji koordynowanych za pomocą komunikatów asynchronicznych.
- Rozdział 5 pokazuje, jak zaprojektować logikę biznesową dla usługi z użyciem wzorca agregatów DDD (domain-driven design) oraz zdarzeń domenowych.
- Rozdział 6 opiera się na rozdziale 5 i wyjaśnia, w jaki sposób rozwijać logikę biznesową za pomocą skoncentrowanego na zdarzeniach wzorca event sourcingu w celu ustrukturyzowania logiki biznesowej i utrzymywania obiektów domenowych.
- Rozdział 7 omawia sposób implementowania zapytań, które pobierają dane rozproszone w wielu usługach, z użyciem wzorca kompozycji API lub Command Query Responsibility Segregation (CQRS).
- Rozdział 8 dotyczy wzorców zewnętrznych API do obsługi żądań ze zróżnicowanego zbioru klientów zewnętrznych, takich jak aplikacje mobilne, aplikacje JavaScript oparte na przeglądarce i aplikacje podmiotów zewnętrznych.
- Rozdział 9 to pierwszy z dwóch rozdziałów na temat automatycznych technik testowania mikroserwisów. Wprowadza ważne koncepcje testowania, takie jak piramida testów, która opisuje względne proporcje każdego rodzaju testu w zestawie testów. Pokazuje także, jak pisać testy jednostkowe, które stanowią podstawę piramidy testów.
- Rozdział 10 opiera się na rozdziale 9 i pokazuje, jak pisać inne typy testów w piramidzie testów, w tym testy integracyjne, testy kontraktowe i testy komponentów.
- Rozdział 11 obejmuje różne aspekty opracowywania usług gotowych do produkcji, w tym bezpieczeństwo, wzorzec konfiguracji zewnętrznej i wzorce obserwowalności usług. Wzorce obserwowalności usług obejmują agregację dzienników, metryki aplikacji i śledzenie rozproszone.
- Rozdział 12 opisuje różne wzorce wdrażania, których można użyć do wdrażania usług, w tym maszyny wirtualne, kontenery i wzorce bezserwerowe. Omówiono w nim także zalety korzystania z siatki usług, warstwy oprogramowania sieciowego, która pośredniczy w komunikacji w architekturze mikroserwisowej.
- Rozdział 13 wyjaśnia, jak przyrostowo refaktoryzować architekturę monolityczną do architektury mikroserwisowej, stosując wzorzec aplikacji dusiciela: wdrażanie nowych funkcjonalności jako usług oraz wyodrębnianie modułów z monolitu i konwertowanie ich na usługi.

W miarę przechodzenia przez te rozdziały poznamy różne aspekty architektury mikroserwisowej.

## O kodzie

Ta książka zawiera wiele przykładów kodu źródłowego, zarówno w ponumerowanych listingach, jak i wewnątrz zwykłego tekstu. W obu przypadkach kod źródłowy jest sformatowany w ten sposób czcionką o stałej szerokości, aby oddzielić go od zwykłego tekstu. Czasami kod jest również **pogrubiony**, żeby móc wskazać miejsca, w których się zmienił w porównaniu z poprzednimi krokami w danym rozdziale, na przykład gdy została dodana w nim nowa funkcjonalność. W wielu przypadkach ponownie sformatowaliśmy oryginalny kod źródłowy; dodaliśmy linie podziału i przerobiliśmy wcięcia, aby uwzględnić dostępne miejsca na stronie w książce. W rzadkich przypadkach nawet to nie było wystarczające, mimo że listingi zawierają znaczniki kontynuacji linii (►). Ponadto często w listingach usuwaliśmy komentarze z kodu źródłowego, jeśli został on opisany w tekście. Wielu listingom towarzyszą adnotacje dotyczące kodu, podkreślające ważne pojęcia.

Każdy rozdział, z wyjątkiem rozdziałów 1, 2 i 13, zawiera kod z przykładowej aplikacji. Kod tej aplikacji można znaleźć w repozytorium GitHub: <https://github.com/microservices-patterns/ftgo-application>.

## Forum książki

Zakup opracowania *Mikroserwis. Wzorce z przykładami w języku Java* umożliwia bezpłatny dostęp do prywatnego forum internetowego prowadzonego przez Manning Publications. Można na nim zamieszczać komentarze na temat książki, zadawać pytania techniczne i otrzymywać pomoc od autorów i innych użytkowników. Aby uzyskać dostęp do forum, należy przejść do strony <https://forums.manning.com/forums/microservices-patterns>. Więcej o forach Manninga i zasadach korzystania można dowiedzieć się także na <https://forums.manning.com/forums/about>.

Zobowiązanie Manninga względem swoich czytelników polega na zapewnieniu miejsca, w którym może odbywać się konstruktywny dialog między poszczególnymi czytelnikami oraz między czytelnikami a autorami. Nie jest to zobowiązanie się do konkretnej formy uczestnictwa ze strony autorów, których wkład na forum pozostaje dobrowolny (i niepłatny). Sugerujemy zadawanie autorom wielu wymagających pytań, aby dzięki temu mogli zgłębiać temat! Forum i archiwa poprzednich dyskusji będą dostępne na stronie internetowej wydawcy tak długo, jak książka będzie publikowana.

## Inne zasoby online

Innym doskonałym źródłem do nauki architektury mikroserwisowej jest moja strona <http://microservices.io>.

Zawiera ona nie tylko pełny język wzorców, ale także linki do innych zasobów, takich jak artykuły, prezentacje i przykładowy kod.