

Spring - wprowadzenie



Cześć

Maciej Koziara

Spring - czym jest?

- Zbiór narzędzi pozwalających tworzyć aplikacje webowe i nie tylko
- Biblioteka
- Framework
- Najpopularniejszy wybór wśród programistów (56% wg. ankiety JetBrains)

Spring - dlaczego powstał?

- Alternatywa dla ciężkiego i skomplikowanego JEE
- Open Source
- Szybki rozwój
- Świeże podejście do rozwiązywania problemów przysporzyło mu popularności

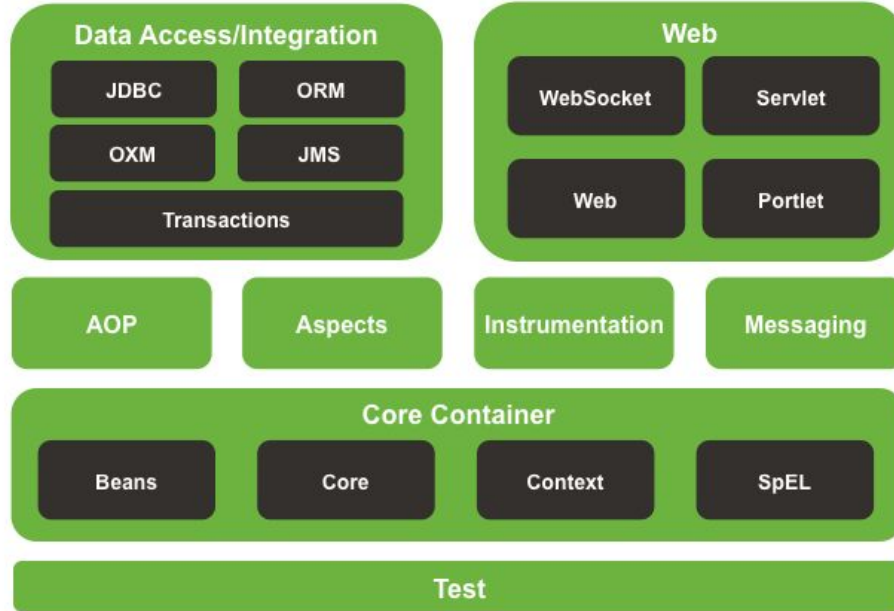
Spring - co go charakteryzuje?

- Spójny model programowania oparty na adnotacjach
- Zbudowany modułowo - korzystaj tylko z tego co potrzebujesz
- Wysoka jakość kodu i dokumentacji
- Flexible - schodzi programiście z drogi, gdy jest to potrzebne

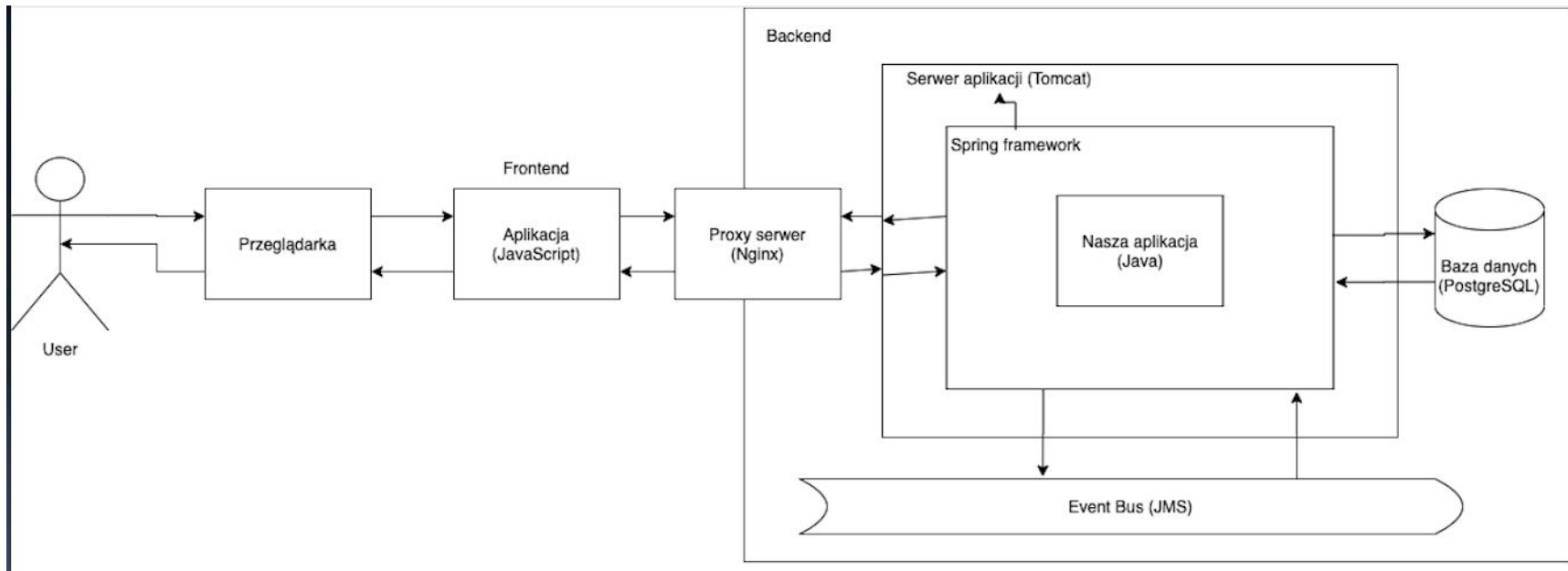
Spring - moduł



Spring Framework Runtime



Spring - gdzie jest jego miejsce?



Spring Boot - charakterystyka

- Każdy moduł wymaga skonfigurowania - żmudne, powtarzalne, narażone na błędy
- AutoConfiguration - domyślna, uznawana za najlepszą konfiguracja dla wszystkich modułów Springa
- Sterowane przez community
- Convention Over Configuration
- Nie odbiera wolności wyboru programiście

Spring Boot - magia



There is no magic. There is only
knowledge, more or less hidden.

Gene Wolfe

“ quoteancy

Spring Boot - AutoConfiguration

- Dostarczana przez tzw. startery
- Zawierają kod i pliki konfiguracyjne ustawiają domyślne wartości
- Pozwalają na dowolne ich nadpisywanie i dostosowywanie do naszych potrzeb
- Analiza zależności

URL - Uniform Resource Locator

`http://localhost:8080/category/1/movies?q=Noc&author=Maciek`

http - scheme

localhost - host, w tym przypadku oznacza, że jest dostępny jedynie lokalnie na naszej maszynie (IP: 127.0.0.1)

8080 - port

/category/1/movies - ścieżka (path), wartość 1 jest zmienna i określa się ją jako path param

q=Noc&author=Maciek - query string, zawierają dodatkowe zmienne

Zapytanie HTTP

- Verb - GET / POST / PUT / DELETE
- Request - żądanie do serwera
- Response - odpowiedź od serwera
- Response body - ciało odpowiedzi, może być interpretowane przez przeglądarkę

Spring Boot - wzorce projektowe

Dependency Injection - wzorzec projektowy do tworzenia obiektów, od których zależą inne obiekty.

Inversion of control - wzorzec projektowy wykorzystywany przez frameworki w celu uzyskania kontroli nad kodem. Framework zarządza tworzeniem obiektu oraz cyklem życia obiektu.

Decoupling - oddzielenie wysokopoziomowych elementów systemu od elementów niskopoziomowych poprzez wprowadzenie interfejsów. Pozwala to na zmianę niskopoziomowych części systemu bez zmian kodu wysokopoziomowego

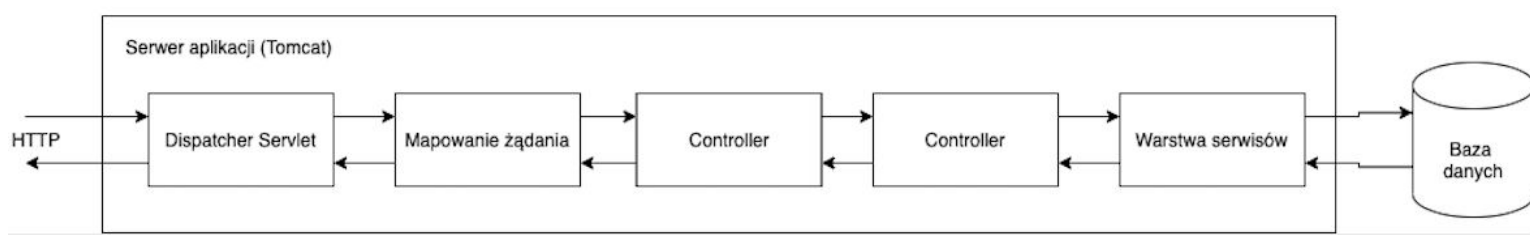
Spring Boot

Dependency Injection achieves **Decoupling** using **Inversion of Control**.

Spring Boot - DI process

1. Budowa grafu zależności
2. Utworzenie beanów bez zależności
3. Dostarczenie ich do kontekstu aplikacji
4. Utworzenie pozostałych beanów, pobranie wymaganych zależności z kontekstu

Spring - obsługa requesta



12 Factor



Metodologia tworzenia nowoczesnych,
skalowalnych i łatwych w utrzymaniu aplikacji.



Dzięki

You can find me at
[@username](#) & [user@mail.me](#)