

Spring Data

Przygotowali:
Jakub Zając
Damian Ciura

Czym jest Spring Data?

Spring Data to projekt w ramach frameworku Spring. Na który składają pomniejsze projekty. Jego celem jest ułatwienie programistom tworzenie warstwy komunikacji z bazami danych, serwisami chmurowymi.

Najważniejsze właściwości Spring Data:

- Repozytoria, niestandardowe abstrakcje mapowania obiektów
- Dynamiczne wprowadzanie zapytań z nazw metod repozytorium
- Klasy bazowe domeny implementacji zapewniające podstawowe właściwości
- Obsługa transparentnego audytu
- Możliwość integracji niestandardowego kodu repozytorium
- Łatwa integracja Spring za pośrednictwem JavaConfig i niestandardowych przestrzeni nazw XML
- Zaawansowana integracja z kontrolerami Spring MVC

Spring Data Repozytoria

Komponent pozwalający na wykorzystanie interfejsów dostarczanych przez Spring Data do tworzenia warstw dostępu do danych bez wykorzystania SQL, lub tworzenia funkcjonalności CRUD

Podstawowe interfejsy dostarczane przez Spring Data:

- CrudRepository - Podstawowy interfejs zawierający metody CRUD
- JpaRepository - Rozszerzenie CrudRepository, dodające sortowanie, paginację, itd.
- PagingAndSortingRepository - Umożliwia stronicowanie oraz sortowanie wyników

Spring Data Aggregate

- DDD (Domain-Driven Design) - Zbieranie jednego typu obiektu z bazy danych w ramach innego, i zarządzanie nim tylko poprzez obiekt zbierający
 - Na przykład `zamówienie` agreguje `produkty`, `produkty` zmieniane są tylko za pośrednictwem `zamówienia`
 - Aggregate Root - jest to obiekt zbierający inne. W powyższym przykładzie: `zamówienie`
- Zapytania - Wszystkie zapytania służące do zbierania w całość
 - Na przykład `sum`, `mean`, itp.
- JPA - Definiowanie zależności
 - `oneToOne`, `ManyToOne`, itp.

Spring Data Query

- Metody zapytań pochodnych - Spring automatycznie tworzy zapytanie bazując na nazwie metody
- Adnotacja `@Query(...)` - W ramach wnętrza `@Query` można utworzyć zapytanie w taki sam sposób jak bez Spring Data i przypisać je do metody
- Zapytanie przez przykład - Jako cel wyszukiwania używany jest obiekt z ustalonymi przez programistę polami, i zwracane są wszystkie obiekty dopasowane do przykładowego
- Specyfikacja - Programista przygotowuje metodę zapewniającą logikę dla warunków wyszukiwania obiektów
- Paginacja i sortowanie - Wbudowane metody pozwalające na dzielenie zbioru wyników oraz jego sortowanie

Spring Data Scroll

- Pozwala na pobieranie wyników w mniejszych częściach, nie ładując naraz całości do pamięci
- Przydatne w przypadku dużych zbiorów danych
- Implementacja zależna od źródła danych

Spring Data Projekcje

- Projekcje pozwalają na częściowe pobieranie obiektów ze źródła danych
- Rodzaje projekcji:
 - Interfejsowe - Programista tworzy interfejs z getterami wybranych pól
 - Klasowe (DTO - Data Transfer Object) - Klasa zawierająca tylko wybrane pola
 - Dynamiczne - Typ projekcji wstawiany jest jako parametr metody
- Istnieją dodatkowo dwa kolejne typy projekcji:
 - Zamknięte - Zwracane są tylko pola zawarte w projekcji
 - Otwarte - Pozwalają na zwracanie pól złożonych

Spring Data Konwersje

- Framework Spring Data automatycznie konwertuje typy Java na typy możliwe do odczytania przez konkretny system bazodanowy
- Możliwe jest tworzenie własnych konwerterów
- Przykłady zastosowania konwerterów:
 - Zmiana BSON na string
 - Mapowanie ENUM'ów
 - Przechowywanie typów rozbudowanych jako prostsze, np.: `BigDecimal` jako `String`

Spring Data Wywołania Zwrotne Encji

- Entity Callbacks - zestaw interfejsów pozwalających zaimplementować reakcje na odpowiednie eventy, bez wykorzystania adnotacji
- Przykłady Callback'ów:
 - BeforeConvertCallback<T> - Przed konwersją encji do formatu bazy danych
 - BeforeSaveCallback<T> - Przed zapisaniem encji
 - AfterSaveCallback<T> - Po zapisaniu encji
 - AfterLoadCallback<T> - Po załadowaniu encji
 - AfterConvertCallback<T> - Po konwersji wiersza/dokumentu bazy danych do encji
 - BeforeDeleteCallback<T> - Przed usunięciem encji
 - AfterDeleteCallback<T> - Po usunięciu encji

Spring Data JDBC

- Lekki framework służący do łączenia z bazą danych, implementujący JDBC (Java Database Connectivity)
- Brak ORM
- Brak zarządzania cyklem życia entity
- Brak Lazy Loading

Spring Data R2DBC

- R2DBC (Reactive Relational Database Connectivity) - Specyfikacja stworzona dla programowania reaktywnego
- Natywna integracja ze Spring WebFlux
- Wspiera tylko relacyjne bazy danych

Spring Data JPA

- Framework służący do łączenia z bazą danych, implementujący JPA (Java Persistence API)
- Wspiera ORM
- Automatycznie wypełnia pola takie jak: `createdAt`, `updatedAt`
- Wymaga dostawcy JPA, takiego jak Hibernate

Spring Data LDAP

- Framework służący do implementacji LDAP (Lightweight Directory Access Protocol)
- Zapewnia dostęp za pomocą metod podobnych do Spring Data JPA do zasobów LDAP
- Pozwala na mapowanie obiektów LDAP na obiekty JAVA

Spring Data REST

- Automatyzacja tworzenia podstawowych endpointów RESTful dla operacji CRUD
- Zapewnia support dla:
 - HATEOAS
 - paginacji
 - sortowania
 - filtrowania
 - eventów cyklu życia
- Pozwala na zarządzanie endpointami, formatami JSON



DEMO

Spring Data Redis

- Framework służący do implementacji Redis
- Pozwala na korzystanie z klasycznych API Redis oraz z uproszczonych wersji
- Pozwala na komunikację z Redis za pomocą repozytoriów Spring Data
- Udostępnia support dla budowania aplikacji reaktywnych z wykorzystaniem Redis

Wspierane Systemy Bazodanowe

Na dzień 01.05.2025 Spring Data oficjalnie posiada moduły wspierające współpracę z następującymi systemami bazodanowymi:

- MongoDB
- Apache
- Cassandra
- MySQL
- PostgreSQL
- Oracle
- MariaDB
- SQL Server
- H2 Database
- Cassandra
- Elasticsearch
- Couchbase
- Neo4j

Moduły społeczności

- Spring Data Aerospike
- Spring Data ArangoDB
- Spring Data Couchbase
- Spring Data Azure Cosmos DB
- Spring Data Cloud Datastore
- Spring Data Cloud Spanner
- Spring Data DynamoDB
- Spring Data YugabyteDB
- Spring Data Elasticsearch
- Spring Data Hazelcast
- Spring Data Jest
- Spring Data Neo4j
- Oracle NoSQL Database SDK for Spring Data
- Spring Data Reindexer
- Spring Data Vault



Q&A

Dziękujemy za uwagę