

Laboratorium 8 - SOA.

Tematyka: Hibernate - zastosowania zaawansowane

Celem laboratorium jest zaznajomienie z technologią pracy z relacyjnymi bazami danych przy użyciu biblioteki Hibernate. Laboratorium nr 8 jest uzupełnieniem zagadnień poruszanych w laboratorium nr 7. W laboratorium tym nauczyliście się Państwo współpracować z pojedynczą tabelą z bazy danych – wykorzystując wbudowane w używane przez państwa środowisko zintegrowane kreatory wygenerowaliście Państwa odpowiednie tabele na podstawie istniejącego obiektu lub w drugą stronę na podstawie tabeli stworzyli odpowiadający jej obiekt.

Dzisiejsze laboratorium obejmuje zagadnienia pracy z bardziej złożonymi strukturami – praca z kilkoma tabelami połączonymi różnymi relacjami - one-to-many lub many-to-many. Przećwiczymy również różne techniki operowania na danych dostarczane przez bibliotekę Hibernate.

NA początku przypomnienie najważniejszych informacji na temat Hibernate?

Hibernate jest najpopularniejszą biblioteką służącą do mapowania obiektowo-relacyjnego w Javie (**ORM / Object Relational Mapping**). Powstała w 2001 z inicjatywy Gavina Kinga, który w późniejszych latach w dużym stopniu przyczynił się do wprowadzenia istotnych zmian do specyfikacji EJB oraz JPA, doprowadzając je do stanu, w którym znamy je obecnie.

Hibernate jest rozwiązaniem wszystkich problemów związanych z operowaniem na bazie danych z perspektywy użytkownika obiektowego. Pozwala on automatycznie mapować obiekty Javy na wiersze w bazie danych oraz odczytywać rekordy z bazy danych i automatycznie tworzyć z nich obiekty. Na dobrą sprawę wykorzystując Hibernate teoretycznie nie musimy mieć większego pojęcia o poprawnym konstruowaniu zapytań w języku SQL, ponieważ będą one budowane za nas.

Obecnie Hibernate to coś dużo bardziej rozbudowanego niż tylko pośrednik pomiędzy Javą a bazą danych. Oprócz podstawowych możliwości ORM znajdziemy tutaj także osobny moduł odpowiedzialny za wyszukiwanie pełnotekstowe w oparciu o silnik Apache Lucene (**Hibernate Search**) oraz rozszerzoną implementację specyfikacji Bean Validation (**Hibernate Validator**).

HQL / JPQL

Hibernate udostępnia 4 podstawowe metody pozwalające wykonywać proste zapytania CRUD na obiektach encji. Oczywiście w realnym świecie takie podstawowe operacje niemal nigdy nie są wystarczające. Z pomocą przychodzi język specjalny język zapytań **HQL (Hibernate Query Language)** lub jego ustandaryzowana wersja **JPQL (Java Persistence Query Language)**. Są to języki mocno zbliżone do SQL, jednak nie operujemy w nich na tabelach, a zamiast tego posługujemy się notacją obiektową.

Szczegółowy opis języka można znaleźć pod tym linkiem:

<https://docs.jboss.org/hibernate/orm/3.3/reference/en/html/queryhql.html>

Criteria API

Criteria API jest kolejnym sposobem budowania bardziej zaawansowanych zapytań do bazy. Przypadnie on do gustu wszystkim osobom, które nie przepadają za językiem SQL i jemu podobnymi. Criteria API pozwala wykonać niemal dowolną operację na bazie danych wykorzystując jedynie notację obiektową, tzn. tworząc w programie odpowiednie obiekty i wywołując na nich odpowiednie metody.

Szczegółowy opis języka można znaleźć pod tym linkiem:

<https://docs.jboss.org/hibernate/orm/3.5/api/org/hibernate/Criteria.html>

Zadanie. Oparciu o dowolne rozwiązanie samodzielnie zbudować aplikację typu Biblioteka. Model danych powinien składać się z obiektów: Czytelnik, Książka, Autor oraz Wypożyczenia. Czytelnik składa się z imienia, nazwiska. Książka to tytuł, id autora. Wypożyczenia przechowuje informacje o wypożyczonych książkach przez czytelnika. Składa się z id książki, id czytelnika oraz daty wypożyczenia i daty zwrotu. Aplikacja umożliwia podgląd, dodawanie, usuwanie i modyfikację poszczególnych pozycji katalogu bibliotecznego.

Dodatkowo należy dorobić opcje pozwalające na wyszukiwanie bardziej zaawansowanych kryteriów wyszukiwania np.

1. podaj wszystkich czytelników, którzy pożyczili książki danego autora (np.Sienkiewicza) w okresie od 1.01.2018 do 1.05.2018
2. Kto przeczytał książkę „kapitan nemo” w podanym okresie czasu
3. Wszystkich autorów których książki pożyczył pan Jan Kowalski (ewentualnie w jakim okresie czasu)
4. Jakiego autora czyta się najwięcej? itp. .

Wykorzystaj wszystkie znane Ci metody operowania na danych (między innymi wymienione wyżej)

Jako interfejs użytkownika można użyć trybu tekstowego lub aplikacji Webowej (zalecane).

Jako pomoc możesz skorzystać z tutoriala znajdującego się pod tym adresem:

<http://www.mkyong.com/tutorials/hibernate-tutorials>