

PROGRAMOWANIE BAZ DANYCH

Na zajęciach będziemy pracować na repozytorium:

<https://github.com/rpieniazek/hibernate-training>

KONFIGURACJA

Aby móc uruchomić projekt konieczna jest baza danych MySQL. Projekt zakłada taką konfigurację:

port: 3307,
user:root,
password:root

Jeżeli Twoje ustawienia są inne, można je dostosować w application.properties.(Przy każdym przełączeniu brancha).

Dodatkowo potrzebna jest schema LIBRARY:
CREATE SCHEMA LIBRARY

JDBC

Przełącz się na branch **jdbc**. Otwórz klasę JDBCExample.

W Bazie danych stwórz tabelę book:

```
create table LIBRARY.book
(
  id bigint auto_increment primary key,
  title varchar(50) not null,
  book_year int null
);
```

- W jaki sposób Statement może być podatny na SQL Injection?
- W jaki sposób ResultSet może nas od tego uchronić?
- (Poza tematem) Czym jest konstrukcja przy try w main?

MAPOWANIE ENCJI

Branch **step1**

- Tworzenie klas encji
- Co się może zdarzyć, gdy klasa nie jest spójna z tabelą w bazie?
- Spróbuj dodawać, usuwać i modyfikować pola, w różnych konfiguracjach.

DAO, SERVICE

Branch **step2**

- Model warstwowy

MAPOWANIA

Na branchu **step3-mappings-before** w pakiecie model są klasy, między którymi trzeba dorobić relacje zgodnie ze schematem z folderu docs. W razie wątpliwości na branchu **step3-mappings-after** znajdują się gotowe rozwiązania.

TWORZENIE ZAPYTAŃ

Na branchu **step4-criteria-before** w klasie BookRepository znajdują się zadania do zaimplementowania. W razie wątpliwości rozwiązania są na **step4-criteria-after**.

JPA REPOSITORY

Branch **step5-jpa-before**

Potestować w jaki sposób zapisywane są encje połączone różnymi rodzajami relacji. Czy kolejność zapisywania ma znaczenie? Kiedy trzeba zapisywać encje osobno, a kiedy robi to za nas hibernate?

- znaleźć książki, z biblioteki o podanej nazwie
- wypisać ilość książek, których tytuł zawiera podany fragment
- wypisać wszystkie książki, które nie mają review

TESTY

Branch step6-tests-before

Testy powinno się wykonywać na innej bazie danych niż aplikacyjna. Dlatego konieczne jest stworzenie bazy testowej:

```
CREATE SCHEMA LIBRARY_TEST
```

W pliku test.properties znajdują się 'namiary' właśnie na taką bazę. Prawdopodobnie tutaj również będziecie musieli pozmienić usera, hasło albo port.

JOOQ

Brancgh step7-jooq

Jooq to pewna alternatywą dla Hibernate, ale jest bardziej ubogi o funkcjonalności. Przekłada się to korzystnie na jego wydajność.

Można myśleć o JooQu jak o rozbudowanym JDBC.

JooQ generuje klasy reprezentujące tabele i kolumny na podstawie schematu bazy danych. Do generowania tych klas służy plugin mavenowy, który dodaliśmy do pom.xml. Zwróć uwagę, że znajdują się tam namiary na bazę (url, logi i hasło), a najprawdopodobniej są inne niż te w repozytorium. Zwróć również uwagę na properties inputSchema (linia 82). Niektórzy mieli nazwę schemy małymi literami (library, a nie LIBRARY), więc to również musicie dopasować.

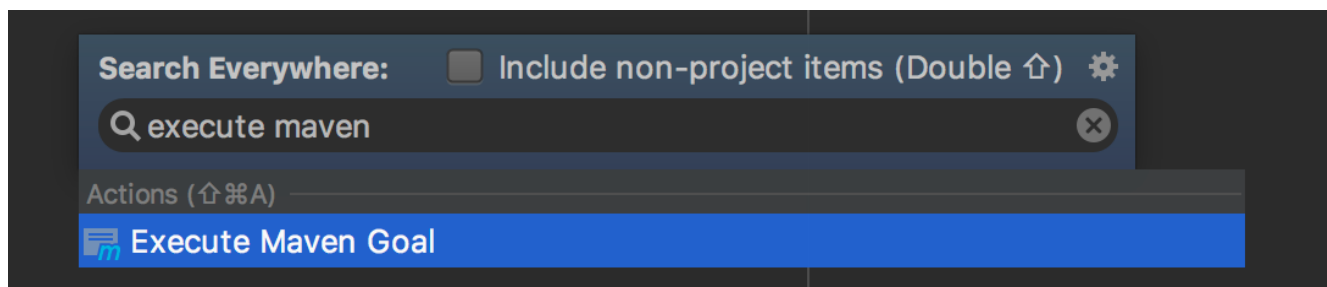
Aby odpalić ten plugin należy w katalogu z projektem wykonać polecenie :

```
mvn clean install
```

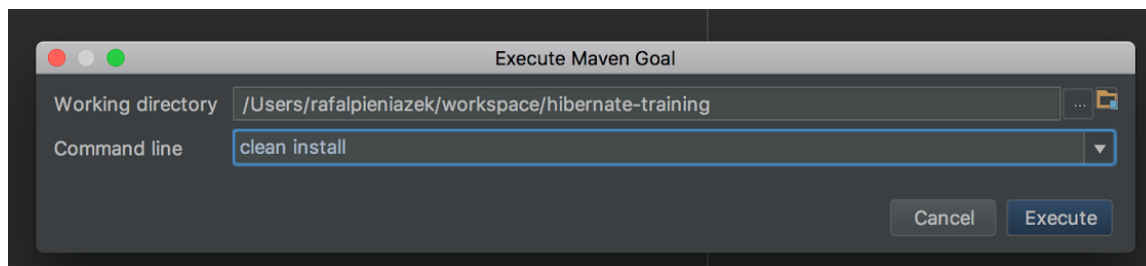
Maven jest zainstalowany na komputerach, które dostaliście, ale jeżeli korzystasz z prywatnego to możliwe, że go nie masz. Można posłużyć się mavenem z IntelliJ:

Naciskasz dwa razy shift i wyszukujecie:

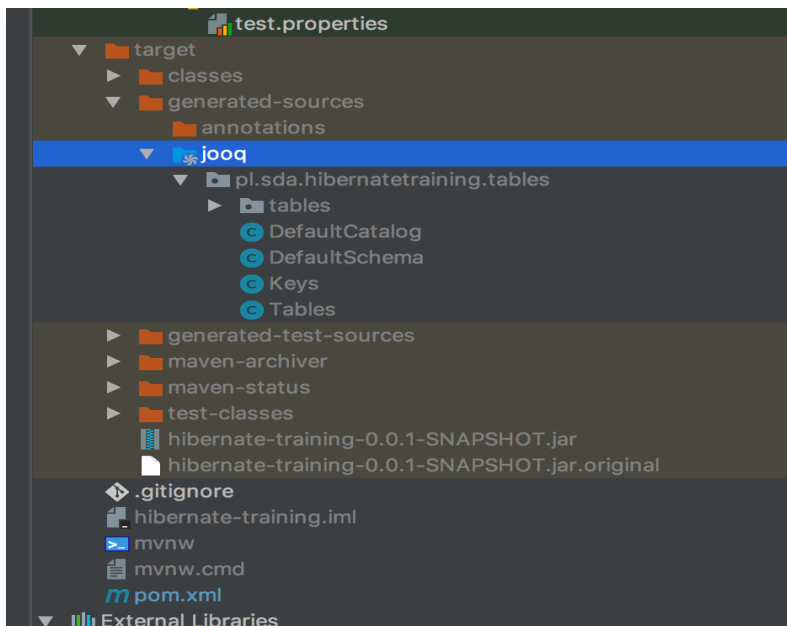
execute maven goal



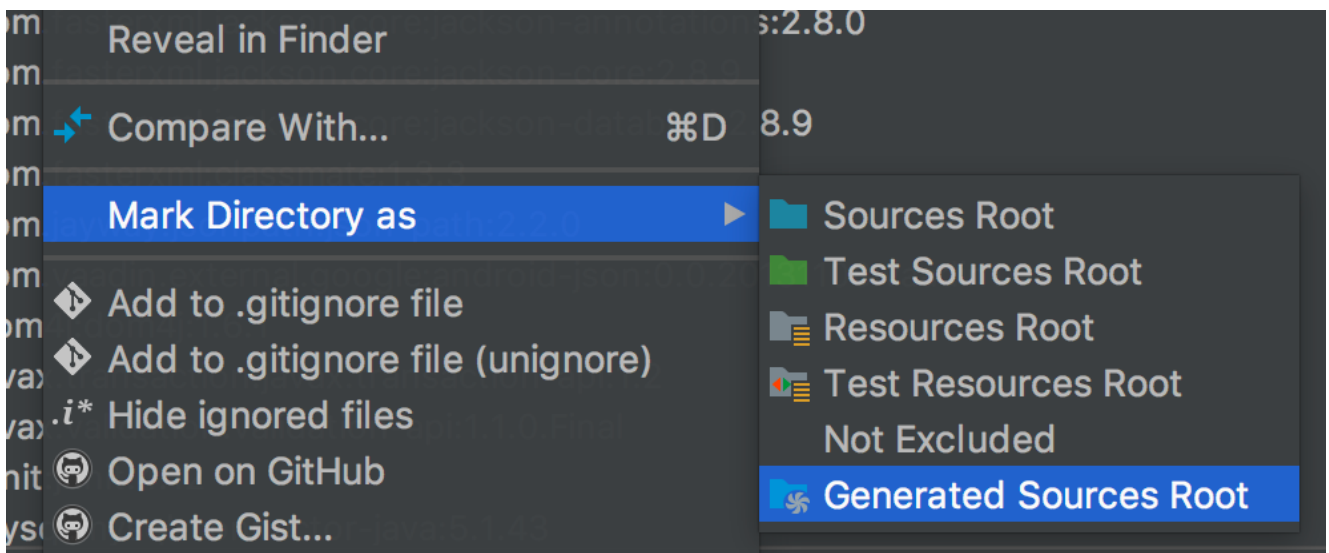
A następnie : **clean install**



Potem, jeżeli build przejdzie poprawnie (patrzecie dokładnie w logi!) pojawi się katalog target/generated-sources/jooq:



Czasem IntelliJ nie załapuje i trzeba ręcznie oznaczyć katalog jooq jako generated sources. W tym celu klikamy prawym na katalog i z menu wybieramy Mark as: **generated sources root**:



Od tej pory możemy odwoływać się w Dao Jooqowych do tych klas. Zobaczcie jak to robiliśmy w klasie BookJooqDao, np BOOK pisany wielkimi literami odnosi się właśnie do tych klas wygenerowanych.

DZIEDZICZENIE W HIBERNATE

W jaki sposób w relacyjnej bazie można przedstawić hierarchię dziedziczenie i jak to skonfigurować w hibernate? Generalnie są trzy sposoby:

- **1. Jedna tabela:** Całą hierarchię trzymamy w jednej tabeli. Zeby rozroznic która encje potomne, dodaje się specjalną kolumnę (dyskryminator). Pola które nie należą do danej encji są poprostu nullami.
- **2.Dwie tabele:** Tworzymy po jednej tabeli dla każdej encji pochodnej. W każdej tabeli znajdują się zduplikowane kolumny z klas nadrzędnych.
- **3. Trzy tabele:** Tworzymy tabele dla nadklasy i po jednej tabeli dla kazdej klasy pochodnej. Dodatkowo w klasach pochodnych dodajemy klucz obcy do tabeli z encjami nadklasy.