

Programowanie komponentowe – ćwiczenie 5

Cel ćwiczenia:

DAO

1. Wykorzystujemy projekt z ćw4 oraz wybraną bazę danych z tabelami dla klas: Car, Engine, Oil, Transmission.
2. Dodajemy starter: spring-boot-starter-jdbc
3. Dodajemy odpowiedni sterownik wybranej bazy danych
4. W pliku application.properties konfigurujemy źródło danych (można to zrobić również w klasie konfiguracyjnej definiując odpowiedni bean), np.

```
spring.datasource.url=jdbc:postgresql://<host>:<port>/<database>
spring.datasource.username=<username>
spring.datasource.password=<password> spring.datasource.driver-class-
name=org.postgresql.Driver
```

5. W klasie konfiguracyjnej lub głównej włączamy managera transakcji za pomocą adnotacji @EnableTransactionManagement (domyślnie nbean nazywa się transactionManager)
Uwaga! Transakcje definiuje się za pomocą adnotacji @Transactional umieszczonej nad metodą. Brak parametru transactionManager spowoduje użycie domyślnego managera transakcji.

5. W pakiecie głównym tworzymy pakiet .dao
6. W pakiecie .dao dodajemy interface CarDao i klasę, która go implementuje CarDaoImpl.
7. W interface CarDao deklarujemy metody:

- List<Car> findAll()
- Car findCarByName(String carName) (można wykorzystać inne pole)
- void insert(Car car)
- void update(Car car)
- void delete(Car car)

8. W klasie CarDaoImpl zadeklarować jdbcTemplate oraz zaimplementować wszystkie metody interface'u. Przykład deklaracji JdbcTemplate:

```
private JdbcTemplate jdbcTemplate;
@Autowired
public void setDataSource(DataSource dataSource) {
    this.jdbcTemplate = new JdbcTemplate(dataSource);
}
```

Uwaga! Wykonując operację na samochodzie wykonujemy ją również na tabelach podrzędnych.

9. W klasie AppRunner wywołać metody CarDao dla przykładowych danych.