



Lesson 35

18.11.2024



+

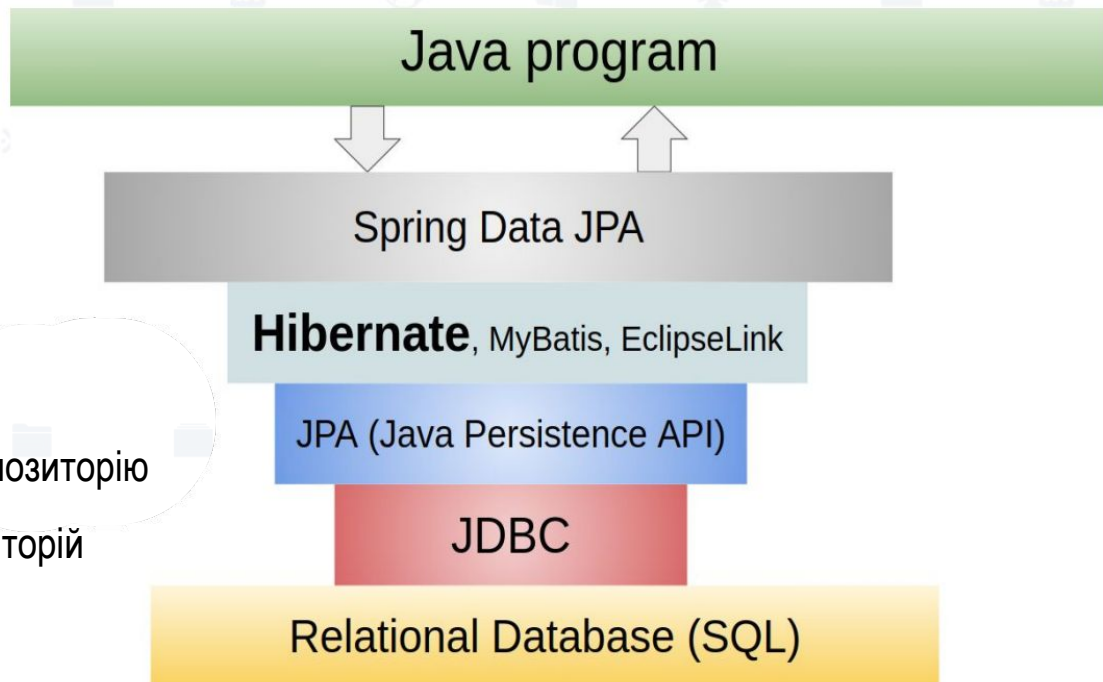


Spring Data JPA

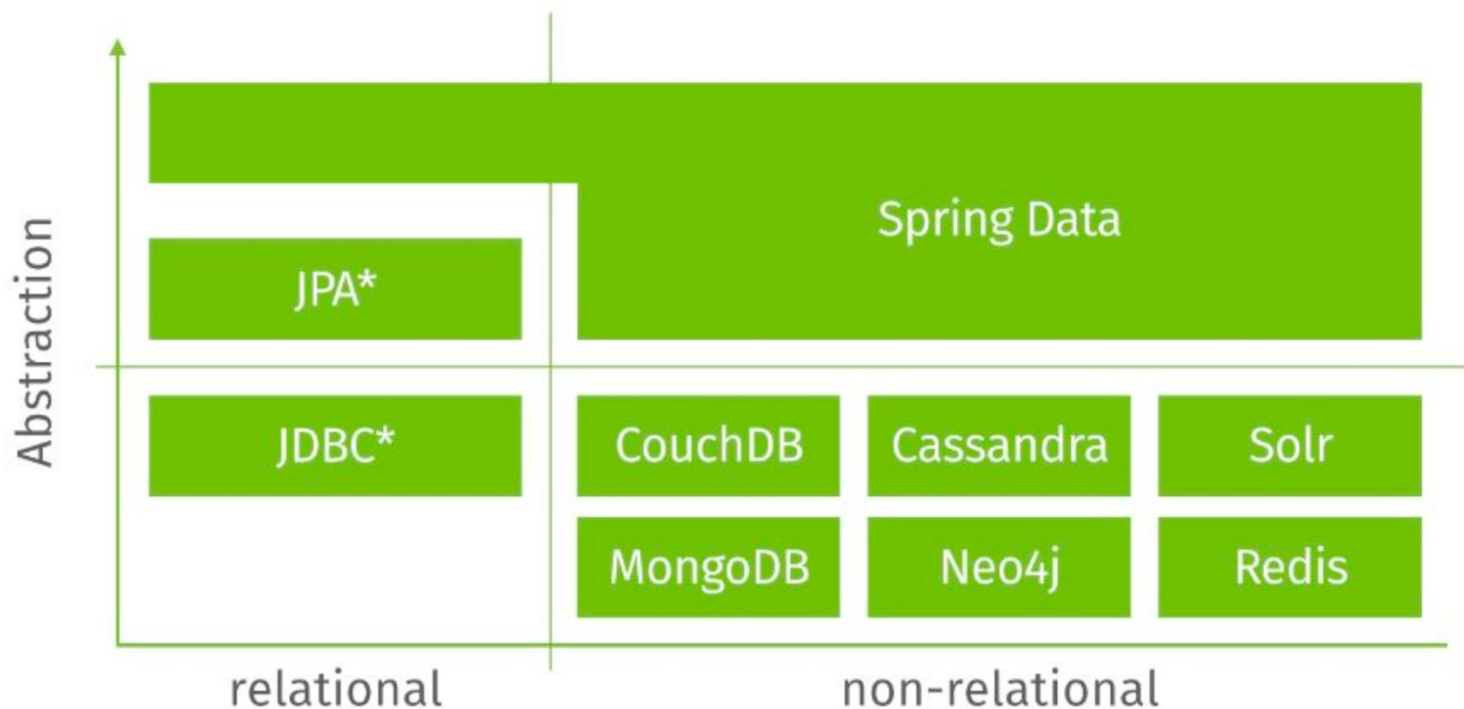
Spring Data - додатковий зручний механізм для взаємодії з сутностями бази даних, організації їх в репозиторії, вилучення даних, зміна, в якихось випадках для цього буде достатньо оголосити інтерфейс і метод у ньому, без імплементації.

Містить:

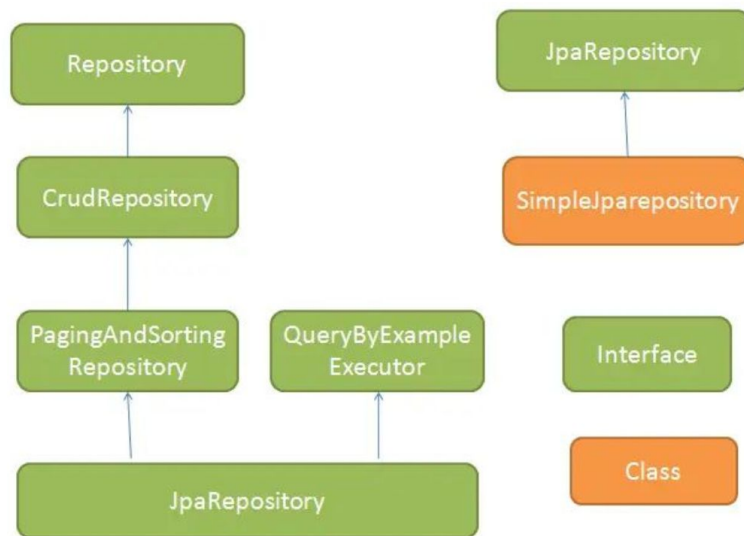
1. Spring Repository
2. Методи запитів із імені методу
3. Конфігурація та налаштування
4. Спеціальна обробка параметрів
5. Користувальницькі реалізації для репозиторію
6. Користувальницький Базовий Репозиторій
7. Методи запитів - Query




Spring Data



Репозиторії Spring Data JPA – це інтерфейси, які ви можете визначити для доступу до даних. Запити JPA створюються автоматично з імен ваших методів. Наприклад, інтерфейс CityRepository може оголосити метод `findAllByState(String state)`, щоб знайти всі міста (city) у цьому штаті (state). Для більш складних запитів ви можете анотувати ваш метод анотацією Query в Spring Data. Сховища Spring Data зазвичай беруться з інтерфейсів Repository чи CrudRepository. Якщо ви використовуєте автоконфігурацію, пошук у репозиторіях відбувається з пакета, що містить ваш основний клас конфігурації (той, який анотований `@EnableAutoConfiguration` або `@SpringBootApplication`).





Методи запитів з імені методу

Запити до сутності можна будувати з імені методу. Для цього використовується механізм префіксів **find...By**, **read...By**, **query...By**, **count...By**, та **get...By**, далі від префікса методу починає розбір решти. Вступний вираз може містити додаткові вирази, наприклад, **Distinct**. Далі перший **By** діє як розділювач, щоб вказати на початок фактичних критеріїв. Можна визначити умови для властивостей сутностей та об'єднати їх за допомогою **And** та **Or**.



Keyword	Sample	JPQL snippet
And	<code>findByLastnameAndFirstname</code>	<code>... where x.lastname = ?1 and x.firstname = ?2</code>
Or	<code>findByLastnameOrFirstname</code>	<code>... where x.lastname = ?1 or x.firstname = ?2</code>
Is, Equals	<code>findByFirstname, findByFirstnameIs, findByFirstnameEquals</code>	<code>... where x.firstname = ?1</code>
Between	<code>findByStartDateBetween</code>	<code>... where x.startDate between ?1 and ?2</code>
LessThan	<code>findByAgeLessThan</code>	<code>... where x.age < ?1</code>
LessThanEqual	<code>findByAgeLessThanEqual</code>	<code>... where x.age <= ?1</code>
GreaterThan	<code>findByAgeGreaterThan</code>	<code>... where x.age > ?1</code>
GreaterThanEqual	<code>findByAgeGreaterThanEqual</code>	<code>... where x.age >= ?1</code>
After	<code>findByStartDateAfter</code>	<code>... where x.startDate > ?1</code>
Before	<code>findByStartDateBefore</code>	<code>... where x.startDate < ?1</code>



NotLike	findByFirstnameNotLike	... where x.firstname not like ?1
StartingWith	findByFirstnameStartingWith	... where x.firstname like ?1 (parameter bound with appended %)
EndingWith	findByFirstnameEndingWith	... where x.firstname like ?1 (parameter bound with prepended %)
Containing	findByFirstnameContaining	... where x.firstname like ?1 (parameter bound wrapped in %)
OrderBy	findByAgeOrderByLastnameDesc	... where x.age = ?1 order by x.lastname desc
Not	findByLastnameNot	... where x.lastname <> ?1
In	findByAgeIn(Collection<Age> ages)	... where x.age in ?1
NotIn	findByAgeNotIn(Collection<Age> ages)	... where x.age not in ?1
True	findByActiveTrue()	... where x.active = true
False	findByActiveFalse()	... where x.active = false
IgnoreCase	findByFirstnameIgnoreCase	... where UPPER(x.firstname) = UPPER(?1)

Спеціальна обробка параметрів

У методах запитів, їх параметрах можна використовувати спеціальні параметри Pageable, Sort, а також обмеження **Top** та **First**. Наприклад ось так можна взяти другу сторінку (індекс з 0), розміром у три елементи та сортуванням по firstName, попередньо вказавши в методі репозиторію параметр Pageable, також буде використано критерії з імені методу - «Шукати по FirstName починаючи з %

@Repository

```
public interface CustomizedEmployeesCrudRepository extends CrudRepository<Employees, Long> {  
    List<Employees> findByFirstNameStartsWith(String firstNameStartsWith, Pageable page);  
}  
  
public void testFindByFirstNameStartsWithOrderByFirstNamePage() {  
    List<Employees> list = employeesCrudRepository  
        .findByFirstNameStartsWith("A", PageRequest.of(1,3, Sort.by("firstName")));  
    list.forEach(e -> System.out.println(e.getFirstName() + " " + e.getLastName()));  
}
```

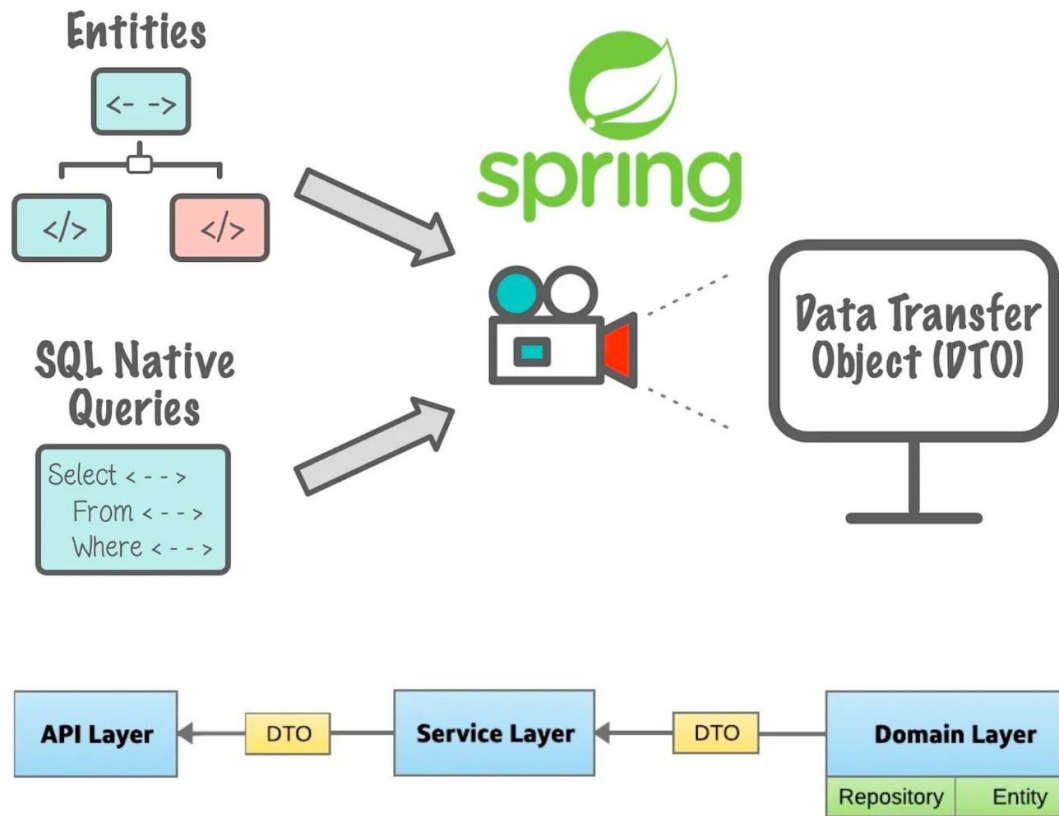


Методи запитів — Query

Якщо потрібний специфічний метод або його реалізація, яку не можна описати через ім'я методу, то це можна зробити через деякий Customized інтерфейс (CustomizedEmployees) та зробити реалізацію обчислення. А можна піти іншим шляхом, вказуючи запит (HQL або SQL), як обчислити цю функцію. Для мого прикладу з `getEmployeesMaxSalary` цей варіант реалізації навіть простіше. Я ще ускладню його вхідним параметром `salary`. Тобто. достатньо оголосити в інтерфейсі метод та запит обчислення

```
@Repository
public interface CustomizedEmployeesCrudRepository extends CrudRepository<Employees, Long>,
    CustomizedEmployees<Employees> {

    @Query("select e from Employees e where e.salary > :salary")
    List<Employees> findEmployeesWithMoreThanSalary(@Param("salary") Long salary, Sort sort)
}
```



Spring data architecture

