

Spring JPA та MVC

Spring JPA – зручний доступ до БД. Може працювати з Hibernate і не тільки.

Spring MVC – зручне написання web-додатків. Використовується як для HTML-сторінок (jsp, сторонні шаблонізатори), так і для REST-сервісів.

Загальна задача

Необхідно розробити веб-систему для автоматизації роботи ветеринарної клініки.

Предметна область: **клієнти** приходять у ветеринарну клініку, і залишають **замовлення**. Після виконання замовлення воно отримує статус виконано.

План роботи

- Проектування БД
- Написання JPA коду, тестування в консолі
- Написання веб-сервісу для виводу результатів у вигляді HTML
- Написання REST-сервісу

Spring JPA

- Створити сутності для Клієнт і Замовлення. Зв'язок один до багатьох (один клієнт може зробити декілька замовлень).
- Клієнт має **ім'я** і **телефон**
- Замовлення має **опис**, **дату**, і **статус** (в роботі, виконано)

Spring JPA

- Створити ClientRepository для вибірки всіх клієнтів з усіма їх замовленнями. ClientRepository повинний наслідувати JpaRepository. Створити метод для пошуку клієнта по імені.
- Створити OrderRepository. Створити в ньому метод для вибірки лише тих замовлень, де статус **в роботі**.

Spring JPA – Service Layer

- Створити ClientService
- Створити OrderService
- Протестувати роботу класів в консолі
(System.out.println)

Spring MVC

- Ініціалізація через XML
- Ініціалізація (і подальша робота) через аннотації
- Перший контролер, що виводить Hello Spring MVC

Spring MVC - JSP

- Написати контролер для виводу всіх клієнтів
- Написати контролер для виводу лише тих замовлень, що зараз в роботі
- Написати контролер для виводу лише тих клієнтів, в яких немає активних замовлень

Додатково

- Перспективи REST
- Для чого потрібний Service-layer
- Spring boot

Spring MVC - REST

- Написати аналогічні запити, як для JSP, але виводити в JSON форматі
- Написати REST метод для додавання нового клієнта. Дані передаються методом POST, метод повертає JSON-відповідь наступного вигляду:

```
{  
  status: "ok"  
}
```