

Дополнительные советы ментора

Реализация пагинации

Для реализации пагинации используйте возможности, предоставляемые `JpaRepository`. Вам нужно определить в интерфейсе репозитория метод поиска, аналогичный тому, который вы использовали ранее, но принимающий в качестве параметра также объект `Pageable`. Например, для поиска вещи ранее использовался метод `List<Item> findByOwnerId`, создайте метод `Page<Item> findByOwnerId(Long ownerId, Pageable pageable)`. Тогда всё остальное для реализации пагинации на уровне базы данных для вас сделает Spring.

Вам нужно будет только изменить вызов к данному методу, передавая в качестве дополнительного параметра описание требуемой страницы. Для этого используйте метод `PageRequest.of(page, size, sort)`. Обратите внимание, что вам нужно будет преобразовать параметры, передаваемые пользователем, — `start` и `size` — к параметрам, требуемым Spring, — `page` и тот же `size`.

Тесты, работающие с базой данных и `@DataJpaTest`

Для тестов, использующих аннотацию `@DataJpaTest`, нужно учесть некоторые особенности. Spring автоматически создаёт Entity Manager для таких тестов, но не с классом `EntityManager`, а с особым классом `TestEntityManager`. Он представляет собой обёртку над `EntityManager`, заточенную под работу в тестах.

Поэтому важно правильно указать тип, когда вы инжектите бин с `EntityManager` в ваш тестовый класс. Кроме того, `TestEntityManager` не предоставляет некоторые методы, доступные в `EntityManager`. Например, не позволяет напрямую выполнять JPQL-запросы. Поэтому, чтобы выполнить такой запрос, вам нужно будет сначала получить экземпляр обычного `EntityManager`. Сравните:

```
private final EntityManager em;
...
TypedQuery<User> query = em.
    .createQuery("Select u from User u where u.email = :email", User.class);
User user = query.setParameter("email", userDto.getEmail()).getSingleResult();
```

И

```
private final TestEntityManager em;
...
TypedQuery<User> query = em.getEntityManager()
    .createQuery("Select u from User u where u.email = :email", User.class);
User user = query.setParameter("email", userDto.getEmail()).getSingleResult();
```