

Кейс №4

Легенда

В нашем приложении Petclinic запускается новая киллер фича – ведение родословных для любых животных. Она ~~якобы~~ умеет собирать и анализировать родословные из разных источников и даже выяснять настоящие имена питомцев, в том числе их титулы, например, “Тузик младший” или “Мухтар Второй” (и сохраняет их в БД на радость владельцам). На случай невозможности автоматического обновления имени, фича предусматривает ручное обновление через уже существующий API-метод `PUT /api/pets/{petId}`.

Сейчас фича проходит ограниченное бета-тестирование на контрольной группе из одного хомяка по имени Basil.

Техническая реализация

Почти вся фича реализована в одном сервисном бине `PedigreeService`. Сервис активируется свойством `enable-pedigree` через командную строку или настройки приложения.

Все тексты якобы загруженных родословных хранятся в кэше

`PedigreeService#pedigreeTextCache`, ключами которого являются ID питомцев в БД клиники, а значениями – сами тексты в формате обычных строк. Кэш реализован как синхронизированная обёртка над `HashMap`; с учётом малой конкуренции за доступ этого кажется вполне достаточно.

Разработка и тестирование производились на PostgreSQL, поэтому рекомендуется именно эта СУБД. Для работы с именами животных (запросы и обновления) в сервисе используется прямое обращение к JDBC и `DataSource`, чтобы гарантировать отсутствие лишних сложностей и побочных эффектов, вносимых Spring’ом. Также во избежание фантомных чтений в приложении используется уровень изоляции транзакций `SERIALIZABLE`.

Автоматическое обновление родословных (сводящееся пока только к смене имени в БД) выполняется `@Scheduled`-методом `updatePedigreeBySchedule()` раз в 5 секунд. Ручное обновление выполняется методом `updatePedigreeByRequest()`, вызываемым из контроллера метода `PetRestController.updatePet()`, если обновляемый питомец входит в бета-тестирование (участник указан в константе `PedigreeService.BETA_TEST_PET`).

При любом виде обновления сервис обеспечивает консистентное изменение данных как в БД, так и в кэше текстов родословных за счёт синхронизации доступа к кэшу и управления транзакциями.

Проблема

Периодически (непонятно при каких условиях) вызов API метода `PUT /api/pets/{petId}` для питомца-участника бета-тестирования зависает, и после этого перестаёт работать автоматическое обновление родословных.

Попутное наблюдение: если метод хоть раз завис, приложение не может завершиться штатно, его приходится “убивать” (останавливать принудительно).

👣 Шаги воспроизведения

1. Запустить приложение с ключом `--enable-pedigree`.
2. Открыть вызов метода `PUT /api/pets/{petId}` через [Swagger UI](#).
3. Указать `petId=2`
4. Вставить следующий JSON в поле для ввода тела PUT-запроса:

```
{
  "name": "Basil The Second",
  "birthDate": "2002-08-06",
  "type": {
    "name": "hamster",
    "id": 6
  },
  "id": 2,
  "ownerId": 2,
  "visits": []
}
```

5. Выполнить запрос кнопкой **Execute**.

Ожидаемый результат: метод успешно завершается со статусом `HTTP 418 (I am a teapot)`

Фактический результат: метод не завершается, ответа нет, в Swagger UI крутится индикатор загрузки

Примечание. Ошибка воспроизводится не всегда, может потребоваться несколько попыток.