

Кейс №2

Легенда

Мы добавляем в PetClinic искусственный интеллект, который ~~якобы~~ умеет оценивать риски заболевания животных на основе множества внешних факторов, таких как общая зоо-эпидемиологическая обстановка в городе, характер жалоб при недавних обращениях в клинику, статистика заболеваемости по видам животных и т.д. Пересчёт рисков осуществляется по расписанию.

Рассчитанные риски ~~как будто бы~~ сохраняются в БД и служат основой для различных фич нашего приложения. Одной из них является предоставление рекомендаций о визитах животного в клинику: когда примерно приехать и для чего (например, для профилактической прививки).

Технические реализация

Почти вся фича реализована классом (сервисом) `DiseaseRiskAiService`. Сервис активируется свойством `enable-risk-ai` через командную строку или настройки приложения. За периодический пересчёт рисков отвечает `@Scheduled`-метод `recalculateDiseaseRisks`, запускаемый раз в 5 секунд.

Выдача рекомендаций осуществляется новым API методом `GET /api/pets/{petId}/recommendedVisits`, который сводится к вызову метода `DiseaseRiskAiService#fetchRecommendedVisits`.

Рекомендации рассчитываются лениво (только по запросу пользователя для конкретного животного) и помещаются в кэш `visitCache` (`ConcurrentHashMap`), потому что:

- их расчёт ~~якобы~~ дóрог по вычислениям;
- животных в клинике довольно много;
- но при этом ещё неизвестно, будет ли востребована эта фича.

Поскольку пересчёт рисков занимает существенное время, на этот период взводится блокировка на запись (`DiseaseRiskAiService#risksLock`), а метод составления рекомендуемых визитов использует эту же блокировку на чтение. Задумка здесь в том, что сколько бы потоков одновременно не пытались строить рекомендации, они не будут мешать друг другу, потому что запрашивают только *чтение* рисков. Дело ещё более смягчается тем, что кэш своим наличием ещё более прореживает число обращений к рискам. Фактическая блокировка может понадобиться только в относительно редком случае действительного пересчёта рисков.

Проблема

Метод API для получения рекомендуемых визитов `GET /api/pets/{petId}/recommendedVisits` периодически (при непонятных условиях) зависает и не даёт никаких результатов. При этом пересчёт рисков по расписанию тоже перестаёт работать.

Попутное наблюдение: после хотя бы одного зависания метода штатный останов приложения начинает занимать не считанные мгновения, а около половины минуты.

Шаги воспроизведения

1. Запустить приложение с ключом `--enable-risk-ai`.
2. Открыть метод `GET /api/pets/{petId}/recommendedVisits` в [Swagger UI](#).
3. Указать в поле `petId` любой ID существующего в БД питомца, например, `1`.
4. Выполнить запрос кнопкой **Execute**.

Ожидаемый результат: вызов завершается ответом со статусом HTTP 200 и телом с описанием рекомендуемых визитов.

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

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