Урок 15. AI Hub: хранилище контекста и разграничение доступа

Дата: 26 октября 2026 г.
Проект: AI Hub

Репозиторий: https://github.com/im449/ai-hub
Текущий коммит: `938a998`
Статус: этап завершён, изменения опубликованы в GitHub.

1. Цель занятия

Подготовить основу для общего хранилища контекста, которым в будущем смогут пользоваться разные AI-ассистенты. Система должна сохранять важные факты, решения и задачи, при этом не допускать автоматического раскрытия личных данных другим интеграциям.

2. Что реализовано

Хранилище контекста

Добавлен модуль `app/context_store.py`, который поддерживает четыре типа записей:

- `fact` — факты;
- `decision` — принятые решения;
- `task` — задачи;
- `note` — заметки.

Для записей предусмотрены заголовок, содержание, владелец, источник и дата создания.

Разграничение доступа

Реализованы два уровня доступа:

- `private` — личная запись;
- `shared` — запись, которую можно предоставить другим интеграциям.

Важно: установка уровня `shared` сама по себе не предоставляет доступ. Необходимо явно выдать разрешение конкретной интеграции.

Добавлены функции выдачи и отзыва разрешений. Выдать или отозвать доступ может только владелец записи. Новые разрешения не выдаются отключённым интеграциям.

Работа с базой данных

- Созданы таблицы `context_entries` и `context_permissions`.
- Добавлен индекс для ускорения выборки записей владельца.
- Включены ограничения внешних ключей SQLite.
- Подготовлено хранение контекста отдельно от кода приложения.

Документация

Обновлён `README.md`: описаны структура проекта, существующие API-маршруты, хранение базы данных и ограничения текущей реализации.

3. Проверки и результаты

Полный набор автоматических тестов прошёл успешно:

21 тест пройден.

В том числе проверены:

- видимость личных записей;
- отсутствие доступа к чужим записям без разрешения;
- выдача и отзыв разрешений;
- запрет изменения разрешений посторонним владельцем;
- обработка некорректных данных;
- ограничения внешних ключей;
- запрет выдачи разрешения отключённой интеграции.

4. Публикация в GitHub

Изменения сохранены отдельным коммитом:

```
938a998 feat: add context storage and access permissions
```

Коммит успешно отправлен в ветку `main`. Локальная ветка синхронизирована с `origin/main`, незакоммиченных изменений на момент проверки не было.

Ссылки:

- Репозиторий: https://github.com/im449/ai-hub
- Коммит: https://github.com/im449/ai-hub/commit/938a998

5. Что изучили

1. Как разделять код приложения по модулям.
2. Как организовать хранение записей в SQLite.
3. Зачем нужны внешние ключи и ограничения целостности данных.
4. Как разделять владельца записи и получателей доступа.
5. Почему недостаточно просто пометить запись как общедоступную.
6. Как проверять правила доступа автоматическими тестами.
7. Как фиксировать готовый этап разработки в Git и публиковать его в GitHub.

6. Текущее состояние проекта

Контекстное хранилище реализовано как внутренний Python-модуль. Внешние HTTP-маршруты для создания, чтения, предоставления и отзыва доступа к записям ещё не реализованы.

Публичный доступ к AI Hub не открывали. Nginx не изменяли, службу приложения специально не перезапускали.

Это намеренное решение: сначала нужно подготовить и проверить защиту API, а затем подключать ассистентов.

7. План следующего занятия

Основная задача — подготовить HTTP API

1. Спроектировать маршруты для работы с контекстом.
2. Получать владельца записи только из данных аутентифицированной интеграции.
3. Не доверять переданным клиентом идентификаторам владельца.
4. Проверять права на чтение, выдачу и отзыв доступа.
5. Добавить HTTP-тесты на успешные запросы и попытки несанкционированного доступа.
6. Повторно запустить полный набор тестов.
7. Проверить возможность безопасного развёртывания на сервере.

После этого оценить и выполнить подключение первого AI-ассистента. Возможность интеграции зависит от доступных механизмов конкретной платформы.

8. Критерии завершения следующего этапа

- API не работает с контекстом без аутентификации.
- Интеграция видит только собственные и явно разрешённые записи.
- Нельзя читать, изменять или расшаривать чужие записи в обход проверок.
- Проверки безопасности проходят автоматически.
- Изменения опубликованы в GitHub.
- Перед развёртыванием проверен `README.md`.
- Публичный доступ включается только после проверки защиты.

Ориентир по времени: около 2–4 часов на базовый API и его проверки. Подключение первого ассистента оценивается отдельно.

Итог занятия

Создана основа для общего контекстного хранилища AI Hub с явными разрешениями доступа. Изменения протестированы и опубликованы в GitHub. Следующий шаг — безопасный HTTP API, после которого можно переходить к подключению первого ассистента.

Revision #3
Created 2026-10-25 21:07:57 UTC by im
Updated 2026-11-09 22:13:16 UTC by im