Урок 19. Публичная демоверсия AI Hub и обновление документации

Проект: AI Hub
Ветка Git: `feature/helga-chat-20261031`
Статус: выполнено и опубликовано на GitHub

1. Цель урока

Подготовить AI Hub к публичной демонстрации: создать открытую главную страницу и демонстрационный чат, изолировать демонстрационные данные от личного контекста владельца и обновить документацию проекта.

2. Что реализовано

Публичная часть

- Создана главная страница `/`.
- Добавлена публичная демонстрационная страница `/demo`.
- Реализован демонстрационный API `/demo/{assistant_slug}/message`.
- Добавлено переключение между Helga, Alice и Orion.
- Для деморежима используются синтетические данные.
- Публичные сообщения не сохраняются в постоянном хранилище.
- Деморежим не обращается к реальной базе контекста и интеграциям владельца.

Защита и маршрутизация

- Панель владельца перенесена на `/owner`.
- Неавторизованные посетители перенаправляются на `/login`.

- Ссылки в интерфейсе владельца обновлены.
- Для публичного API настроено ограничение частоты запросов через Nginx.
- Демонстрационный режим честно обозначен как локальный сценарий без LLM.

Документация

Обновлён `README.md`:

- добавлены ссылки на публичные страницы;
- описана изоляция демонстрационного режима;
- указаны ограничения Demo Provider;
- удалён уже выполненный пункт из Roadmap.

3. Проверки

Успешно проверены:

- главная страница — HTTP 200;
- публичное демо — HTTP 200;
- страница входа — HTTP 200;
- закрытая панель без авторизации — перенаправление на `/login`;
- запросы к демочату для Helga, Alice и Orion;
- неизвестный профиль помощника — HTTP 404;
- публичный HTTPS-доступ;
- конфигурация Nginx;
- работоспособность systemd-сервиса.

Пользователь подтвердил, что страницы выглядят и работают нормально.

4. Чему научились

1. Разделять публичные и закрытые маршруты FastAPI.
2. Создавать отдельный публичный API для демонстрационного режима.
3. Использовать синтетический контекст, не раскрывая реальные данные.
4. Настраивать ограничение частоты запросов через Nginx.
5. Проверять доступность приложения локально и через HTTPS.
6. Фиксировать изменения отдельными Git-коммитами и публиковать их в feature-ветке.
7. Синхронизировать README с фактическим состоянием проекта.

5. GitHub

Ветка: `feature/helga-chat-20261031`

Созданы и опубликованы коммиты:

- `8a7f404` — `feat: add public AI Hub demo`
- `03452b1` — `docs: update AI Hub README for public demo`

Локальная ветка синхронизирована с `origin/feature/helga-chat-20261031`.

В `main` изменения не переносились.

6. Текущее состояние проекта

AI Hub располагает:

- постоянным хранилищем контекста;
- системой интеграций и разграничением доступа;
- защищённой панелью владельца;
- профилями Helga, Alice и Orion;
- интерфейсом чата;
- публичной изолированной демонстрацией.

Публичный многоассистентский чат пока использует локальный Demo Provider, а не полноценную генерацию ответов через LLM.

7. Следующий урок

Подключение настоящей LLM к Helga.

План:

1. Подключить LLM-провайдера через серверный API.
2. Передавать модели сообщения и разрешённый контекст AI Hub.
3. Сохранить разграничение доступа и защиту секретов.
4. Проверить полноценный диалог с Helga.
5. После успешной реализации подготовить общий механизм для Alice и Orion.

8. Итог

Публичная демонстрация AI Hub создана, проверена и опубликована в GitHub. Документация приведена в соответствие с текущим состоянием проекта.

Урок №18 завершён.

Revision #2
Created 2026-11-01 20:27:58 UTC by im
Updated 2026-11-09 22:13:53 UTC by im