Инструкция по установке и эксплуатации ПО «GYMBOT»

Документ подготовлен для внесения ПО в Единый реестр российских программ (реестр Минцифры России) и содержит сведения, необходимые для установки и эксплуатации программного обеспечения.

Сопутствующий документ: ПО_Описание_и_функциональные_характеристики.md.

---

1. Системные требования

- **ОС:** Linux — Ubuntu 22.04/24.04 LTS или Debian (входят в перечень разрешённых ОС Минцифры).
- **Python:** версия 3.10 и выше (на рабочем сервере — Python 3.12).
- **Ресурсы:** от 2 ГБ ОЗУ, 2 vCPU; от 15 ГБ диска (из них около 1 ГБ — медиа-библиотека).
- **Сеть:** исходящий доступ в интернет к программным интерфейсам каналов взаимодействия (ВКонтакте, MAX), к программному интерфейсу языковых моделей Cloud.ru Evolution Foundation Models (ООО «Облачные технологии») и к платёжной системе CloudPayments.
- **Дополнительно:** пакеты `nginx` и `certbot` для публикации веб-канала и панели администратора по HTTPS.

2. Состав дистрибутива

Дистрибутив представляет собой набор исходных файлов на языке Python и вспомогательных ресурсов:

- `handlers/`, `services/`, `db/` — прикладное ядро: сценарии взаимодействия, генерация программы тренировок и питания, планировщик рассылок, работа с данными, интеграции;
- `vk/`, `max/`, `web/` — каналы взаимодействия с пользователем: ВКонтакте, MAX и собственный веб-чат на сайте правообладателя. Каждый канал содержит адаптер (`bridge/`), настройки и автотесты; прикладное ядро при этом единое и от канала не зависит;
- `viewer/` — панель администратора: переписка, пользователи, аналитика, партнёрская программа;
- `vk/requirements.txt`, `max/requirements.txt`, `web/requirements.txt`, `viewer/requirements.txt` — списки зависимостей (по одному на компонент);
- `db/schema.sql` — схема базы данных;
- `max/certs/` — корневые сертификаты УЦ Минцифры (их требует программный интерфейс MAX);
- `scripts/` — сервисные сценарии (очистка журналов, наполнение справочника упражнений);
- `assets/` — медиа-контент (фото/видео/аудио/PDF), поставляется отдельным пакетом;
- `.env.example` — образец файла конфигурации.

3. Установка

1. Установить системные пакеты:

   ```
   sudo apt update
   sudo apt install -y python3 python3-venv python3-pip nginx certbot python3-certbot-
   nginx sqlite3
   ```

2. Разместить исходные файлы (например, в `/opt/gymbot`) и перейти в каталог.
3. Создать виртуальное окружение для каждого канала. Окружения раздельные: клиентские библиотеки каналов предъявляют несовместимые требования к версиям общих зависимостей.

   ```
   python3 -m venv .venv-vk  && .venv-vk/bin/pip  install -r vk/requirements.txt
   python3 -m venv .venv-max && .venv-max/bin/pip install -r max/requirements.txt
   python3 -m venv .venv-web && .venv-web/bin/pip install -r web/requirements.txt
   ```

4. Создать окружение панели администратора (ей нужен и клиент языковой модели — для подсказок справочника продуктов в журнале питания):

```
python3 -m venv .venv-viewer && .venv-viewer/bin/pip install -r
viewer/requirements.txt -r max/requirements.txt
```

5. Настроить конфигурацию (см. раздел 4): скопировать `.env.example` в `.env` и заполнить значения.
6. База данных создаётся автоматически при первом запуске (применяется `db/schema.sql`); отдельной команды миграции не требуется.

Достаточно установить только те каналы, которые планируется использовать: они независимы и запускаются отдельными процессами.

4. Конфигурация

Конфигурация задаётся переменными окружения (файл `.env` рядом с кодом или `EnvironmentFile` в systemd). Ключевые переменные:

Переменная	Назначение	Обязательна
`LLM_API_KEY`	Ключ доступа сервисного аккаунта Cloud.ru к языковым моделям	Да (для диалогового ассистента и анализа фото)
`LLM_API_URL`	Адрес программного интерфейса моделей	Нет (`https://foundation-models.api.cloud.ru/v1`)
`LLM_MODEL`	Модель диалогового ассистента	Нет (`GigaChat/GigaChat-3-Pro`)
`LLM_VISION_MODEL`	Модель анализа фотографии блюда	Нет (`Qwen/Qwen3.6-35B-A3B`)
`LLM_VISION_EXTRA`	Дополнительные параметры запроса анализа фото (JSON)	Нет
`PLATE_ENABLED`	Включение анализа фотографии блюда (`1` — включено)	Нет (выключено)
`PLATE_FREE_TOTAL` / `PLATE_DAILY_LIMIT`	Число анализов фото: до оплаты — всего, после оплаты — в сутки	Нет (3 и 5)
`FOOD_LINK_SECRET`	Ключ подписи личных ссылок на журнал калорий; общий у каналов и панели. Смена делает недействительными уже выданные ссылки	Для журнала калорий
`FOOD_LINK_BASE`	Публичный адрес, по которому открывается журнал (панель отдаёт его по пути `/app/`), например `https://quiz.gym-bot.ru`	Для журнала калорий
`FOOD_LINK_TTL_HOURS`	Срок действия ссылки на журнал, часов	Нет (24)
`VK_GROUP_TOKEN`	Ключ доступа сообщества ВКонтакте	Да (для канала «ВКонтакте»)
`MAX_BOT_TOKEN`	Ключ доступа бота MAX	Да (для канала «MAX»)
`MAX_BOT_USERNAME`	Имя бота MAX (используется в ссылке-приглашении)	Нет
`WEB_APP_SECRET`	Ключ подписи идентификатора посетителя веб-чата. **Менять нельзя:** смена обнуляет прогресс всех посетителей	Да (для веб-канала)
`WEB_HOST` / `WEB_PORT`	Адрес и порт веб-канала	Нет (127.0.0.1:5002)
`WEB_PUBLIC_URL`	Публичный адрес веб-канала (ссылки и переходы)	Нет
`WEB_BOT_TITLE` / `WEB_BOT_LOGO`	Название и логотип в интерфейсе веб-чата	Нет
`VK_DB` / `MAX_DB` / `WEB_DB`	Пути к файлам базы данных каналов (у каждого канала своя база)	Нет
`VK_LOG` / `MAX_LOG` / `WEB_LOG`	Пути к файлам журналов	Нет
`FITBOT_CACHE_SUFFIX`	Суффикс файла кэша идентификаторов медиа; **у каждого канала свой** (идентификаторы выдаёт сам канал)	Нет
`INVITE_URL`	Ссылка-приглашение, используемая в сценарии рекомендации друзьям	Нет
`PRIVACY_URL`	Адрес действующей политики обработки персональных данных — показывается на экране согласия	Нет (есть значение по умолчанию)

Переменная	Назначение	Обязательна
PRIVACY_VERSION	Редакция политики (например, 2026-07-29). При изменении значения согласие запрашивается у пользователей заново	Нет
SKIP_ASSISTANT_STEP	Отключение экрана подключения ассистента в сценарии онбординга	Нет
PAY_URL_MAIN / PAY_URL_DRIP	Адреса оплаты для тарифов	Нет
SUPPORT_HANDLE / ASSISTANT_HANDLE	Контакты поддержки и персонального ассистента	Нет
VIEWER_PASSWORD	Пароль/инвайт-код доступа к панели администратора (пусто → авторизация выключена)	Для рабочей панели
VIEWER_SECRET	Ключ подписи сессий панели	Нет (генерируется)
VIEWER_HOST / VIEWER_PORT	Адрес и порт панели	Нет (127.0.0.1:5001)
PARTNER_INVITE_CODE	Код регистрации партнёров	Нет
CP_PUBLIC_ID, CP_API_SECRET	Реквизиты CloudPayments для сверки оплат (пусто → выключено)	Нет
GSHEET_KEY_FILE, PAYMENTS_GSHEET_ID	Внешняя таблица оплат — **опционально, по умолчанию отключено**	Нет

Обязательный минимум: ключ LLM_API_KEY и ключ доступа того канала, который запускается (проверяется функциями assert_ready_vk / assert_ready_max / assert_ready_web в настройках соответствующего канала). Секреты (ключи доступа, пароли) в дистрибутив не помещаются.

5. Запуск и эксплуатация

5.1. Запуск

```
.venv-vk/bin/python3     -m vk.main      # канал «ВКонтакте»
.venv-max/bin/python3    -m max.main     # канал «MAX»
.venv-web/bin/python3    -m web.main     # веб-чат на сайте правообладателя
.venv-viewer/bin/python3 viewer/app.py   # панель администратора, http://127.0.0.1:5001
```

На рабочем сервере запуск обеспечивают службы systemd: по одной на канал (gymbot-vk, gymbot-max, gymbot-web), служба панели администратора (gymbot-viewer), а также задание периодической очистки журналов. Управление — стандартными командами systemctl start|stop|restart|status <служба>.

Пример модуля службы:

```
[Unit]
Description=GYMBOT — канал ВКонтакте
After=network-online.target

[Service]
User=gymbot
WorkingDirectory=/opt/gymbot
EnvironmentFile=/opt/gymbot.env
ExecStart=/opt/gymbot-vk-venv/bin/python3 -m vk.main
Restart=always
RestartSec=5
MemoryMax=900M

[Install]
WantedBy=multi-user.target
```

5.2. Подключение каналов

Каналы подключаются через панель администратора (вкладка «Боты»): введённый ключ доступа проверяется обращением к программному интерфейсу канала, для веб-канала — проверкой публичного адреса. Поддерживаются активация и отключение канала, смена ключа доступа с сохранением базы данных, управление ссылками с метками рекламных кампаний.

5.3. Обслуживание

- **Журналы:** файлы в каталоге `logs/` (по одному на канал) и системный журнал `journalctl`.
- **Режим базы данных:** SQLite работает в режиме журналирования **WAL** — это позволяет каналу и панели администратора обращаться к базе одновременно. Рядом с файлом `<база>.db` появляются служебные файлы `-wal` и `-shm`; при резервном копировании копировать их вместе с основным файлом либо предварительно останавливать службу.
- **Состояние диалогов** хранится в самой базе, поэтому перезапуск и обновление не сбивают пользователей с текущего шага сценария.
- **Резервное копирование:** файлы баз данных (`*.db`) — обычные файлы SQLite; копируются штатными средствами при остановленной службе либо вместе со служебными файлами WAL.
- **Очистка журналов сообщений:** периодическое задание (`scripts/prune_message_log.py`) удаляет записи старше заданного срока хранения; персональные данные защищены правами доступа `600` на файлах баз данных и журналов.
- **Обновление:** заменить файлы исходного кода новой версией и перезапустить службы; данные (файлы БД) при этом сохраняются, состояние диалогов пользователей не теряется.

5.4. Остановка

```
systemctl stop gymbot-vk gymbot-max gymbot-web gymbot-viewer   # на сервере
# либо локально:
pkill -9 -f "vk.main"; pkill -9 -f "max.main"; pkill -9 -f "web.main"
```

6. Проверка работоспособности

- Запуск канала сопровождается в журнале сообщением о старте и началом получения событий.
- Веб-канал отвечает по адресу `WEB_HOST:WEB_PORT` (на сервере — по HTTPS через nginx); открытие страницы должно начинать диалог.
- Панель администратора доступна по адресу `VIEWER_HOST:VIEWER_PORT`.
- Отправка сообщения в канале должна инициировать сценарий анкеты; диалоговый ассистент отвечает через языковую модель Cloud.ru.
- Фотография блюда, отправленная в любом канале (в веб-чате — кнопкой с изображением камеры), должна вернуть состав, оценку калорийности и рекомендацию. Снимок передаётся модели внутри запроса и на сервере не сохраняется.
- Проверка доступа к языковым моделям без запуска каналов — `scripts/probe_llm.py`.
- Автотесты (по каналам и по прикладному ядру) запускаются командами, приведёнными в инструкции для экспертной проверки экземпляра.

Все компоненты ПО распространяются под свободными лицензиями; ПО функционирует на ОС Linux из перечня разрешённых, без установки запрещённых иностранных программ. Полный перечень компонентов и лицензий — в документе описания функциональных характеристик.