

Настройка в GitHub

1. Репозиторий кода

- Все скрипты (`Child_deduplication.py`, `Duplicate_Check.py` и др.), а также вспомогательные файлы (`config_loader.py`, `system_config.json`, HTML-страницы) хранятся в репозитории GitHub.
- Этот репозиторий — основное место хранения кода: при каждом изменении в GitHub сервис Render автоматически обновляет развернутую версию.

2. Управление секретами

- Конфиденциальные данные, такие как вебхук Bitrix, **не хранятся в GitHub**.
- Скрипты получают их через переменные окружения (например, `B24_WEBHOOK_URL`).

3. Контроль версий

- Можно использовать ветки для разработки и тестирования, а потом объединять их в `main`, когда код готов.
- GitHub сохраняет историю изменений, что позволяет откатить изменения при необходимости.

Настройка в Render

1. Веб-сервис

- В Render код разворачивается как **Web Service**.
- Render автоматически устанавливает зависимости (из `requirements.txt`) и запускает Flask-приложение, которое отображает страницу со скриптами.

2. Переменные окружения

- В настройках Render → Environment добавляются:
 - `B24_WEBHOOK_URL` → вебхук Bitrix (секретное значение).
 - при необходимости `CONFIG_DIR` → путь к папке для хранения конфигурации на постоянном диске.

3. Постоянный диск (опционально)

- Чтобы изменения в `system_config.json` сохранялись после обновлений, можно подключить в Render постоянный диск.
- Переменная `CONFIG_DIR` указывает путь к этому диску, и скрипты читают/записывают настройки именно туда.

4. Автоматическое обновление

- Render связывается с репозиторием GitHub.
- При каждом изменении в ветке `main` Render пересобирает и разворачивает приложение.

5. Запуск скриптов

- После развёртывания открывается URL приложения Render.
- На странице `scripts.html` отображаются все скрипты с кнопками **Запустить**.

- Сотрудники могут запускать скрипты напрямую через веб-страницу, без командной строки.