

Переменные окружения для Netlify

После создания сайта в Netlify, необходимо добавить следующие переменные окружения в настройках сайта:

Путь: Site settings → Environment variables

Обязательные переменные:

```
# База данных (замените на ваши данные)
DATABASE_URL="postgresql://username:password@host:port/database_name"

# NextAuth настройки
NEXTAUTH_URL="https://www.upak.space"
NEXTAUTH_SECRET="your-production-secret-key-here"

# API бэкенда UPAK
NEXT_PUBLIC_API_URL="https://upak.tech:8001"

# Окружение
NODE_ENV="production"
```

Как добавить переменные в Netlify:

1. Откройте ваш сайт в панели Netlify
2. Перейдите в **Site settings** (настройки сайта)
3. Найдите раздел **Environment variables** (переменные окружения)
4. Нажмите **Add variable** для каждой переменной
5. Введите **Key** (имя) и **Value** (значение)
6. Сохраните изменения
7. Сделайте **Redeploy** сайта



Важно:

- Замените значения переменных на реальные данные вашего проекта
- NEXTAUTH_SECRET должен быть уникальной строкой (можно сгенерировать на openssl.org)
- DATABASE_URL должен указывать на вашу реальную PostgreSQL базу данных



Версия Node.js:

Проект использует Node.js v20 (указано в .nvmrc). Netlify автоматически использует эту версию для сборки.



База данных:

Проект использует Prisma ORM с PostgreSQL. Особенности сборки:

Для статичных страниц (без базы данных):

- Сборка пройдет успешно даже без DATABASE_URL
- Prisma клиент не инициализируется автоматически
- API роуты требующие БД не будут работать

Для полной функциональности с базой данных:

1. База данных должна быть доступна из Интернета
2. DATABASE_URL корректно настроен
3. Все таблицы созданы (`npx prisma db push`)

**Специальная сборка для Netlify:**

Проект использует несколько подходов для сборки:

1. `build-netlify-production.js` (рекомендуемый):

- Использует `@netlify/plugin-nextjs` для правильной обработки SSR
- Автоматически настраивает конфигурацию для Netlify
- Поддерживает динамические API маршруты
- Мокает Prisma если нет DATABASE_URL

2. `build-static.js` (резервный):

- Создает полностью статический export
- Исключает все API routes
- Подходит для демо-сайтов без backend

3. Ключевые исправления:

- ☒ Динамические страницы переведены на client-side
- ☒ Убрана серверная логика из компонентов времени сборки
- ☒ API routes работают только в runtime, не в build time