

Auth Flow

Общая схема аутентификации NextAuth.js + SurrealDB

Поток использует стандартный механизм **OAuth 2.0/OpenID Connect (OIDC)**, управляемый **NextAuth.js**, и кастомный **адаптер** для интеграции с **SurrealDB**.

1. Начало процесса

Шаг	Участник	Действие
1.1	Пользователь	Нажимает кнопку "Войти через Google" или "Войти через GitHub" на фронтенде Next.js.
1.2	Next.js (Клиент)	Иницирует вызов <code>signIn('google')</code> или <code>signIn('github')</code> из NextAuth.js.
1.3	Next.js (Сервер, NextAuth.js API)	Получает запрос и перенаправляет пользователя на страницу аутентификации выбранного OAuth-провайдера (Google/GitHub).

2. Аутентификация у провайдера (Google/GitHub)

Шаг	Участник	Действие	Обработка ошибок
2.1	Пользователь	Вводит свои учетные данные (если не авторизован) и предоставляет разрешение вашему приложению.	Ошибка: Неправильны учетные данные или отказ в разрешении.
2.2	Провайдер	Аутентифицирует пользователя и отправляет код авторизации обратно в конечную точку обратного вызова NextAuth.js (<code>/api/auth/callback/[provider]</code>).	Ошибка: Истечение срока действия код или недопустимы <code>redirect_ur</code>

Шаг	Участник	Действие	Обработка ошибок
2.3	Next.js (Сервер, NextAuth.js API)	Получает код и обменивает его на токены доступа/идентификации и профиль пользователя через API-интерфейс провайдера (запрос с использованием секретного ключа).	

3. Обработка данных и сессии (NextAuth.js + SurrealDB)

Этот этап критически важен, и в нем задействован **Кастомный Адаптер SurrealDB** NextAuth.js.

Шаг	Участник	Действие	Обработка ошибок
3.1	NextAuth.js (Адаптер)	Проверяет, существует ли пользователь в SurrealDB по предоставленному ID провайдера. (Вызывает метод <code>getUserByAccount</code>).	
3.2	SurrealDB	Запрос: Найти пользователя по <code>providerAccountId</code> и <code>provider</code> в таблице <code>account</code> .	Ошибка: Сбой подключения к БД, ошибки запроса (таймаут).
3.3	NextAuth.js (Адаптер)	Если пользователь найден: Обновляет его данные (вызывает <code>updateUser</code>) и создает/обновляет запись сессии (вызывает <code>createSession</code>) в SurrealDB. Если пользователь НЕ найден: Создает нового пользователя (вызывает <code>createUser</code>), соответствующую запись аккаунта (вызывает <code>linkAccount</code>) и затем сессию.	Ошибка: Ошибка записи/обновления в БД.

Шаг	Участник	Действие	Обработка ошибок
3.4	NextAuth.js	Создает JWT-токен (если используется стратегия <code>jwt</code>) или использует ID сессии (если используется стратегия <code>database</code>).	
3.5	Next.js (Сервер)	Устанавливает HTTP-куки для сессии/JWT в браузере пользователя.	
3.6	Next.js (Сервер)	Перенаправляет пользователя обратно на фронтенд (на главную страницу или страницу, с которой был начат вход).	Ошибка: Ошибка перенаправления.

4. Завершение и использование

Шаг	Участник	Действие
4.1	Next.js (Клиент)	Фронтенд использует хук <code>useSession()</code> для проверки наличия сессии (куки).
4.2	Next.js (Сервер)	При запросах данных сервер проверяет сессию с помощью <code>getSession()</code> или middleware для получения информации о пользователе и управления доступом к API (включая SurrealDB).

Обработка ошибок (Сводка)

Обработка ошибок в NextAuth.js в основном происходит через перенаправление с URL-параметром `?error=...`.

1. Ошибки провайдера/OAuth (Шаг 2)

Если аутентификация у провайдера не удалась:

- **Действие:** NextAuth.js перенаправляет пользователя на URL-адрес `/api/auth/error?error=...`.
- **Пример:** `?error=AccessDenied` (пользователь отказал в доступе) или `?error=OAuthCallback` (сбой при обмене токеном).

- **Решение:** Вы должны отобразить **понятное сообщение об ошибке** на странице `/api/auth/error` на основе значения параметра `error`.

2. Ошибки адаптера/базы данных (Шаг 3)

Если SurrealDB не работает или возникает ошибка записи/чтения:

- **Действие:** Адаптер SurrealDB или коллбэк **бросает исключение**. NextAuth.js перехватывает его.
- **Пример:** `DatabaseError` или `CallbackError`.
- **Решение:** NextAuth.js обычно перенаправляет на `/api/auth/error?error=Configuration` или внутреннюю ошибку, которую необходимо отследить в **логах сервера**.