Auth Service API

Base URL

- Local: http://localhost:${PORT}
- Default PORT: 5000 (or use PORT in .env)

Test Frontend

A simple HTML test interface is available at:

- http://localhost:${PORT}/

The test frontend provides forms for signup, signin, Google OAuth, and account deletion.

Overview

This service provides email/password auth, Google OAuth, and account deletion. It uses Supabase Auth under the hood and issues a service JWT for downstream services on sign-in.

Authentication

Protected routes require a Bearer token:

Authorization: Bearer <access_token>

For Google OAuth, the backend sets httpOnly cookies on callback:

- sb-access-token
- sb-refresh-token

Environment Variables

Required

- SUPABASE_URL
- SUPABASE_SERVICE_ROLE_KEY
- JWT_SECRET
- GOOGLE_CLIENT_ID
- GOOGLE_CLIENT_SECRET
- BREVO_API_KEY
- BREVO_SENDER_NAME
- BREVO_SENDER_EMAIL

Optional

- PORT
- FRONTEND_URL
- NODE_ENV (production enables secure cookies)

Endpoints

POST /api/v1/auth/signup

Create a user and send a verification link in the welcome email.

Request body

```
{
  "email": "user@example.com",
  "password": "Password123!",
  "frontendUrl": "https://app.example.com"
}
```

Rules

- frontendUrl must be a valid https:// URL.
- If frontendUrl is omitted, FRONTEND_URL is used.
- Verification redirect is ${frontendUrl}/verify.

Response 201

```
{
  "message": "User created",
  "user_id": "<uuid>"
}
```

Errors

- 400 invalid input or Supabase errors

curl

```
curl -X POST http://localhost:5000/api/v1/auth/signup \
  -H "Content-Type: application/json" \
  -d '{"email":"user@example.com","password":"Password123!","frontendUrl":"https://app.example.com"}'
```

POST /api/v1/auth/signin

Sign in with email/password. Returns service JWT plus Supabase refresh token.

Request body

```
{
  "email": "user@example.com",
  "password": "Password123!"
}
```

Response 200

```
{
  "user": { "id": "<uuid>", "email": "user@example.com" },
  "access_token": "<service_jwt>",
  "refresh_token": "<supabase_refresh_token>"
}
```

Errors

- 401 invalid credentials

curl

```
curl -X POST http://localhost:5000/api/v1/auth/signin \
  -H "Content-Type: application/json" \
  -d '{"email":"user@example.com","password":"Password123!"}'
```

DELETE /api/v1/auth/delete-account

Delete the current user. Requires Bearer token.

Headers

- Authorization: Bearer <access_token>

Response 200

```
{
  "message": "Account deleted"
}
```

Errors

- 401 missing or invalid token
- 500 server error

curl

```
curl -X DELETE http://localhost:5000/api/v1/auth/delete-account \
  -H "Authorization: Bearer <access_token>"
```

GET /api/v1/auth/google

Start Google OAuth. Redirects to Google consent screen.

Query params

- frontendUrl (required): https://app.example.com

Response

- 302 redirect to Google

curl (follow redirects)

```
curl -L "http://localhost:5000/api/v1/auth/google?
frontendUrl=https://app.example.com"
```

GET /api/v1/auth/google/callback

Handle Google OAuth callback from Supabase. Sets httpOnly cookies and redirects to the frontend.

Query params

- code (required)
- frontendUrl (required)

Behavior

- Exchanges code for session with Supabase.
- Sets cookies: sb-access-token, sb-refresh-token.
- Redirects to frontendUrl (or FRONTEND_URL) with path /auth/callback if none is provided.

Errors

- 400 missing code or invalid frontendUrl

Notes

- frontendUrl must be https:// only.
- For local development, use a valid https URL or adjust validation logic.
- Cookies are secure and SameSite=None in production, Lax in non-production.