

NeoZen

AuthN + AuthZ

Sign-in and permissions for Python apps.

NeoZen handles authentication and authorization: as a library inside your app, or as an identity provider in front of all of them.

```
# a library, in your app
$ uv add "neozen[flask]"

# or a service, in front of your apps
$ docker compose up -d

# the same code, either way
$ neozen serve
```

WHAT IT DOES

Everything after "who are you?"

Sign in

Passwords, magic links, passkeys, TOTP and email codes. Google and GitHub sign-in.

Sessions

Hashed tokens with a sliding expiry, revoked one at a time or all at once.

Permissions

Roles, permissions and policies. Groups, and invitations by email.

Single sign-on

An OAuth 2.1 server and an OpenID Connect provider for your other apps.

Provisioning

SCIM 2.0 for users and groups. LDAP and Active Directory as a user store.

Administration

A web console for users, roles, sessions and the audit log. CSV import and export.

AS A LIBRARY

It fits **your** app, not the reverse

Your framework

Adapters for Flask, Litestar and Starlette. The core has no web framework in it, so anything else can call it.

Your database

SQLAlchemy, Peewee, Tortoise ORM or plain SQL, behind one repository protocol.

Your User model

NeoZen depends on protocols, not base classes: keep the model you have.

Take only what you need: one sign-in factor and sessions, or the whole identity provider.

AS A SERVICE

Or run it on its own

- ✓ An admin console, ready on first boot
- ✓ OpenID Connect and SCIM for your other apps
- ✓ PostgreSQL, its schema created on first start
- ✓ Health and readiness checks for your monitoring
- ✓ Docker Compose, or a Helm chart for Kubernetes

```
$ cd deploy/docker
$ docker compose up -d

# the first admin's password
$ docker compose logs neozen \
    | grep BOOTSTRAP -A 6

# then open the console
$ open http://localhost:8000/login
```

SECURITY

Secure by default

Out of the box

Argon2id password hashing. Timing-safe comparisons. Session tokens stored hashed, and rotated at sign-in. PKCE, and refresh tokens that detect replay.

One setting each

Rate limiting on sign-in and registration. Account lockout after failed attempts. An audit log of security events.

Tested by more than 1,600 tests, type-checked by mypy, ty and pyrefly.

GET STARTED

Five minutes to your first sign-in

Developer Quickstart

A Litestar app with registration, sign-in and sessions.

```
uv add neozen
```

Self-Hosted Quickstart

NeoZen as an identity provider, with Docker Compose.

```
docker compose up -d
```

neozen.lab.abilian.com · Apache 2.0, no paid tier · 0.9 is the first public release.