Supabase CLI Local stack and deploys

CONCEPTS

The local stack runs in Docker containers: start Docker first, and expect the first start to pull images.

Local stack supabase start
↓ db diff
Migration supabase/migrations
↓ db push
Linked project remote database
↓ gen types
Typed client your app

Most db and migration commands take --local (default), --linked or --db-url <url> to pick the target.

LOCAL STACK

supabase init
Create supabase/ config folder

supabase start
Start local Supabase containers

supabase status
Show URLs and keys of the local stack

supabase stop
Stop the local containers

supabase stop --no-backup
Stop and wipe local data

supabase start -x *<services>*
Start without some services

supabase status -o env
Print status as env vars

LOGIN & LINK

supabase login
Log in with an access token

supabase login --token *<token>*
Log in non-interactively

supabase link --project-ref *<ref>*
Link folder to a hosted project

supabase projects list
List your projects

supabase projects api-keys --project-ref *<ref>*
Show project API keys

supabase orgs list
List organizations

DATABASE

supabase db diff -f *<name>*
Write local schema changes as migration

supabase db diff --linked
Diff against the linked project

supabase db push
Apply pending migrations to linked DB

supabase db push --dry-run
Show what would be pushed

supabase db pull
Fetch remote schema as a migration

supabase db reset
Recreate local DB from migrations and seed

supabase db reset --no-seed
Reset without running seed.sql

supabase db dump -f *<file>*
Dump schema to a file

supabase db dump --data-only -f *<file>*
Dump data only

supabase db lint
Check schema for errors

MIGRATIONS

supabase migration new *<name>*
Create an empty timestamped migration

supabase migration list
Compare local and remote migrations

supabase migration up
Apply pending migrations locally

supabase migration repair *<version>* **--status applied**
Fix remote migration history

EDGE FUNCTIONS

supabase functions new *<name>*
Scaffold a function

supabase functions serve
Serve functions locally

supabase functions deploy *<name>*
Deploy one function

supabase functions deploy
Deploy all functions

EDGE FUNCTIONS (cont.)

supabase functions serve --env-file *<file>*
Serve with env vars from a file

supabase functions list
List deployed functions

SECRETS

supabase secrets set *<NAME>*=*<value>*
Set a function secret

supabase secrets set --env-file *<file>*
Set secrets from a file

supabase secrets list
List secret names

TYPES & TESTS

supabase gen types --lang typescript --local
TypeScript types from local DB

supabase gen types --lang typescript --linked
TypeScript types from linked project

supabase test db
Run pgTAP tests

RECIPES

First local setup

```
supabase init
supabase start
supabase status
```

Schema change to production

```
supabase migration new add_posts
# edit the migration SQL
supabase db reset
supabase db push --dry-run
supabase db push
```

Capture Studio edits as migration

```
supabase db diff -f <name>
supabase db reset
supabase migration list
```

Ship a function

```
supabase functions new hello
supabase functions serve
supabase secrets set
API_KEY=<value>
supabase functions deploy hello
```

HELP

supabase *<command>* **--help**
Usage for a command