
Supabase Storage Engine

coverage 84%

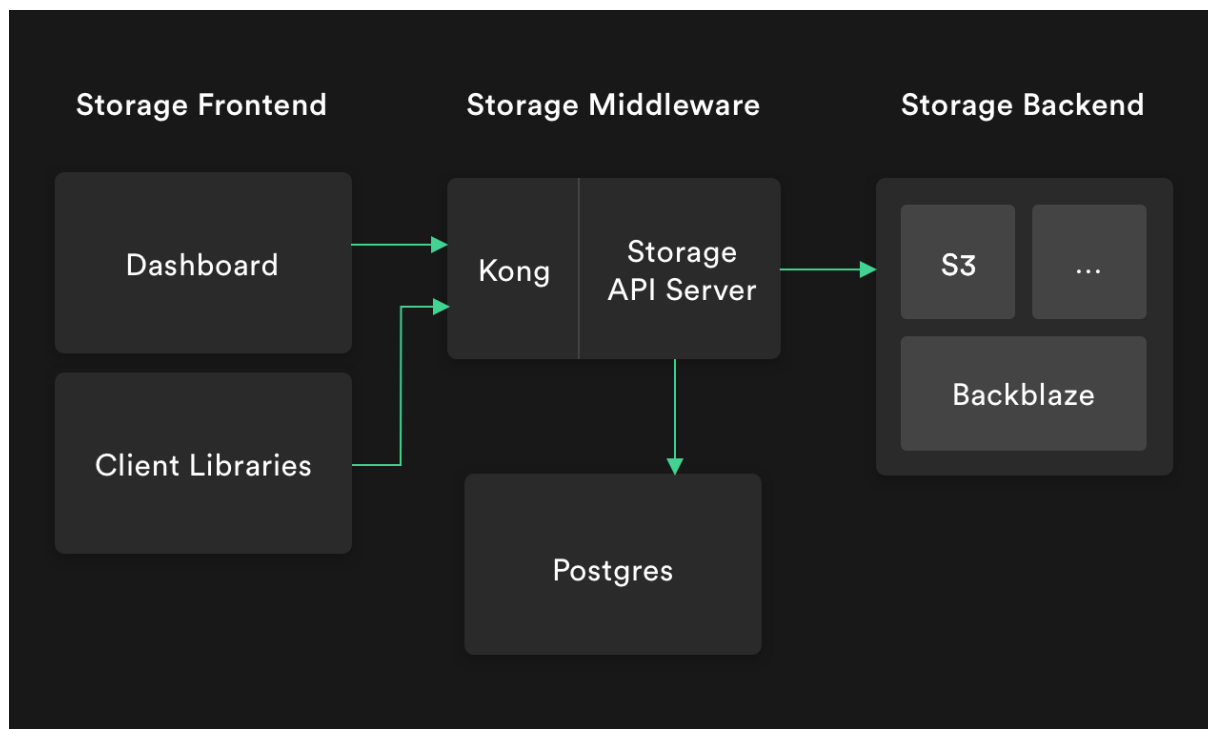
A scalable, light-weight object storage service.

Read this post on why we decided to build a new object storage service.

- Multi-protocol support (HTTP, TUS, S3)
- Uses Postgres as its datastore for storing metadata
- Authorization rules are written as Postgres Row Level Security policies
- Integrates with S3 Compatible Storages
- Extremely lightweight and performant

Supported Protocols

- ☑ HTTP/REST
- ☑ TUS Resumable Upload
- ☑ S3 Compatible API



Documentation

- OpenAPI Spec

-
- Storage Guides
 - Client library

Development

- Copy `.env.sample` to `.env` file.
- Copy `.env.test.sample` to `.env.test`.

```
1 cp .env.sample .env && cp .env.test.sample .env.test
```

Your root directory should now have both `.env` and `.env.test` files.

- Then run the following:

```
1 # this sets up a postgres database and postgres locally via docker
2 npm run infra:restart
3 # Start the storage server
4 npm run dev
```

The server should now be running at `http://localhost:5000/`

The following request should insert and return the list of buckets.

```
1 # insert a bucket named avatars
2 curl --location --request POST 'http://localhost:5000/bucket' \
3   --header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
      eyJyb2xlIjoic2VydmV9b2xlIiwiaWF0IjoxNjE5NTMxOTg1LCJleHAiOjE5MjkwMDc5ODV9
      .th840KK0Iz8QchDyXZRrojmKSEZ-OuitQm_5DvLiSic' \
4   --header 'Content-Type: application/json' \
5   --data-raw '{
6     "name": "avatars"
7   }'
8
9 # get buckets
10 curl --location --request GET 'http://localhost:5000/bucket' \
11   --header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
      eyJyb2xlIjoic2VydmV9b2xlIiwiaWF0IjoxNjE5NTMxOTg1LCJleHAiOjE5MjkwMDc5ODV9
      .th840KK0Iz8QchDyXZRrojmKSEZ-OuitQm_5DvLiSic'
```

Testing

To perform your tests you can run the following command: `npm test`