

Documentation

- As previously done, Extract the source code zip file and open with VS code.
- Open the terminal and run “npm install” to install the dependencies.
- Next Rename the .env.example file as .env
- In there you have to fill some information.
- First grab your openAI API key and set it there.
- Next create a new project in supabase and grab your SUPABASE URL and SUPABASE ANNON KEY.
- fill them in corresponding spaces in env file.

In supabase there are some few steps to do.

- Go to you project dashboard.
- Go to SQL editor section and paste the following code to create the table name ‘files_info’ and click on Run button
-

```
CREATE TABLE files_info (  
  
  id SERIAL PRIMARY KEY,  
  
  name VARCHAR(255) NOT NULL,  
  
  batches INT NOT NULL,  
  
  text_size INT NOT NULL,  
  
  status VARCHAR(50) DEFAULT 'Pending'  
  
);
```

```
codehash-001 Free / ingest-app / Enable branching

Ask Supabase AI to modify your query

1 CREATE TABLE files_info (
2   id SERIAL PRIMARY KEY,
3   name VARCHAR(255) NOT NULL,
4   batches INT NOT NULL,
5   text_size INT NOT NULL,
6   status VARCHAR(50) DEFAULT 'Pending'
7 );
8
```

- Next we have to create the document table and other relevant functions for Embedding documents
- To do that again go to SQL editor section and paste the following code. **(I will attach a text file with relevant SQL codes for better understanding)**

-- Enable the pgvector extension to work with embedding vectors

```
create extension vector;
```

-- Create a table to store your documents

```
create table documents (
```

```
id bigserial primary key,
```

```
content text, -- corresponds to Document.pageContent
```

```
metadata jsonb, -- corresponds to Document.metadata
```

```
embedding vector(1536) -- 1536 works for OpenAI embeddings, change if needed
```

```
);
```

-- Create a function to search for documents

```
create function matching_docs (
```

```
query_embedding vector(1536),
```

```
match_count int DEFAULT null,

filter jsonb DEFAULT '{}'

) returns table (

id bigint,

content text,

metadata jsonb,

embedding jsonb,

similarity float

)

language plpgsql

as $$

#variable_conflict use_column

begin

return query

select

id,

content,

metadata,

(embedding::text)::jsonb as embedding,

1 - (documents.embedding <=> query_embedding) as similarity

from documents

where metadata @> filter

order by documents.embedding <=> query_embedding

limit match_count;

end;
```

\$\$;

- Then we have set up the supabase storage for upload documents (what needed for ingest in future if any case) .
- To do that go to storage tab of Supabase dashboard.
- Click on “New bucket” button.
- you will see a pop up and give the name as “files” and make sure to toggle the public bucket button.


Create storage bucket


Name of bucket Buckets cannot be renamed once created.

files

Only lowercase letters, numbers, dots, and hyphens

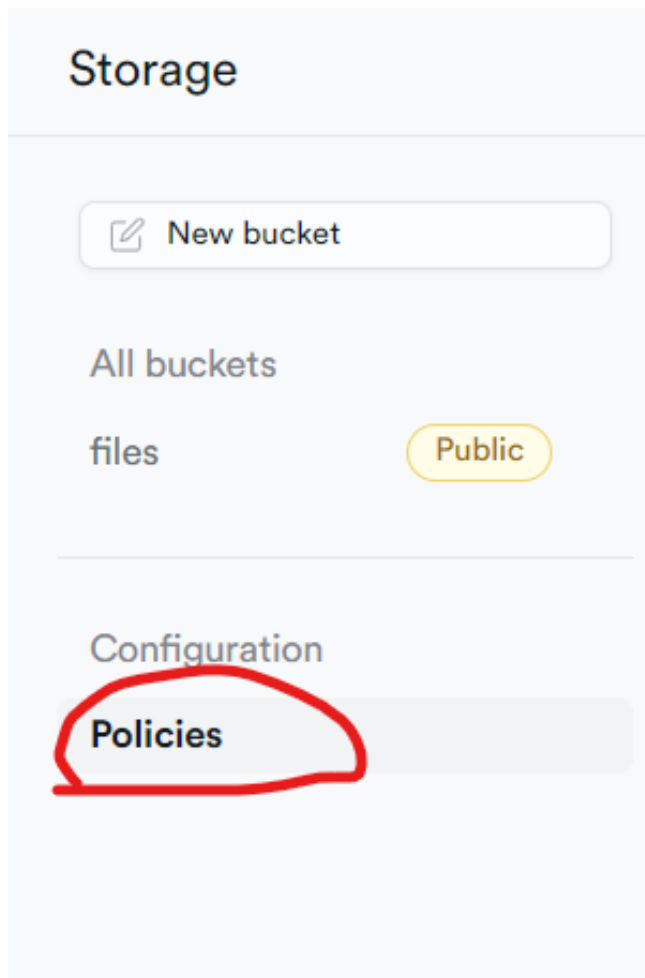
☒ Public bucket
Anyone can read any object without any authorization

 **Public buckets are not protected**
Users can read objects in public buckets without any authorization.
Row level security (RLS) policies are still required for other operations such as object uploads and deletes.

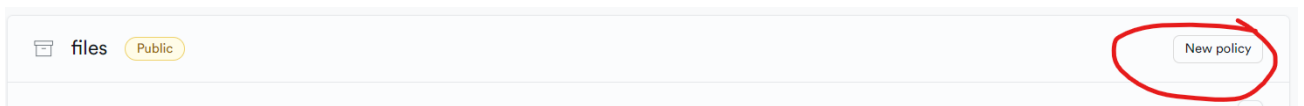
Additional configuration 

Cancel Save

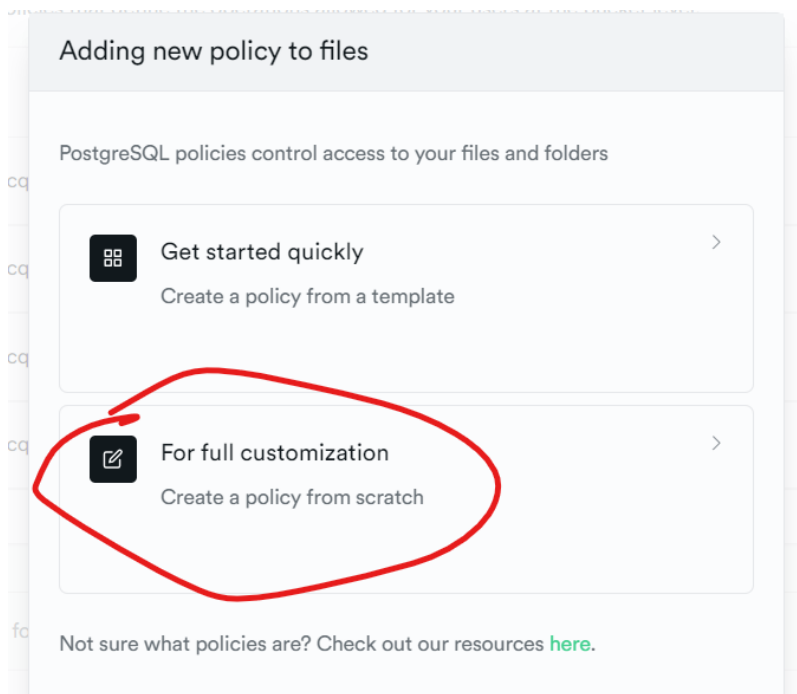
- Finally we have to setup the row level securities for the storage bucket.
- To do that click on the Policies tab in sidebar



- Then click on the new policy button



- Next choose For full customization



- Then give a name for the policy (any name) select all the methods as following image and Click on the review button

Adding new policy to files

Policy name

A descriptive name for your policy

name here9/50

Allowed operation

Based on the operations you have selected, you can use the highlighted functions in the [client library](#).

☒ SELECT

☒ INSERT

☒ UPDATE

☒ DELETE

upload

download

list

update

move

copy

remove

createSignedUrl

createSignedUrls

getPublicUrl

Target roles

Apply policy to the selected roles

Defaults to all (public) roles if none selected

Policy definition

Provide a SQL conditional expression that returns a boolean.

```
1  bucket_id = 'files'
```

- Then you are done with all the steps in supabase dashboard

How the admin UI works?

- Navigate to localhost:3000/admin to access admin page
- Log in with the give username and password in .env file
- When admin click on select files button it will allow to select a file
- After select a File it will upload to supabase bucket storage
- We are using bucket storage for storing files instead of sever, because in some case if server restarts the fill will be lost
- It will show the uploading status under table and it will show relevant file name No of 1KB batches and text size, status and actions
- After upload, it will automatically update the table as status to uploaded.
- Next admin can ingest the file to vector storage in supabase.
- vector storage is where the chatbot getting context for generate the answers.
- After ingesting it will show the status as ingested.
- Admin have other two options.

delete from dataset - this will delete the data from vector storage but will keep in supabase file storage bucket

Delete permanently - this will delete both data and file from vector storage and file storage too.