# Generate Embeddings

Generate text embeddings using Edge Functions.

This guide will walk you through how to generate high quality text embeddings in Edge Functions using its built-in AI inference API, so no external API is required.

## Build the Edge Function

Let's build an Edge Function that will accept an input string and generate an embedding for it. Edge Functions are server-side TypeScript HTTP endpoints that run on-demand closest to your users.

### 1 Set up Supabase locally

Make sure you have the latest version of the Supabase CLI installed.

Initialize Supabase in the root directory of your app and start your local stack.

```
supabase init
supabase start
```

### 2 Create Edge Function

Create an Edge Function that we will use to generate embeddings. We'll call this `embed` (you can name this anything you like).

This will create a new TypeScript file called `index.ts` under `./supabase/functions/embed`.

```
supabase functions new embed
```

### 3 Setup Inference Session

Let's create a new inference session to be used in the lifetime of this function. Multiple requests can use the same inference session.

Currently, only the `gte-small` (https://huggingface.co/Supabase/gte-small) text embedding model is supported in Supabase's Edge Runtime.

./supabase/functions/embed/index.ts

```
const session = new Supabase.ai.Session('gte-small');
```

### 4 Implement request handler

Modify our request handler to accept an `input` string from the POST request JSON body.

Then generate the embedding by calling `session.run(input)`.

./supabase/functions/embed/index.ts

```
Deno.serve(async (req) => {
  // Extract input string from
  const { input } = await req.
  // Generate the embedding fr
  const embedding = await sess
    mean_pool: true,
    normalize: true,
  });
  // Return the embedding
  return new Response(
    JSON.stringify({ embedding
    { headers: { 'Content-Type
  );
});
```

Note the two options we pass to `session.run()`:

`mean_pool`: The first option sets `pooling` to `mean`. Pooling referes to how token-level embedding representations are compressed into a single sentence embedding that reflects the meaning of the entire sentence. Average pooling is the most common type of pooling for sentence embeddings.

`normalize`: The second option tells to normalize the embedding vector so that it can be used with distance measures like dot product. A normalized vector means its length (magnitude) is 1 - also referred to as a unit vector. A vector is normalized by dividing each element by the vector's length (magnitude), which maintains its direction but changes its length to 1.

5 Test it!

To test the Edge Function, first start a local functions server.

Then in a new shell, create an HTTP request using cURL and pass in your input in the JSON body.

```
curl --request POST 'http://localhost:54321/functions/v1/embed' \
  --header 'Authorization: Bearer ANON_KEY' \
  --header 'Content-Type: application/json' \
  --data '{ "input": "hello world" }'
```

Be sure to replace `ANON_KEY` with your project's anonymous key. You can get this key by running `supabase status`.

## Next steps

Learn more about embedding concepts

Store your embeddings in a database

Edit this page on GitHub

Need some help? Contact support

Latest product updates? See Changelog

Something's not right? Check system status



—

Contributing

Author Styleguide

Open Source

SupaSquad

Privacy Settings