

## ACESSO DE BANCO DE DADOS DO SUPABASE

```
// Setup type definitions for built-in Supabase Runtime APIs
import "jsr:@supabase/functions-js/edge-runtime.d.ts";
import { createClient } from 'jsr:@supabase/supabase-js@2'

Deno.serve(async (req) => {
  try {
    const supabase = createClient(
      Deno.env.get('SUPABASE_URL') ?? "",
      Deno.env.get('SUPABASE_ANON_KEY') ?? "",
      { global: { headers: { Authorization: req.headers.get('Authorization')! } } }
    )

    // TODO: Change the table_name to your table
    const { data, error } = await supabase.from('table_name').select('*')

    if (error) {
      throw error
    }

    return new Response(JSON.stringify({ data }), {
      headers: { 'Content-Type': 'application/json' },
      status: 200,
    })
  } catch (err) {
    return new Response(JSON.stringify({ message: err?.message ?? err }), {
      headers: { 'Content-Type': 'application/json' },
      status: 500
    })
  }
})
```



}}

Esse código foi escrito em TypeScript e roda no Deno (runtime usado pelo Supabase Edge Functions). Ele cria uma função serverless que recebe requisições HTTP, conecta ao banco de dados do Supabase usando as variáveis de ambiente do projeto e executa um `SELECT *` em uma tabela específica. O resultado é retornado em formato JSON com status 200 em caso de sucesso, ou uma mensagem de erro com status 500 se algo falhar.

## UPLOAD DE ARMAZENAMENTO SUPABASE

```
// Setup type definitions for built-in Supabase Runtime APIs
import "jsr:@supabase/functions-js/edge-runtime.d.ts";
import { createClient } from 'jsr:@supabase/supabase-js@2'
import { randomUUID } from 'node:crypto'

const supabase = createClient(
  Deno.env.get('SUPABASE_URL') ?? "",
  Deno.env.get('SUPABASE_ANON_KEY') ?? ""
)

Deno.serve(async (req) => {
  const formData = await req.formData()
  const file = formData.get('file')

  // TODO: update your-bucket to the bucket you want to write files
  const { data, error } = await supabase
    .storage
    .from('your-bucket')
    .upload(
      `${file.name}-${randomUUID()}`,
```



(61) 3035-8900

SIGA Área Especial para Indústria nº 02  
Setor Leste - Gama - DF  
CEP: 72445-020

```
    file,  
    { contentType: file.type }  
  )  
  if (error) throw error  
  return new Response(  
    JSON.stringify({ data }),  
    { headers: { 'Content-Type': 'application/json' } }  
  )  
})
```

**Esse código em TypeScript, rodando no Deno (Supabase Edge Function), cria um endpoint que recebe arquivos via HTTP e faz upload deles para um bucket no Supabase Storage, salvando com nomes únicos.**

### **API integrada do Node**

```
// Setup type definitions for built-in Supabase Runtime APIs  
import "jsr:@supabase/functions-js/edge-runtime.d.ts";  
import { randomBytes } from "node:crypto";  
import { createServer } from "node:http";  
import process from "node:process";  
  
const generateRandomString = (length) => {  
  const buffer = randomBytes(length);  
  return buffer.toString('hex');  
};
```



```
const randomString = generateRandomString(10);  
console.log(randomString);
```

```
const server = createServer((req, res) => {  
  const message = Hello;  
  res.end(message);  
});
```

```
server.listen(9999);
```

**Esse código em JavaScript rodando no Node.js gera uma string aleatória no console e sobe um servidor HTTP simples na porta 9999, que responde com "Hello" a qualquer requisição.**

