

✓ Setup: Use Supabase Only for File Storage in Node.js

📦 Step 1: Install Supabase Client

```
npm install @supabase/supabase-js
```

🔑 Step 2: Create Supabase Config File

Create `supabaseClient.js`:

```
// supabaseClient.js
const { createClient } = require("@supabase/supabase-js");

const SUPABASE_URL = "https://your-project-id.supabase.co";
const SUPABASE_ANON_KEY = "your-anon-key"; // Only use anon key here

const supabase = createClient(SUPABASE_URL, SUPABASE_ANON_KEY);

module.exports = supabase;
```

📁 Step 3: Upload a File to Supabase Storage

Assume you want to upload a file sent from an HTML form or via API. Here's an example using Express:

```
// uploadFile.js
const supabase = require("./supabaseClient");
const fs = require("fs");

async function uploadFile(localFilePath, fileNameInBucket) {
  const fileBuffer = fs.readFileSync(localFilePath);

  const { data, error } = await supabase.storage
    .from("your-bucket-name")
    .upload(fileNameInBucket, fileBuffer, {
      contentType: "image/jpeg", // adjust for your file type
      upsert: true, // overwrites if file with same name exists
    });

  if (error) {
    console.error("Upload failed:", error.message);
  } else {
    console.log("File uploaded successfully:", data);
  }
}
```

Use like this:

```
uploadFile("./uploads/avatar.jpg", "avatars/user123.jpg");
```

Step 4: Get Public URL of the Uploaded File

```
const supabase = require("./supabaseClient");

function getPublicURL(filePath) {
  const { data } = supabase.storage
    .from("your-bucket-name")
    .getPublicUrl(filePath);

  return data.publicUrl;
}

// Example usage
const url = getPublicURL("avatars/user123.jpg");
console.log("Public URL:", url);
```

Step 5: Create Bucket (Only Once)

If you haven't created the storage bucket yet:

1. Go to [Supabase dashboard](#)
2. Select your project → **Storage** → **New Bucket**
3. Give it a name (e.g. **avatars**)
4. Set **Public** or **Private** access depending on your use case

Notes

- If storing **sensitive files**, keep the bucket **private** and use `createSignedUrl()` to generate time-limited access links.
- You can also delete or list files using:

```
await supabase.storage
  .from("your-bucket-name")
  .remove(["avatars/user123.jpg"]);
```

Let me know if you need:

- A full Express route to handle file upload from browser
- Signed URLs for private buckets
- Integration of MongoDB record with Supabase file URL

Happy to help!