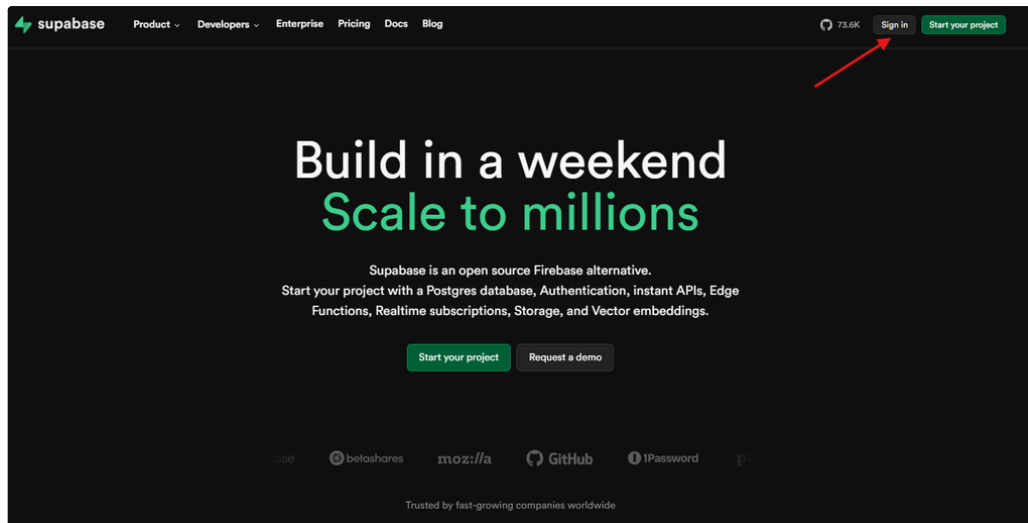


Supabase

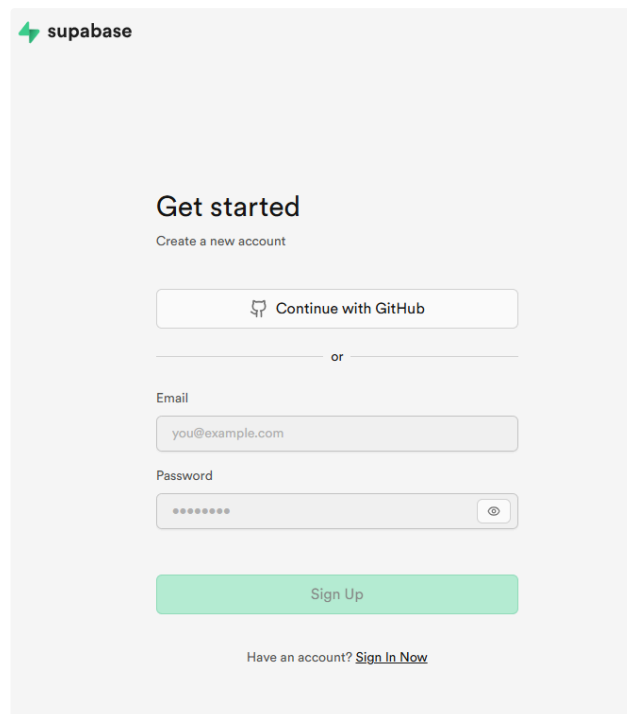
Vediamo ora come iscriversi e creare uno spazio storage in supabase.

Andiamo sul sito:

[Supabase | The Open Source Firebase Alternative](#)



e registriamoci

A screenshot of the Supabase 'Get started' registration form. The form is titled 'Get started' and 'Create a new account'. It features a 'Continue with GitHub' button. Below this, there is a section for email and password registration. The email field contains 'you@example.com'. The password field is masked with dots and has a toggle for visibility. A green 'Sign Up' button is at the bottom. At the very bottom, there is a link: 'Have an account? [Sign In Now](#)'.

Una volta cliccato per la registrazione ci chiederà di fare un check della nostra email, accettiamo e proseguiamo



Confirm your email address to start building with Supabase

You can start building with Supabase right away once you've confirmed that latuamail@mail.it is your email. Click the button below to confirm.

Confirm Email Address

If you didn't request for this, you can safely ignore this email.

Un volta confermato verremo indirizzati nella pagina di creazione di una nuova organizzazione

Create a new organization

This is your organization within Supabase.
For example, you can use the name of your company or department.

Name
What's the name of your company or team?

Type of organization
What would best describe your organization?

Plan
Pricing [?](#)

[Cancel](#) [Create organization](#)

You can rename your organization later

Diamo un nome a piacimento e lasciamo il resto invariato , clicchiamo quindi su [create organization](#).

Ci chiederà poi di creare un nuovo progetto, lasciamo tutto invariato [tranne il nome del progetto e la password che possiamo inventare](#).

Create a new project

Your project will have its own dedicated instance and full Postgres database.
An API will be set up so you can easily interact with your new database.

Organization

Project name **Inventati**

Database Password [Copy](#)

This password is strong. [Generate a password](#)

Region
Select the region closest to your users for the best performance.

[SECURITY OPTIONS >](#)

[Cancel](#) [Create new project](#)

You can rename your project later

procediamo con ["create new project"](#), **ATTENZIONE CREARE UNA PASSWORD FORTE** , potremmo decidere di farla generare ma ricordiamoci **[ASSOLUTAMENTE DI COPIARLA E SALVARCELA ALTRIMENTI VERRA' PERSA](#)**

Entreremo nella pagina all'interno della quale abbiamo le chiavi API che utilizzeremo nel nostro progetto

Progetto 1

Security IssuesProject StatusConnect

Welcome to your new project

Your project has been deployed on its own instance, with its own API all set up and ready to use.

Get started by building out your database

Start building your app by creating tables and inserting data. Our Table Editor makes Postgres as easy to use as a spreadsheet, but there's also our SQL Editor if you need something more.

Table Editor

SQL editor

About Database

id

name

iso2

1	Angola	AO
2	Timor-Leste	TL
3	Serbia	RS
4	Bahamas	BS
5	Chile	CL
6	Denmark	DK

```

1 create table countri
2 id bigint generate
3 name text
4 iso2 text not null
5 iso3 text
6 local_name text
7 continent continen
8 );

```

Explore our other products

Supabase provides all the backend features you need to build a product. You can use it completely, or just the features you need.

Authentication

A complete user management system that works without any additional tools.

Explore AuthAbout Auth

Storage

Store, organize, and serve any file types of any size from multiple buckets.

Explore StorageAbout Storage

Edge Functions

Write custom code without deploying or scaling servers, with fast deploy times and low latency.

Explore FunctionsAbout Functions

Realtime

Listen to your PostgreSQL database in realtime via websockets.

Explore RealtimeAbout Realtime

Connecting to your new project

Interact with your database through the [Supabase client libraries](#) with your API keys.

More information about your project's keys can be found in your project's API settings.

View API settings

About APIs

Project API

Your API is secured behind an API gateway which requires an API Key for every request. You can use the parameters below to use Supabase client libraries.

Project URL

chiave

Copy

A RESTful endpoint for querying and managing your database.

API Key

anonpublic

Copy

This key is safe to use in a browser if you have enabled Row Level Security (RLS) for your tables and configured policies. You may also use the service key which can be found [here](#) to bypass RLS.

Javascript

Dart

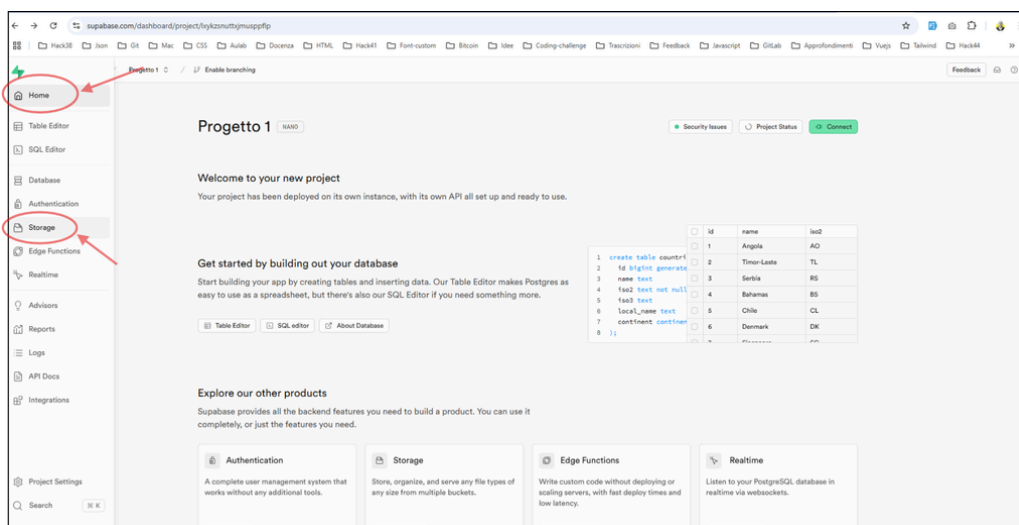
```

import { createClient } from '@supabase/supabase-js'

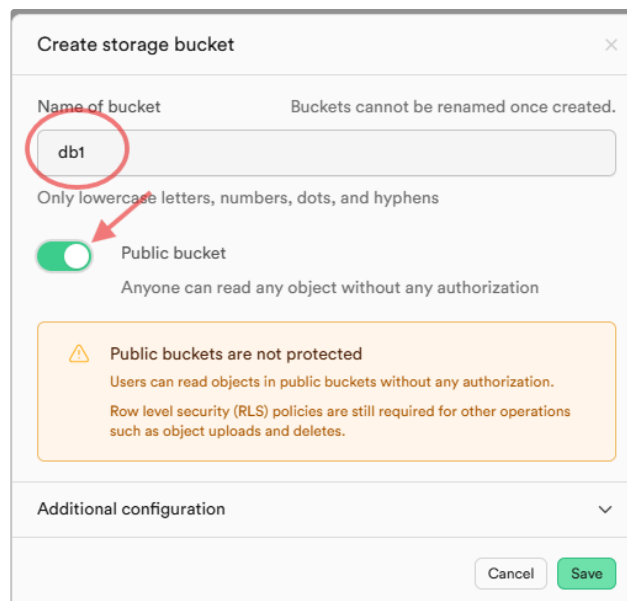
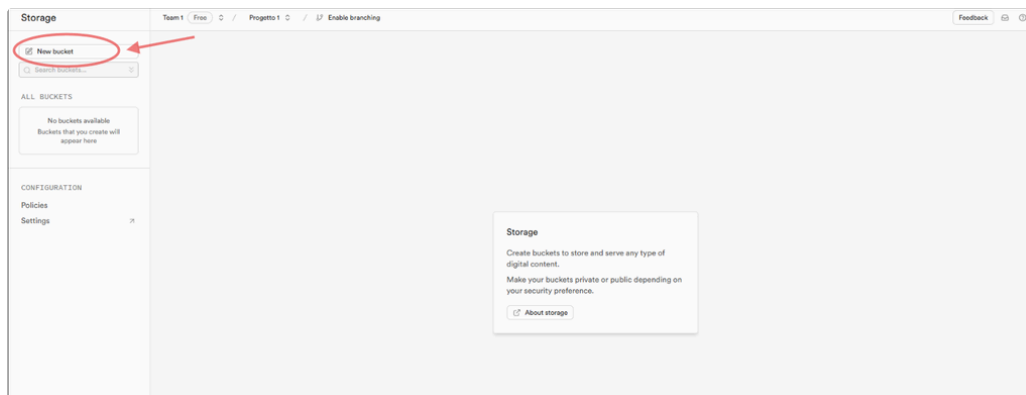
const supabaseUrl = 'https://lxykzsnuttxjmsppflp.supabase.co'
const supabaseKey = process.env.SUPABASE_KEY
const supabase = createClient(supabaseUrl, supabaseKey)

```

Ora andiamo sull'icona della casetta dove ci verrà mostrato il menu e clicchiamo quindi nella sezione "storage"



e una volta in storage creiamo un nuovo bucket

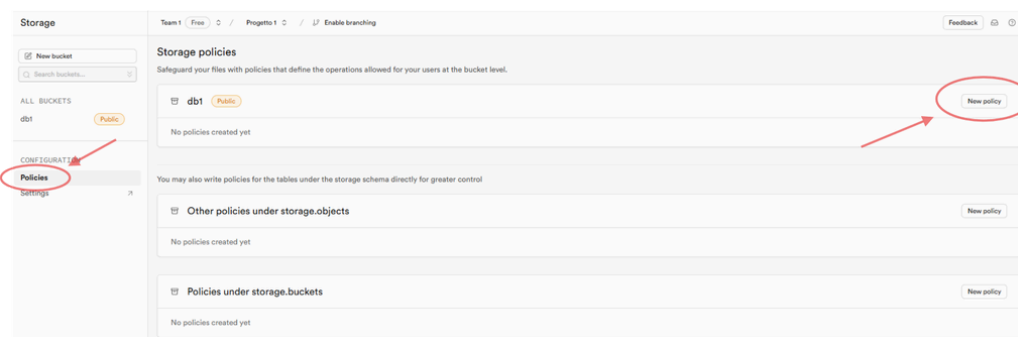


The 'Create storage bucket' dialog box is shown. The 'Name of bucket' field contains 'db1' and is circled in red. Below it, the 'Public bucket' toggle is turned on and also circled in red. A red arrow points from the toggle to the 'Public bucket' label. A warning box states: 'Public buckets are not protected. 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.' At the bottom, there are 'Cancel' and 'Save' buttons.

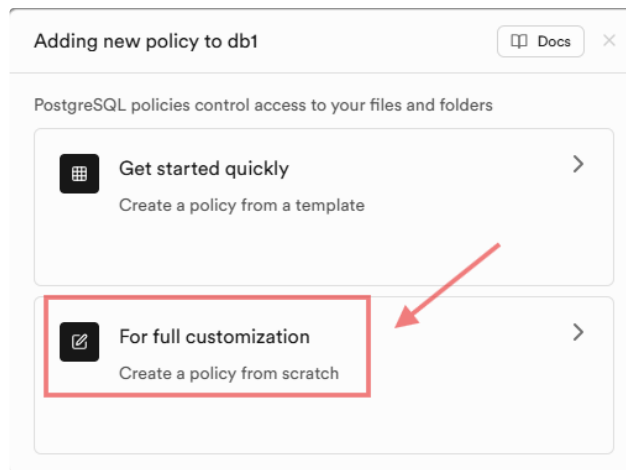
Diamo un nome a scelta e rendiamolo pubblico, fatto questo clicchiamo su save.

Abbiamo quindi creato il nostro bucket all'interno del quale inseriremo le immagini

Dobbiamo sbloccare i permessi di manipolazione quindi andiamo su policies

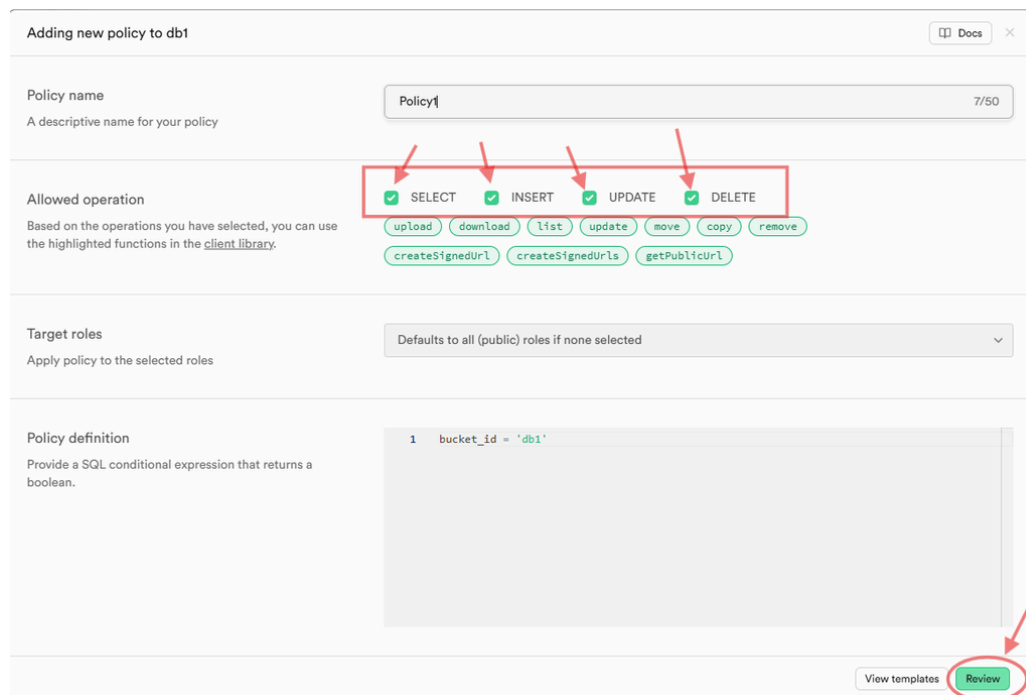


E creiamo una nuova policy sul nostro bucket



Clicchiamo su **“For full customization”**

E selezioniamo **select, insert, update, delete** e diamo un nome alla policy che inventeremo



Clicchiamo su **“review”**

Reviewing policies to be created for db1

Docs

These are the SQL statements that will be used to create your policies. The suffix appended to the end of your policy name (`[hashString]_[number]`) just functions as a unique identifier for each of your policies.

Add policy for the SELECT operation under the policy "Policy1"

```
1 CREATE POLICY "Policy1 24j7_0" ON storage.objects FOR SELECT TO public USING (bucket_id = 'db1');
```

Add policy for the INSERT operation under the policy "Policy1"

```
1 CREATE POLICY "Policy1 24j7_1" ON storage.objects FOR INSERT TO public WITH CHECK (bucket_id = 'db1');
```

Back to edit

Save policy

Arrivati in questa schermata clicchiamo su save *(non vi preoccupate se le query dovessero risultare differenti)*

Dovremo avere poi questo risultato

Storage

Team 1

Free

Progetto 1

Enable branching

Feedback

New bucket

Search buckets...

ALL BUCKETS

db1 Public

CONFIGURATION

Policies

Settings

Storage policies

Safeguard your files with policies that define the operations allowed for your users at the bucket level.

db1 Public

SELECT Policy1 24j7_0

INSERT Policy1 24j7_1

UPDATE Policy1 24j7_2

DELETE Policy1 24j7_3

4 policies in db1

You may also write policies for the tables under the storage schema directly for greater control

Other policies under storage.objects

No policies created yet

Policies under storage.buckets

No policies created yet

Abbiamo finito, in questo storage vedremo comparire le nostre immagini ma non avremo bisogno di tenerlo aperto tranne che per casi straordinari, se in futuro dovessimo aver bisogno di recuperare le chiavi da inserire nel nostro progetto andiamo sempre sulla home cioè il simbolo della casetta.