

GABRIELE ROMANATO

Come usare Supabase in Node.js

Supabase è un'ottima alternativa open-source a Firebase, basata su PostgreSQL. Offre autenticazione, database in tempo reale, storage e molto altro. In questa guida vedremo come integrarlo in un'applicazione Node.js.

Installazione

Per iniziare, installa il client Supabase utilizzando npm:

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

Configurazione

Una volta installato il pacchetto, configura il client Supabase con il tuo URL e la chiave API. Puoi trovare queste informazioni nella dashboard di Supabase.

```
const { createClient } = require('@supabase/supabase-js');

const SUPABASE_URL = 'https://your-project-url.supabase.co';
const SUPABASE_KEY = 'your-anon-key';

const supabase = createClient(SUPABASE_URL, SUPABASE_KEY);
```

Operazioni di base

Inserire dati

Per inserire dati in una tabella, utilizza il metodo insert:

```
async function insertData() {
  const { data, error } = await supabase
    .from('users')
    .insert([
      { name: 'Mario Rossi', email: 'mario@example.com' }
    ]);

  if (error) console.error(error);
  else console.log(data);
}

insertData();
```

Recuperare dati

Per ottenere dati da una tabella, usa il metodo select:

```
async function getData() {
  const { data, error } = await supabase
    .from('users')
    .select();

  if (error) console.error(error);
  else console.log(data);
}

getData();
```

Aggiornare dati

Puoi aggiornare i dati con il metodo update:

```
async function updateData() {
  const { data, error } = await supabase
    .from('users')
    .update({ name: 'Luigi Bianchi' })
    .eq('email', 'mario@example.com');

  if (error) console.error(error);
  else console.log(data);
}

updateData();
```

Eliminare dati

Per eliminare un record, usa il metodo delete:

```
async function deleteData() {
  const { data, error } = await supabase
    .from('users')
    .delete()
    .eq('email', 'mario@example.com');

  if (error) console.error(error);
  else console.log(data);
}

deleteData();
```

Autenticazione

Supabase offre anche autenticazione con email e password. Ecco come registrare un utente:

```
async function signUp() {
  const { user, error } = await supabase.auth.signUp({
    email: 'test@example.com',
    password: 'password123'
  });

  if (error) console.error(error);
  else console.log(user);
}

signUp();
```

E per effettuare il login:

```
async function signIn() {
  const { user, error } = await supabase.auth.signInWithPassword({
    email: 'test@example.com',
    password: 'password123'
  });

  if (error) console.error(error);
  else console.log(user);
}

signIn();
```

Conclusione

Supabase semplifica notevolmente la gestione del database e dell'autenticazione in un'applicazione Node.js. Grazie alla sua API intuitiva, puoi integrare facilmente funzionalità avanzate senza dover gestire server complessi.

Applicazioni Correlate



-

Node.js Placeholder Image

Applicazione per la generazione con Node.js di immagini segnaposto.

Docker Docker Compose Node.js JavaScript ExpressJS



-

Node.js URL Shortener

Implementazione in Node.js di un sistema per l'abbreviazione degli URL.

DockerDocker ComposeNode.jsJavaScriptExpressJSMongoDB

A large yellow square containing the letters 'JS' in a bold, black, sans-serif font, centered within the square.

-

JavaScript App Hash Change

Applicazione che sfrutta gli hash degli URL per gestire contenuto dinamico in JavaScript.
DockerDocker ComposeNode.jsJavaScriptExpressJSMysql