

GABRIELE ROMANATO

Menu

Introduzione a PostgreSQL in Node.js

PostgreSQL è uno dei database relazionali più potenti e popolari disponibili, mentre Node.js è un ambiente di runtime JavaScript molto utilizzato per lo sviluppo di applicazioni server-side. Combinate insieme, queste tecnologie offrono una solida base per la creazione di applicazioni web robuste e performanti. In questo articolo, esploreremo come utilizzare PostgreSQL in Node.js, fornendo una guida dettagliata per chi inizia.

Prima di iniziare a utilizzare PostgreSQL con Node.js, è necessario installare il database sul proprio sistema. Le istruzioni per l'installazione possono variare a seconda del sistema operativo, ma sono generalmente disponibili sul sito ufficiale di PostgreSQL. Assicurarsi di prendere nota del nome utente, della password e del nome del database che si desidera utilizzare, poiché ne avremo bisogno più avanti.

Una volta installato PostgreSQL, possiamo procedere con la configurazione di un progetto Node.js. Assicuriamoci di aver già installato Node.js sul nostro sistema. Creiamo una nuova cartella per il nostro progetto e inizializziamolo utilizzando npm (Node Package Manager):

```
mkdir my-postgres-project
cd my-postgres-project
npm init -y
```

Successivamente, installiamo il modulo pg che ci permetterà di interagire con il database PostgreSQL:

```
npm install pg --save
```

Ora che abbiamo configurato il nostro progetto, possiamo iniziare a scrivere codice per connetterci al database PostgreSQL. Creiamo un file app.js e iniziamo con il seguente codice:

```
const { Pool } = require('pg');

const pool = new Pool({
  user: 'il_tuo_nome_utente',
  host: 'localhost',
  database: 'il_tuo_nome_database',
  password: 'la_tua_password',
  port: 5432 // Porta di default di PostgreSQL
});
```

```
pool.query('SELECT NOW()', (err, res) => {
  if (err) {
    console.error('Errore nella connessione al database:', err);
  } else {
    console.log('Connessione al database PostgreSQL avvenuta con successo:',
res.rows[0].now);
  }
  pool.end(); // Chiudiamo la connessione
});
```

Questo codice crea una connessione al database PostgreSQL utilizzando il modulo pg. Assicuriamoci di sostituire i valori per user, host, database e password con le credenziali del nostro database.

Una volta stabilita la connessione al database, possiamo eseguire query SQL per interagire con i dati. Modifichiamo il nostro file app.js per eseguire una query di esempio:

```
const { Pool } = require('pg');

const pool = new Pool({
  user: 'il_tuo_nome_utente',
  host: 'localhost',
  database: 'il_tuo_nome_database',
  password: 'la_tua_password',
  port: 5432
});

pool.query('SELECT * FROM users', (err, res) => {
  if (err) {
    console.error('Errore nell'esecuzione della query:', err);
  } else {
    console.log('Risultati della query:', res.rows);
  }
  pool.end();
});
```

Assicurarsi di sostituire `SELECT * FROM users` con la propria query SQL e adattarla allo schema del proprio database.

Conclusioni

In questo articolo, abbiamo visto come utilizzare PostgreSQL in Node.js per connettersi a un database, eseguire query SQL e interagire con i dati. PostgreSQL offre una vasta gamma di funzionalità avanzate per la gestione dei dati, mentre Node.js fornisce un ambiente robusto per lo sviluppo di applicazioni server-side. Combinando queste tecnologie, è possibile creare applicazioni web potenti e scalabili. Continua ad esplorare le funzionalità di entrambe le tecnologie e approfondire la tua conoscenza per sfruttarle al massimo nelle tue applicazioni.

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.
Docker Docker Compose Node.js JavaScript Express JS MongoDB



-

JavaScript App Hash Change

Applicazione che sfrutta gli hash degli URL per gestire contenuto dinamico in JavaScript.
Docker Docker Compose Node.js JavaScript Express JS MySQL