

Connecting with Node.js

This guide explains how to establish a connection between a Node.js application and a PostgreSQL database using the `pg` package. It walks through the necessary setup, configuration, and execution of a simple SQL query.

Variables

Certain parameters must be provided to establish a successful connection to a PostgreSQL database. Below is a breakdown of each required variable, its purpose, and where to find it. Here’s what each variable represents:

Variable	Description	Purpose
<code>USER</code>	PostgreSQL username, from the Elestio service overview page	Identifies the database user who has permission to access the PostgreSQL database.
<code>PASSWORD</code>	PostgreSQL password, from the Elestio service overview page	The authentication key required for the specified <code>USER</code> to access the database
<code>HOST</code>	Hostname for PostgreSQL connection, from the Elestio service overview page	The address of the server hosting the PostgreSQL database.
<code>PORT</code>	Port for PostgreSQL connection, from the Elestio service overview page	The network port is used to connect to PostgreSQL. The default port is <code>5432</code> .
<code>DATABASE</code>	Database Name for PostgreSQL connection, from the Elestio service overview page	The name of the database being accessed. A PostgreSQL instance can contain multiple databases.

These values can usually be found in the Elestio service overview details as shown in the image below, make sure to take a copy of these details and add it to the code moving ahead.



postgresql-2p7j1

PostgreSQL

Running

Open terminal

Delete service

Clone this service

Overview

Tools

Backups

Metrics

Monitoring

Logs

Audit

Security

Alerts

Termination protection

Disabled. VM can be powered off and terminated.

Protection deactivated



Database Admin

Display your database credentials

Hide DB Credentials

Host postgresql-2p7j1-u7774.vm.elestio.app



Port 25432



User postgres



Password *****

Show password



CLI PGPASSWORD=***** psql --host=postgresql-2p7j1-u7774.vm.elestio.app --port=25432 --username=postgres

Show password



Prerequisites

• Install Node.js and NPM

- Check if Node.js is installed by running:

```
node -v
```

- If not installed, download it from nodejs.org and install.
- Verify npm installation:

```
npm -v
```

• Install the `pg` Package

The `pg` package enables Node.js applications to interact with PostgreSQL. Install it using:

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

Code

Once all prerequisites are set up, create a new file named `pg.js` and add the following code:

```
const pg = require("pg");

// Database connection configuration
```

```
const config = {
  user: "USER",
  password: "PASSWORD",
  host: "HOST",
  port: "PORT",
  database: "DATABASE",
};

// Create a new PostgreSQL client
const client = new pg.Client(config);

// Connect to the database
client.connect((err) => {
  if (err) {
    console.error("Connection failed:", err);
    return;
  }
  console.log("Connected to PostgreSQL");

  // Run a test query to check the PostgreSQL version
  client.query("SELECT VERSION()", [], (err, result) => {
    if (err) {
      console.error("Query execution failed:", err);
      client.end();
      return;
    }

    console.log("PostgreSQL Version:", result.rows[0]);

    // Close the database connection
    client.end((err) => {
      if (err) console.error("Error closing connection:", err);
    });
  });
});
```

To execute the script, open the terminal or command prompt and navigate to the directory where `pg.js`. Once in the correct directory, run the script with the command

```
node pg.js
```

If the connection is successful, the terminal will display output similar to:

```
Connected to PostgreSQL
PostgreSQL Version: {
  version: 'PostgreSQL 16.8 (Debian 16.8-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc
(Debian 12.2.0-14) 12.2.0, 64-bit'
}
```

Revision #2

Created 20 March 2025 06:34:47 by kaiwalya

Updated 20 March 2025 08:37:40 by kaiwalya