

Connecting with Node.js

This guide explains how to establish a connection between a Node.js application and a KeyDB database using the [redis](#) package. It walks through the necessary setup, configuration, and execution of a simple KeyDB command.

Variables

To successfully connect to a KeyDB instance, you'll need to provide the following parameters. These can typically be found on the Elestio service overview page.

Variable	Description	Purpose
HOST	KeyDB hostname (from Elestio service overview)	The address of the server hosting your KeyDB instance.
PORT	KeyDB port (from Elestio service overview)	The port used for the KeyDB connection. The default KeyDB port is 6379.
PASSWORD	KeyDB password (from Elestio service overview)	Authentication key used to connect securely to the KeyDB instance.

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.



keydb

KeyDB

Cluster

Running

Open terminal

Delete cluster

Add node

Overview

Nodes

Backups

Audit

Termination protection

Disabled. VM can be powered off and terminated.

Protection deactivated



Auto-Failover

Enabled. In case of failure, the cluster will automatically attempt to recover

Auto-Failover activated



Nodes

2 Nodes: 1 Primary, 1 Replica

Add node

Database Admin

Display your database credentials

Hide DB Credentials

Host

keydb-u7774.vm.elestio.app



Port

23647



User

root



Password

Show password



CLI

redis-cli -h keydb-u7774.vm.elestio.app -p 23647 -a *****

Show password



Prerequisites

Install Node.js and NPM

- Check if Node.js is installed by running:

```
node -v
```

- If not installed, download and install it from nodejs.org.
- Confirm npm is installed by running:

```
npm -v
```

Install the redis Package

The redis package enables communication between Node.js applications and KeyDB.

```
npm install redis --save
```

Code

Create a new file named `keydb.js` and add the following code:

```
const keydb = require("redis");

// KeyDB connection configuration
const config = {
  socket: {
    host: "HOST",
    port: PORT,
  },
  password: "PASSWORD",
};

// Create a Redis client
const client = keydb.createClient(config);

// Handle connection errors
client.on("error", (err) => {
  console.error("KeyDB connection error:", err);
});

// Connect and run a test command
(async () => {
  try {
    await client.connect();
    console.log("Connected to KeyDB");

    // Set and retrieve a test key
    await client.set("testKey", "Hello KeyDB");
    const value = await client.get("testKey");
    console.log("Retrieved value:", value);

    // Disconnect from KeyDB
    await client.disconnect();
  } catch (err) {
    console.error("Error:", err);
  }
})();
```

```
} catch (err) {  
  console.error("KeyDB operation failed:", err);  
}  
})();
```

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

```
node keydb.js
```

If the connection is successful, the output should resemble:

```
Connected to KeyDB  
Retrieved value: Hello KeyDB
```

Revision #2

Created 2025-06-26 07:40:39 UTC

Updated 2025-06-26 07:44:28 UTC