



Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



<https://www.pexels.com/de-de/foto/person-die-eine-grune-pflanze-halt-1072824/>

Database seeding in Node.js



Lars Wächter · [Follow](#)

Published in [Level Up Coding](#)

3 min read · Aug 21, 2021



Listen



Share



More

In this article I'd like to talk about database seeding using Node.js and MySQL: what it is and how to implement it. You'll notice that it's actually quite easy and straightforward. For this tutorial, I chose MySQL as the database system but this

procedure should also work with any other one. In this case, there might be some small modifications necessary regarding the SQL query execution.

Let's start with an explanation of database seeding. [Source](#)

Database seeding is the initial seeding of a database with data. Seeding a database is a process in which an initial set of data is provided to a database when it is being installed. It is especially useful when we want to populate the database with data we want to develop in future.

So our goal is to “feed” the database with dummy data on its initialization. This can be very helpful especially during the development process or for onboarding new employees that run the development environment (database) locally on their machine.

First of all, we create a new SQL script that includes the queries for inserting the dummy data. This script will be later executed using Node.js.

The seeding script: `./db/seeding.sql`



```
/* Insert admin account */  
INSERT INTO user (email, firstname, lastname, password)  
VALUES ('admin@email.com', 'John', 'Doe', ?);
```

SQL snippet

This SQL command inserts a new data set into the `user` table. The `?` is a placeholder that gets replaced with a variable, the initial password, when calling the script using Node.js. You can add more queries here of course.

Make sure that the database tables were created before running the script otherwise it'll fail. Most ORMs take care of that.

Next, we write the Node.js script, which is required to establish a database connection and execute the SQL snippet we just created.

The Node.js script: `./db/index.js`

```
const mysql = require("mysql2")
const fs = require("fs")
const bcrypt = require("bcryptjs")

// Load .env variables
require("dotenv").config()

// Read SQL seed query
const seedQuery = fs.readFileSync("db/seed.sql", {
  encoding: "utf-8",
})

// Connect to database
const connection = mysql.createConnection({
  host: process.env.DB_HOST,
  user: process.env.DB_USERNAME,
  password: process.env.DB_PASSWORD,
  database: process.env.DB_NAME,
  multipleStatements: true, // IMPORTANT
})

connection.connect()

// Generate random password for initial admin user
const psw = Math.random()
  .toString(36)
  .substring(2)
const hash = bcrypt.hashSync(psw, 10)

console.log("Running SQL seed...")

// Run seed query
connection.query(seedQuery, [hash], err => {
  if (err) {
    throw err
  }

  console.log("SQL seed completed! Password for initial admin account: " + psw)
  connection.end()
})
```

Node.js script

What happens here?

1. Load the environment variables using `dotenv`
2. Read the SQL snippet
3. Connect to the database (MySQL)
4. Generate a random password (hashed)
5. Execute the SQL snippet

At the end, the admin password is logged to the console if everything worked fine. Instead of using `dotenv` you can also use `fs.readFileSync` to load your database credentials from any other file.

If you're not using MySQL you simply have to modify the database connection setup. I'm sure there's an npm package for your database system that can handle this.

Last but not least let's extend the `package.json` scripts to simplify the execution of the Node script.

```
{
  "scripts": {
    // ...
    "seed": "node db/index.js"
    // ...
  }
}
```

Now you can run the database seeding with a single command from the terminal: `npm run seed`. That's it! Your database is filled with dummy data and you can focus on the important stuff.

There's also a GitHub repository available including an example application that makes use of database seeding. Have a look.

GitHub - larswaechter/expressjs-api: An example Node.js REST-API built with TypeScript & Express.js

This repository is a dummy Node.js REST-API built with TypeScript and Express. You might want to use it as codebase for...

github.com

[Nodejs](#)[Programming](#)[JavaScript](#)[Web Development](#)[Database](#)[Follow](#)

Written by Lars Wächter

120 Followers · Writer for Level Up Coding

cs student | software developer

More from Lars Wächter and Level Up Coding




Lars Wächter in Level Up Coding

Path aliases with TypeScript in Node.js


Some time ago I included path aliases in my TypeScript Node.js projects. Since they make the code look much cleaner in my opinion I like...

3 min read · Jan 17, 2020

 349  5



 Fareed Khan in Level Up Coding

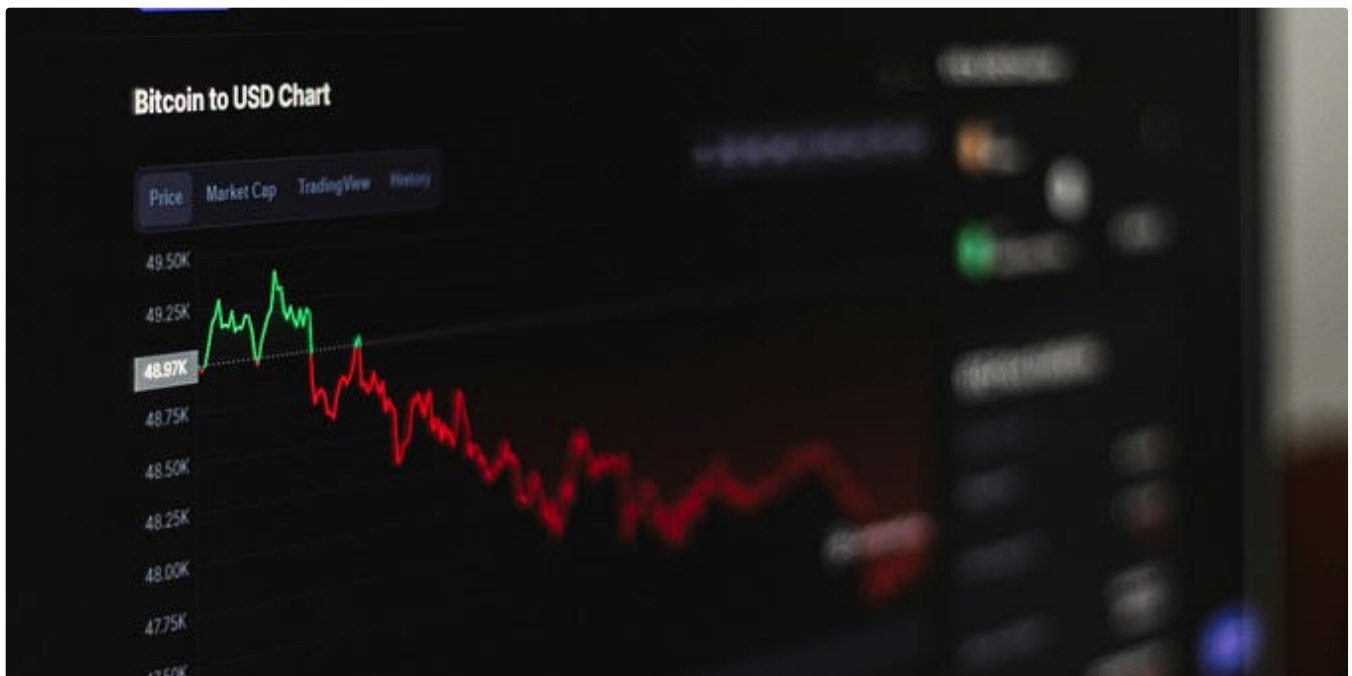
Building a Million-Parameter LLM from Scratch Using Python


A Step-by-Step Guide to Replicating LLaMA Architecture

25 min read · Dec 7

 1.6K  19



 Nikhil Adithyan in Level Up Coding

An Algo Trading Strategy which made +8,371%: A Python Case Study

Backtesting of a simple breakout trading strategy with APIs and Python

7 min read · Nov 28




714



20



 Lars Wächter in Level Up Coding

Improving Minimax performance

10 techniques for improving the Minimax algorithm's performance.

8 min read · Nov 30, 2021



148



See all from Lars Wächter

See all from Level Up Coding

Recommended from Medium



Icegreen

A simple guide to setting up a project with Node.js, Typescript, Postgres and TypeORM

The purpose of this article is to teach you how to start your project from scratch with Node.js, express, typescript, postgres and typeORM.

7 min read · Aug 14

▼ Relational databases


15 Min

Connect your database

Using Prisma Migrate

Install Prisma Client

Querying the database

 Kemal Taufik Fikri

Learn Prisma

First of all we just need to go to prisma website to get the tutorial. The website link is below:

9 min read · Jul 6

Lists



General Coding Knowledge

20 stories · 722 saves



Stories to Help You Grow as a Software Developer

19 stories · 662 saves



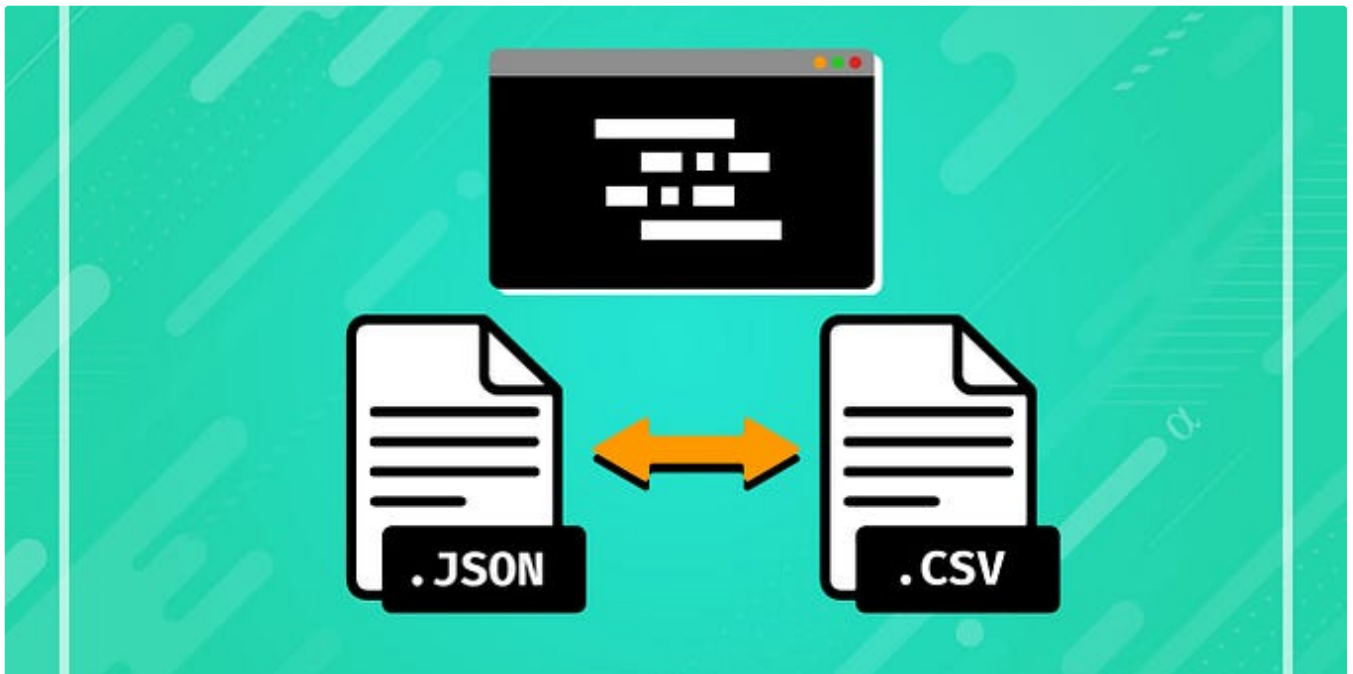
Coding & Development

11 stories · 346 saves



ChatGPT

23 stories · 348 saves



 Jekanola Adedipupo

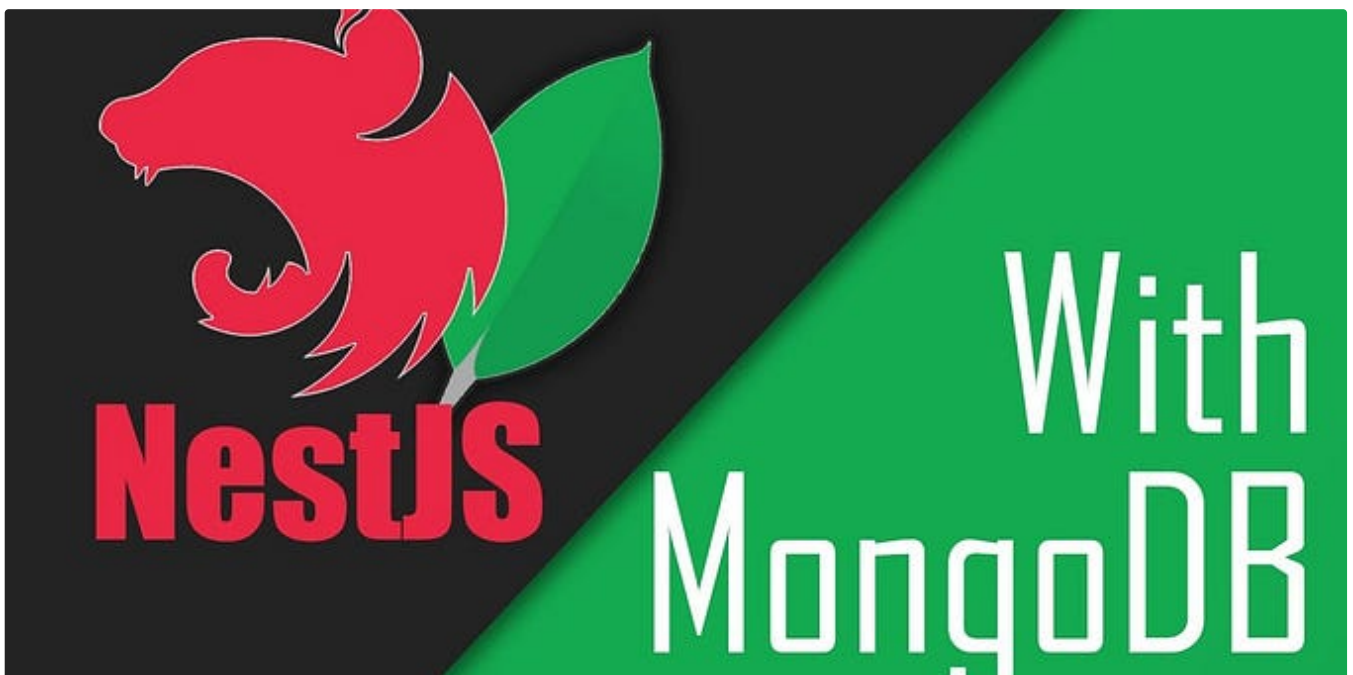
Using Node.js to write from JSON to CSV

In the world of modern web development, data exchange and transformation are essential tasks. One common scenario is converting data from...

4 min read · Aug 17

 8  2




 Bobby Gagnon

MongoDB Migrations with NestJS

NestJS and MongoDB are both great tools. One issue I'm hoping to solve is to add a bit more automation to MongoDB migrations.

5 min read · Jul 22



 cengkuru michael

How to Update Node.js to Any Version: A Step-by-Step Guide

Introduction

2 min read · Aug 3





mongoDB®



Shemeera Fonseka

User Management Application using Node, Express and MongoDB

In this article, you would know about how to download and install MongoDB to windows, and how to implement application using Node.js and...

11 min read · Jul 27



23



1



See more recommendations