

NPM Packages Build A Complete Backend



Express.js – Web Framework

Express.js is a fast, unopinionated, and minimalist web framework for Node.js. It simplifies the development of server-side applications.

```
const express = require('express');
const app = express();

app.get('/', (req, res) => {
  res.send('Hello World!');
});

app.listen(3000, () => {
  console.log
('Server is running on port 3000');
});
```



Mongoose – MongoDB ODM

Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It provides a schema-based solution to model your data.

```
const mongoose = require('mongoose');

mongoose.connect('mongodb://localhost/mydatabase',
{ useNewUrlParser: true, useUnifiedTopology: true });

const userSchema = new mongoose.Schema({
  name: String,
  age: Number
});

const User = mongoose.model('User', userSchema);

const newUser = new User({ name: 'John', age: 30 });
newUser.save().then(() => console.log('User saved!'));
```



Dotenv – Environment Variables

Dotenv loads environment variables from a .env file into process.env. It keeps your configuration separate from your codebase.

```
require('dotenv').config();  
  
const dbPassword = process.env.DB_PASSWORD;  
  
console.log(`Database Password: ${dbPassword}`);
```



Bcrypt – Password Hashing

Bcrypt is a library to help you hash passwords securely. It's essential for implementing user authentication.

```
const bcrypt = require('bcrypt');
const saltRounds = 10;
const myPlaintextPassword = 's0/\\P4$$w0rd';

bcrypt.hash(myPlaintextPassword, saltRounds,
  (err, hash) => {
    if (err) throw err;
    console.log(`Hashed Password: ${hash}`);
  });
```



Jsonwebtoken – JWT for Authentication

Jsonwebtoken is a library to generate and verify JSON Web Tokens (JWT). It's commonly used for secure authentication.

```
const jwt = require('jsonwebtoken');

const user = { id: 1, name: 'John' };
const token = jwt.sign(user, 'secret_key',
  { expiresIn: '1h' });

console.log(`JWT: ${token}`);

jwt.verify(token, 'secret_key', (err, decoded)
=> {
  if (err) throw err;
  console.log(`Decoded JWT: ${decoded}`);
});
```



Nodemailer – Sending Emails

Nodemailer is a module for Node.js applications to send emails easily. It's useful for sending notifications, confirmations, and other emails.

```
const nodemailer = require('nodemailer');

const transporter = nodemailer.createTransport({
  service: 'gmail',
  auth: {
    user: 'youremail@gmail.com',
    pass: 'yourpassword'
  }
});

const mailOptions = {
  from: 'youremail@gmail.com',
  to: 'recipient@example.com',
  subject: 'Hello',
  text: 'Hello World!'
};

transporter.sendMail(mailOptions, (error, info) => {
  if (error) {
    return console.log(error);
  }
  console.log(`Email sent: ${info.response}`);
});
```

