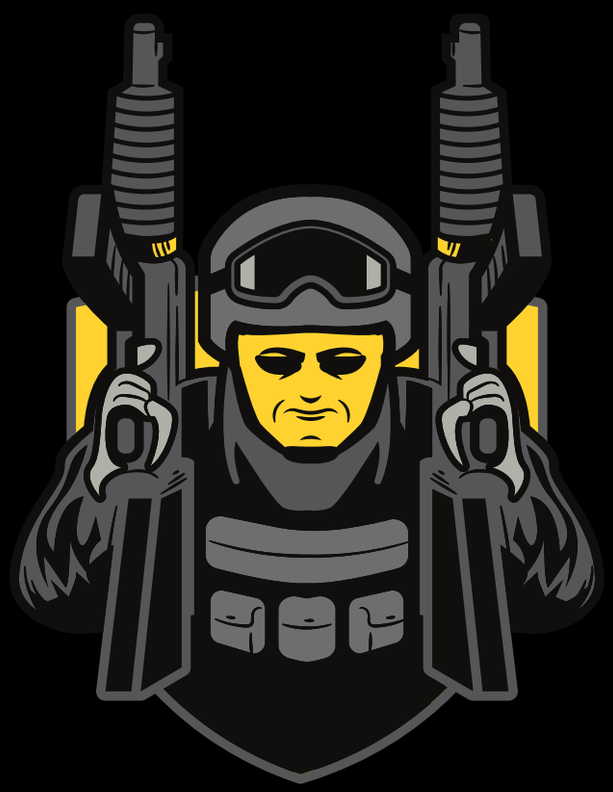


TOP 30 NPM PACKAGES TO ENHANCE YOUR APP DEVELOPMENT

@somanathbarik

EXPRESS

WHEN TO USE EXPRESS:

✅ To leverage vast ecosystem and plugins
✅ Manage complex web portal
✅ Seamless integration with asynchronous programming patterns
✅ Determining a minimalist framework without any restrictions

@somanathbarik

SOCKET.IO

WHEN TO USE SOCKET.IO:

✅ Develop Node js application with quick data updates and real-time communication

✅ Need to gather and analyze data to generate insights into the users

✅ Want to monitor and track system metrics, and server statuses, which alert instantly about specific interactions

✅ Built application on a gaming platform, IoT, customer support, and sports update

@somanathbarik

GRAPHQL

WHEN TO USE GRAPHQL:

- ✅ To minimize bandwidth usage and develop applications with limited network resources
- ✅ Abstract specific language data based on user preferences
- ✅ To fetch complex data structure and interactions into a single request
- ✅ Streamline integration with third-party APIs

@somanathbarik

HELMET

WHEN TO USE HELMET:

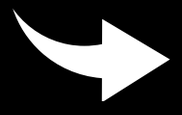
✅ If your applications deal with sensitive data, such as finance or personal details

✅ Eliminate unauthorized access by exposing APIs accessed by external users

✅ Your web application requires user authentication to protect against common security vulnerabilities

✅ If the header enforces HTTP requests and secures all communication between client and server

@somanathbarik



MULTER

WHEN TO USE MULTER:

✅ To utilize different types of file uploads like single, multiple, and upload within required data structures

✅ When your app requires support file uploads like images, documents, videos, and other types of files

✅ Developing user-generated content applications, such as content management systems, social media platforms, and blogging medium

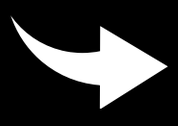
✅ To integrate with multiple storage solutions like cloud storage services, such as Google Cloud Storage or AWS S3.

@somanathbarik

LODASH

WHEN TO USE LODASH:

- ✅ Simplify the JavaScript development process
- ✅ Handling complex data structures efficiently
- ✅ Suitable for cross-platform applications
- ✅ Choosing a functional programming style



@somanathbarik

MOMENT

WHEN TO USE MOMENT:

- ✅ For operating in different times and zones
- ✅ Managing time-related calculations, such as adding or subtracting time intervals
- ✅ Lightweight and efficient solutions for dates
- ✅ Simplify and streamline your date and time-related code

@somanathbarik

ASYNC

WHEN TO USE ASYNC:

✅ To rectify errors in your asynchronous code consistently and centralized

✅ Enhance application performance and responsiveness with Async parallel execution of numerous tasks concurrently

✅ Improve web application user experience with long-run asynchronous operations

✅ Want to complete tasks within a specific timeframe for your asynchronous operations

@somanathbarik

BLUEBIRD

WHEN TO USE BLUEBIRD:

✅ To deal with asynchronous workflows that include various operations with dependencies or sequential performance

✅ Suitable for workflow automation systems or tools as it simplifies the implementation of workflow

✅ To leverage its promise-based approach for handling database operations, update, query, and delete data

✅ Creating middleware or plugins for libraries or frameworks that implement promise

@somanathbarik

UNDERSCORE

WHEN TO USE UNDERSCORE:

- ✅ Utilize to manipulate collection or data, such as mapping, filtering, or decreasing arrays
- ✅ To implement objects, such as iteration, and particular properties for application
- ✅ Compatibility with backend and frontend development
- ✅ Enhance development efficiency by decreasing the need to write custom functions and use well-tested methods

@somanathbarik

WINSTON

WHEN TO USE WINSTON:

- ✅ Reliable logging solution for your Node js application
- ✅ Quickly identify and analyze errors in the Node application
- ✅ Monitor or track user behaviors to enhance user experience
- ✅ To improve debugging and troubleshooting processes by generating detailed logs

@somanathbarik

AXIOS

WHEN TO USE AXIOS:

✅ For making HTTP requests from your applications
✅ Asynchronous operations and simplifying error handling
✅ Implementing authorization and authentication mechanisms
✅ Eliminating pending requests or sending various requests simultaneously

@somanathbarik

MINIFY

WHEN TO USE MINIFY:

✅ Working on front-end development projects
✅ Minified version to control and deployment processes
✅ Improve loading speed and website performance
✅ Minimize the size of assets

@somanathbarik

REDUX

WHEN TO USE REDUX:

- ✅ For sharing multiple components to access the same data
- ✅ Maintaining consistent and predictable application state
- ✅ Leveraging extensive resources and community support
- ✅ Integrating middleware for handling asynchronous actions or caching

@somanathbarik

PASSPORT

WHEN TO USE PASSPORT:

✅ Need to implement token-based authentication utilizing JSON Web Tokens (JWT)
✅ A lightweight and flexible authentication middleware that supports various authentication strategies
✅ Secure different routes or endpoints of your app based on user permissions
✅ Compatible with numerous web frameworks and seamless integration into existing Node js application

@somanathbarik

BABEL

WHEN TO USE BABEL:

- ✅ To write modern JavaScript code employing the latest ECMAScript features
- ✅ Customizable toolchain to configure particular transformations and plugins for your JavaScript code
- ✅ Want a compiler that is compatible with older systems or frameworks
- ✅ Seamless integration into the existing build process to automate code transformation and browser-compatible code

@somanathbarik

JEST

WHEN TO USE JEST:

✅ Write code in different languages along with the integration of multiple tools
✅ Easy to use and Beginner-appropriate testing frameworks
✅ Working on open-source projects that demand robust testing
✅ Leverage reliable test coverage to manage complex scenarios

@somanathbarik

LINTER

WHEN TO USE LINTER:

- ✅ To catch potential errors and bugs during the early development process
- ✅ Streamline code reviews by automating the identification of primary code problems
- ✅ Minimizing the risk of vulnerabilities by securing coding practices
- ✅ Enhance code portability by detecting compatibility issues through browsers, platforms, and environments

@somanathbarik

MONGOOSE

WHEN TO USE MONGOOSE:

✅ To establish relationships between MongoDB collections, enabling to define of associations and related datail

✅ Looking for a scalable and flexible solution for working with MongoDB

✅ Developing Node js application that can be easily adapted to modify your data model

✅ Extensive plugins or extensions to expand the functionality of MongoDB database

@somanathbarik

SEQUELIZE

WHEN TO USE SEQUELIZE:

- ✅ When you need to perform CRUD (Create, Read, Update, Delete) operations on your database
- ✅ Develop a real-life application that efficiently requires data handling and synchronization with the database
- ✅ Manage database migrations and handle changes to the database schema over time
- ✅ To support data validation and ensure data integrity to handle the database

@somanathbarik

NODEMON

WHEN TO USE NODEMON:

✅ Automate solution to monitor and identify file changes in your codebase or when the server restarts

✅ Easy to set up and integrate into the development environment without significant code change or configuration

✅ To develop web applications that demand continuous testing and validation in the development process

✅ When you want to ensure an uninterrupted and smooth development process

@somanathbarik

REQUEST

WHEN TO USE REQUEST:

- ✅ To abstract data from external APIs or web services to integrate with your apps
- ✅ For synchronizing data between numerous systems or databases
- ✅ Interact with RESTful APIs to submit data as Request simplifies the HTTP requests
- ✅ Want to monitor external services for bugs or issues as Request helps to perform monitoring

@somanathbarik

CHEERIO

WHEN TO USE CHEERIO:

✅ To parse HTML templates and update or modify them with data

✅ To monitor and collect data like pricing or product information from competitor’s website

✅ Simplifies the organizing and analyzing data process from HTML sources

✅ Test and validate HTML functionality and structure during development or maintenance process

@somanathbarik

PM2

WHEN TO USE PM2:

✅ To generate detailed reports and metrics for optimizing application performance

✅ When you want to build flexible application using Node Js Microservices architecture with centralized process management

✅ Automatic code reload upon changes, making development and testing workflows efficient

✅ Apps looking for granular control of log management and rotations to ensure the use of disk space

@somanathbarik

MORGAN

WHEN TO USE MORGAN:

✅ Log incoming HTTP requests and responses time for your Node js applications
✅ Identify potential errors of bottlenecks during the initial development process
✅ Implement request rate limiting or access control based on request patterns
✅ Easy-to-use solutions for logging HTTP requests in your Node apps

@somanathbarik

SHARP

WHEN TO USE SHARP:

✅ When you need to generate thumbnails or preview images using special dimensions
✅ Convert images between different formats, such as converting PNG into JPEG
✅ Improve the visual quality of the image by implementing sharpening algorithms
✅ Enhance your application’s performance and loading times by employing progressive image-loading techniques

@somanathbarik

CHAI

WHEN TO USE CHAI:

✅ To boost testing process efficiency with Chai’s extensive feature set

✅ Building applications for industries such as finance, e-commerce, and healthcare, where accurate information is crucial

✅ To identify failures or errors during the early development stage through the testing process

✅ Use an automation test for an application requiring continuous integration and deployment processes

@somanathbarik

JOI

WHEN TO USE JOI:

- ✅ Enforce data validation rules and inputs for API request payload
- ✅ Enhance quality and reliability of your application through data validation
- ✅ For flexible and customizable JavaScript library
- ✅ Verify configuration files or environment variables

@somanathbarik

NODEMAILER

WHEN TO USE NODEMAILER:

- ✅ For using simple plaintext and HTML email formats
- ✅ Integrating with third-party email services like Outlook or Gmail to send email
- ✅ Ideal for sending emails in bulk and running email marketing campaigns
- ✅ Sending different language transactional emails and supporting international email delivery systems

@somanathbarik

PUPPETEER

WHEN TO USE PUPPETEER:

- ✅ To automate end-to-end testing of your web application
- ✅ Enhance user interface by simulating user interaction and navigating application flows
- ✅ Analyze and optimize web page performance with techniques like measuring load times
- ✅ Identify visual changes on web pages by rendering layouts

@somanathbarik

Thank you!

@somanathbarik