

# Full Stack JavaScript Developer

*Glossary - Backend Development with Node.js*



---

<b>Access Modifier</b>	Used in classes to declare how a property or method can be accessed from the application
<b>Behavior Driven Development</b>	A development style built on Test Driven Development where the focus is user interaction and stakeholders
<b>Comparison Test</b>	A type of test that compares strings, numbers, objects, or arrays
<b>Compile</b>	Convert code from one language to another language with a different level of abstraction. Ex. Java to Machine Code.
<b>Container</b>	A tool used to encapsulate the entirety of an application including runtime, libraries, and files to run independently of global configurations on an operating system
<b>DELETE</b>	HTTP request that removes data from the server
<b>dependencies</b>	Dependencies used in both development and production
<b>devDependencies</b>	Dependencies that are only necessary for development
<b>Duck typing</b>	A programming paradigm where if two or more structures (functions, interfaces, objects) have the same properties, they can be used interchangeably regardless of any type declarations
<b>Endpoint</b>	An endpoint is the URL of the REST API with the method that gets, adds to, or modifies the data of an API in some way
<b>Enumerated type</b>	A set of constants that are automatically indexed and can be called by their name or index
<b>Explicit</b>	When type is declared by the developer
<b>Generic</b>	A way to write a function that is reusable across different types.
<b>GET</b>	HTTP request that retrieves data from the server
<b>GraphQL</b>	A query language used for working with APIs

<b>Idempotency</b>	When making multiple identical requests to the API produce the same results each time
<b>Implicit Typing</b>	When type is inferred by the compiler
<b>Interface</b>	Used as a blueprint to declare the shape of something reusable such as functions, objects, and classes
<b>Interpreted Language</b>	The language is read by a runtime and executed on the spot and errors are found on execution
<b>Libuv</b>	A library written in C that provides multithreading to Node.js and allows for heavy processing
<b>Lodash</b>	A popular library for performing utility functions for things like arrays and numbers
<b>Microservice</b>	A piece of a larger application. In Microservice applications, the application is broken down into encapsulated microservices that can be maintained individually
<b>Middleware</b>	Functionality that runs between a request to the server and the response from the server
<b>Monolithic Architecture</b>	A unified architecture where there is no separation between services or components of the application
<b>Multithreaded language</b>	The ability of a language to run 2 or more concurrent tasks on what are known as threads. A CPU has 2 threads. If you are on an 8 core machine, you could run 16 concurrent tasks.
<b>Node Package Manager (npm)</b>	A tool for managing project dependencies via command line as well as a website hosting more than 1 million third-party packages that can be used for your project
<b>Node Version Manager (nvm)</b>	A software package that allows a user to run a different version of Node.js for each project on the same machine
<b>Numerical Matchers Test</b>	A test of numerical values within a specified range of the expected value
<b>package-lock.json</b>	A JSON file that contains all of the information for the dependencies of an app's installed modules
<b>package.json</b>	A JSON file that acts as a manifest for your project including name, author, version, description, license, dependencies, scripts, etc.

<b>PATCH</b>	HTTP request that updates data on the server
<b>POST</b>	HTTP request that sends data to the server
<b>Prettier</b>	A code formatting package that can be integrated into projects to improve code consistency and readability
<b>PUT</b>	HTTP request that replaces data on the server
<b>Query parameter</b>	A key-value pair added to the end of a URL to supply data through the URL that can be used by the application
<b>REPL (Read, Evaluate, Print, Loop)</b>	An environment used for running programming languages
<b>REST</b>	Representational State Transfer, a method for working with data
<b>RESTful</b>	Term to describe an API that is implemented following REST principles. Often used interchangeably with REST (e.g. REST/RESTful APIs).
<b>Route</b>	The name or path used to access endpoints
<b>Router</b>	Middleware that directs your application to different routes
<b>Runtime</b>	The final phase in an application where the code is run
<b>Runtime</b>	A runtime is an environment that is used to interpret and run a programming language
<b>Self-contained application</b>	The use of self-contained implies the application receives data from no external sources
<b>Spec</b>	An individual test
<b>Stakeholder</b>	Any individual or group of individuals with an invested interest in something
<b>Strong-Typed</b>	Types are important and require declarations. Steps need to be taken to change variables to different types
<b>Suite</b>	A group of related tests
<b>Superset</b>	A language that extends or builds on top of another language or standard
<b>Test-Driven Development</b>	A development style where tests are written before development

<b>Transpile</b>	Convert a language to a similarly abstracted language. Ex. ES6 to ES5
<b>Truthiness</b>	When a conditional proves to be truth-like such as the boolean <b>true</b> or a condition being <b>true</b> , or a value not equal to <b>0</b> , <b>NaN</b> , <b>undefined</b> , <b>null</b> , or empty.
<b>Tuple</b>	A data type of an array with a set number of values where all value types are known
<b>Weak-Typed</b>	Types aren't declared and in some cases mixed and concatenated
<b>Websocket API</b>	A type of stateful API that allows the server to know what the user is doing and vice versa