**[Introducing Node](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction" \l "introducing_node" \o "Permalink to Introducing Node)**

[Node](https://nodejs.org/) (or more formally Node.js) is an open-source, cross-platform runtime environment that allows developers to create all kinds of server-side tools and applications in [JavaScript](https://developer.mozilla.org/en-US/docs/Glossary/JavaScript). The runtime is intended for use outside of a browser context (i.e. running directly on a computer or server OS). As such, the environment omits browser-specific JavaScript APIs and adds support for more traditional OS APIs including HTTP and file system libraries.

From a web server development perspective Node has a number of benefits:

* Great performance! Node was designed to optimize throughput and scalability in web applications and is a good solution for many common web-development problems (e.g. real-time web applications).
* Code is written in "plain old JavaScript", which means that less time is spent dealing with "context shift" between languages when you're writing both client-side and server-side code.
* JavaScript is a relatively new programming language and benefits from improvements in language design when compared to other traditional web-server languages (e.g. Python, PHP, etc.) Many other new and popular languages compile/convert into JavaScript so you can also use TypeScript, CoffeeScript, ClojureScript, Scala, LiveScript, etc.
* The node package manager (NPM) provides access to hundreds of thousands of reusable packages. It also has best-in-class dependency resolution and can also be used to automate most of the build toolchain.
* Node.js is portable. It is available on Microsoft Windows, macOS, Linux, Solaris, FreeBSD, OpenBSD, WebOS, and NonStop OS. Furthermore, it is well-supported by many web hosting providers, that often provide specific infrastructure and documentation for hosting Node sites.
* It has a very active third party ecosystem and developer community, with lots of people who are willing to help.

Other common web-development tasks are not directly supported by Node itself. If you want to add specific handling for different HTTP verbs (e.g. GET, POST, DELETE, etc.), separately handle requests at different URL paths ("routes"), serve static files, or use templates to dynamically create the response, Node won't be of much use on its own. You will either need to write the code yourself, or you can avoid reinventing the wheel and use a web framework!

[Express](https://expressjs.com/) is the most popular *Node* web framework, and is the underlying library for a number of other popular [Node web frameworks](https://expressjs.com/en/resources/frameworks.html). It provides mechanisms to:

* Write handlers for requests with different HTTP verbs at different URL paths (routes).
* Integrate with "view" rendering engines in order to generate responses by inserting data into templates.
* Set common web application settings like the port to use for connecting, and the location of templates that are used for rendering the response.
* Add additional request processing "middleware" at any point within the request handling pipeline.

While *Express* itself is fairly minimalist, developers have created compatible middleware packages to address almost any web development problem. There are libraries to work with cookies, sessions, user logins, URL parameters, POST data, security headers, and *many* more. You can find a list of middleware packages maintained by the Express team at [Express Middleware](https://expressjs.com/en/resources/middleware.html) (along with a list of some popular 3rd party packages).

## What is MongoDB?

**MongoDB** is a document-oriented NoSQL database used for high volume data storage. Instead of using tables and rows as in the traditional relational databases, MongoDB makes use of collections and documents. Documents consist of key-value pairs which are the basic unit of data in MongoDB. Collections contain sets of documents and function which is the equivalent of relational database tables. MongoDB is a database which came into light around the mid-2000s.

**MongoDB Features**

* 1. Each database contains collections which in turn contains documents. Each document can be different with a varying number of fields. The size and content of each document can be different from each other.
  2. The document structure is more in line with how developers construct their classes and objects in their respective programming languages. Developers will often say that their classes are not rows and columns but have a clear structure with key-value pairs.
  3. The rows (or documents as called in MongoDB) doesn't need to have a schema defined beforehand. Instead, the fields can be created on the fly.
  4. The data model available within MongoDB allows you to represent hierarchical relationships, to store arrays, and other more complex structures more easily.

Mongoose

Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB.

The three main advantages of using Mongoose versus native MongoDB are:

1. MongooseJS provides an abstraction layer on top of MongoDB that eliminates the need to use named collections.
2. Models in Mongoose perform the bulk of the work of establishing up default values for document properties and validating data.
3. Functions may be attached to Models in MongooseJS. This allows for seamless incorporation of new functionality.
4. Queries use function chaining rather than embedded mnemonics which result in code that is more flexible and readable, therefore more maintainable as well.

Mongoose Concepts

Mongoose uses schemas to model the data an application wishes to store and manipulate in MongoDb. This includes features such as type casting, validation, query building, and more.

The *schema* describes the attributes of the properties (aka fields) the application will manipulate. These attributes include such things as:

* Data type (e.g. String, Number, etc.).
* Whether or not it is required or optional.
* Is it’s value unique, meaning that the database is allowed to contain only one document with that value in that property.