

Node.js- MongoDB

By
Surekha Srinivasan

What is Node.js?

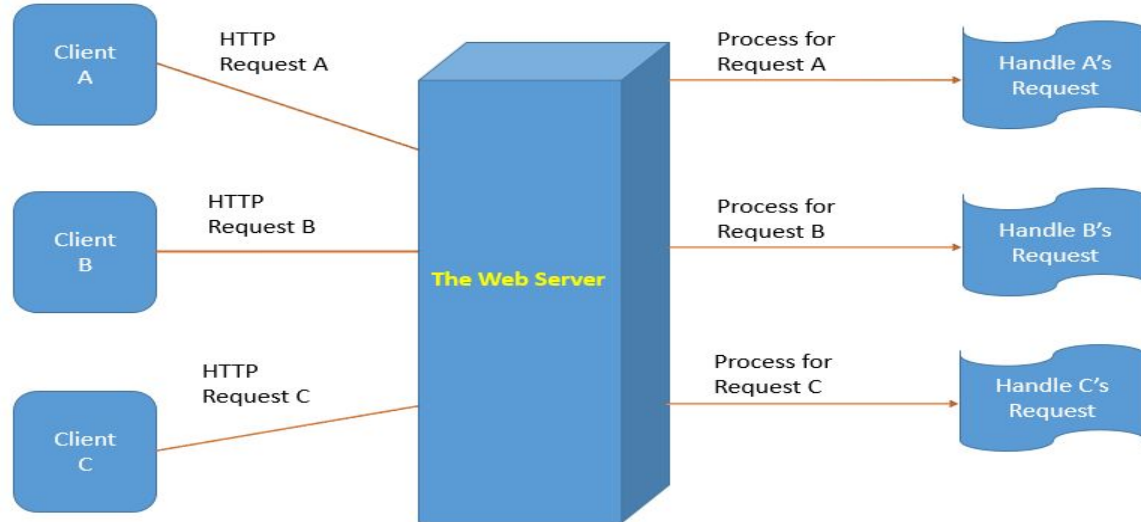
Node.js is an open source server framework.

Node.js uses JavaScript to build entire server side application.

Node.js runs on various platforms - Windows, Linux, Unix, Mac OS etc.

How do Node.js handles a file request?

A common task for a web server can be to open a file on the server and return the content to the client.



Here is how Node.js handles the file request:

Sends the task to the computer's file system.

Ready to handle the next request.

When the file system has opened and read the file, the server returns the content to the client.

Node.js eliminates the wait, and simply continues with the next request.

Node.js runs single-threaded, non-blocking, asynchronous programming, which is very memory efficient.

What Can Node.js Do?

- Node.js can generate dynamic page content
- Node.js can create, open, read, write, delete, and close files on the server
- Node.js can collect form **data**
- Node.js can **add, delete, modify data in your database**

What is a Module in Node.js?

Modules are JavaScript libraries or a set of functions you want to include in your application

How to include Modules?

To include a module, use the `require()` function with the name of the module.

For example:

```
var http = require('http');
```

Now your application has access to HTTP module and, is able to create a server.

What Is MongoDB?

```
Example: {  
  firstName: "Jamie",  
  lastName: "Munro"  
}
```

Node.js can use many different databases, and there are several approaches you can use for performing **Create, Read, Update and Delete** (CRUD) operations. One of the most popular NoSQL database is MongoDB.

MongoDB is a database that stores the data as documents.

MongoDB stores documents with a dynamic structure. These documents are saved inside a collection.

Most commonly these documents resemble a JSON-like structure as shown above:

In the above example, the document defines a `user` object. So this `user` object then would be part of a collection called `users`.

What Is Mongoose?

Mongoose is a JavaScript framework that is commonly used in a Node.js application with a MongoDB database.

Mongoose acts as a front end to MongoDB. It is an Object Data Modeling /Object Document Mapper (ODM). That is everything in Mongoose starts with a Schema. Each schema maps to a MongoDB collection and defines the shape of the documents within that collection.

Once schemas and models are defined, Mongoose have different functions that allow you to validate, save, delete, and query your data using common MongoDB functions.

How to Access MongoDB database with Node.js

Here we can write the code that will allow the Node.js application to connect to MongoDB.

Three operations will be covered: **connecting, writing, and reading** from the database.

First we need to download and install Mongodb package:

npm install mongodb (in your c9 terminal)

Now Node.js can use the mongodb module to manipulate the database.

```
var mongo = require('mongodb');
```

To Create Database

We can create MongoDB database, by creating a MongoClient object, then specify a connection URL and the name of the database we want to create.

Here we will create a database named 'School'

//Write the below code in db.js file

```
var MongoClient = require('mongodb').MongoClient;  
var url = "mongodb://surekhasrinivasan-firstworkspace-5563392/school";
```

```
MongoClient.connect(url, function(err, db) {  
  if (err) throw err;  
  console.log("Database is created!");  
  db.close();  
});
```

Save the file and run on c9 terminal as node db.js. Output = "Database is created!"

Create Collection

To create a collection we can use `createCollection()` javascript method.

Our database being 'School' we can name our collection as 'Students'.

We can use Mongo shell to create collection as shown below:

```
db.createCollection("Students")
```

To insert Documents in collection

To insert a document in the collection we can use `insertOne()` or `insertMany()` javascript methods.

We can insert document containing students first name and last name.

We can use Mongo shell to insert document as shown below:

```
db.Students.insert({"firstname":"Surekha", "lastname":"Srinivasan"})
```

To Find Documents

To find one document or find all documents in the collection we can use `findOne()` or `find()` javascript methods.

We can use Mongo shell to find document as shown below:

```
db.Students.find({"firstname":"Surekha"}) //finds a specific document
```

```
db.Students.find() //finds all the documents
```

To Delete or Remove Documents

To delete one document or delete all documents in the collection we can use `deleteOne()` or `deleteMany()` javascript methods.

We can use Mongo shell to delete or remove document as shown below:

```
db.Students.remove({"firstname":"Surekha"}) //removes a specific document
```

```
db.Students.remove({}) //removes all the documents. Be Careful!
```

To Update Documents

To update one document or many documents in the collection we can use `updateOne()` or `updateMany()` javascript methods.

We can use Mongo shell to update document as shown below:

```
db.Students.update({"firstname":"Surekha"}, {$set:{"Grade":4}})//updates a specific document
```

Thank you