TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN &

TRUYỀN THÔNG VIỆT  HÀN

**Khoa Khoa Học Máy Tính**



CLOUD COMPUTING (3)

**EXERCISE 2 :**

**LET US NODE**

Sinh viên thực hiện : **Nguyễn Lê Hữu Duy**

**Mai Thanh Trí Đông**

**Nguyễn Chí Hiếu**

**Trần Quang Hiếu**

**Nguyễn Đức Gia Khanh**

Giảng viên hướng dẫn: **Ts. Đặng Quang Hiển**

Đà Nẵng, 01  tháng 09 năm 2023

**TABLE OF CONTENTS**

## **Day 1 : Node basic concepts - Libuv, Event loop, Libev.**

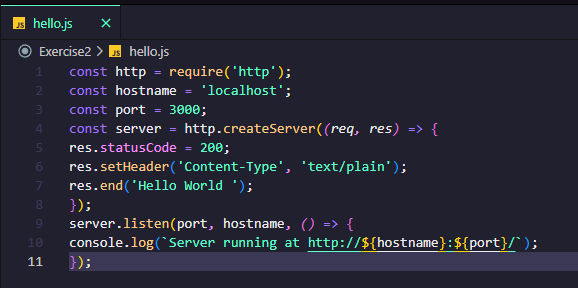
In this day , I learnt about How Node Works , V8 JavaScript engine , Event driven I/O – libuv , Working of libuv – core of Node.js so after that I learned about multi threading and single threading and Importance of event loop , How Event Loop Works. Finally , I learned about how to install Node js in My Operating system

## **Day 2 : Building a Web Server in Node.**

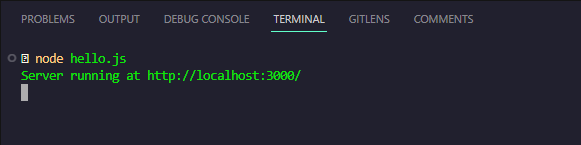
In this day , I learnt how to create a simple web server using Node using a built in http

module.

Let's build a simple Web application that prints Hello World on the browser upon user's request. I will use the native http module of Node to achieve the Web Server functionality. Here is the code:

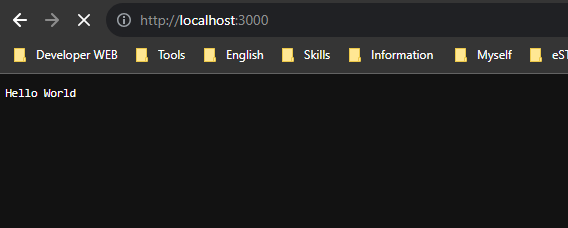


After that , I use **node hello.js** to execute my code and we can following message on the terminal



At this time , you can open my browser and type localhost:3000 and hit enter. You should

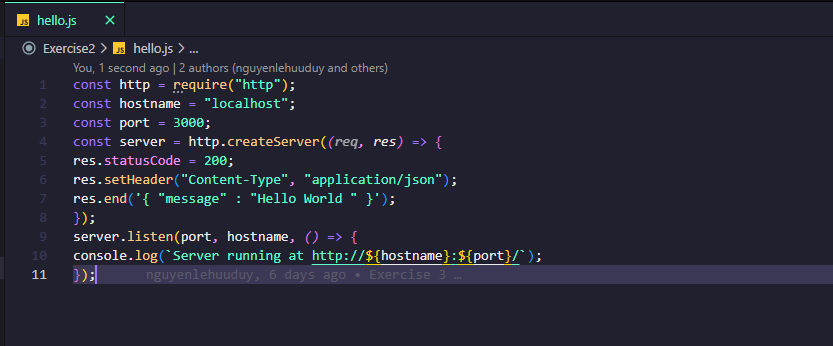
see the following message.



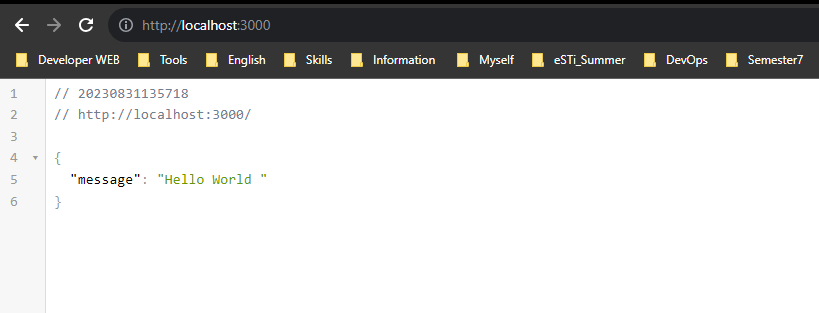
Our server responds to a simple message as a text, however, in practice we need to handle different types of responses. Let’s look over some of the common response types.

One of the most common responses that you need to handle while developing a Node application is JSON.

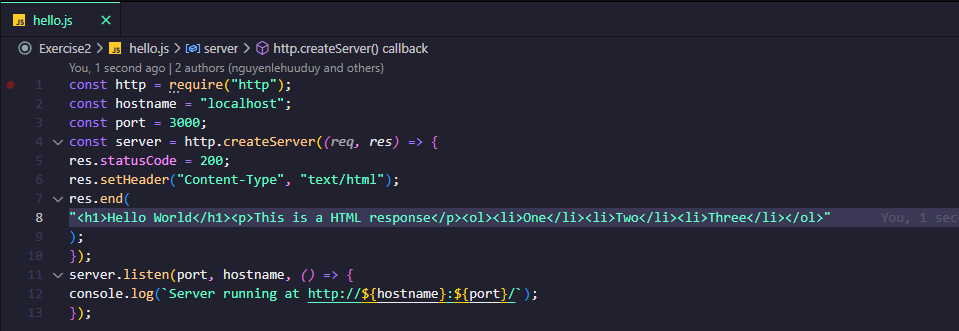
Here is how you can send JSON responses.



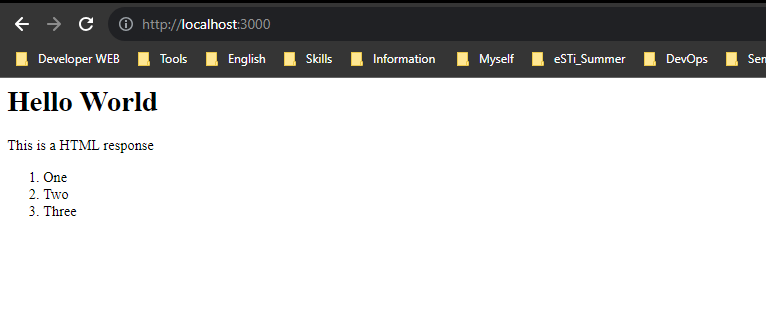
If you run this code, and navigate to the browser, you should see the following response.



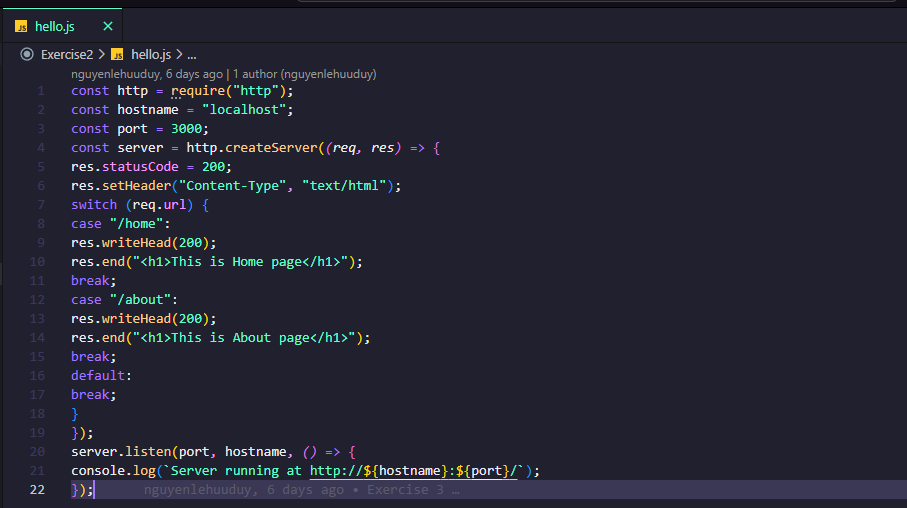
Let’s look over how to send HTML as a response.



When you run this code, you should see the following response.

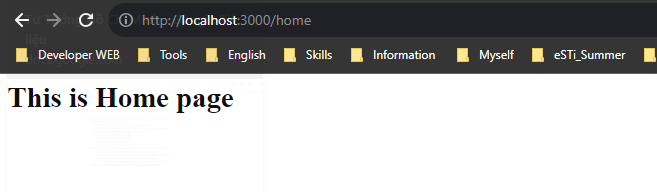


So , Let’s go ahead and create different routes to support multiple responses.

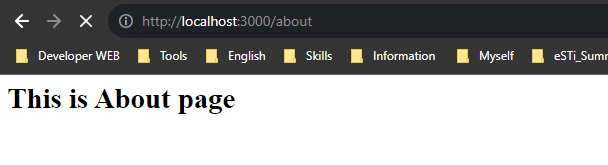


As you can see, we are using switch cases to determine different routes and on each route, we are sending different responses to each route.

Navigate your browser to localhost:3000/home to view the response.



Navigate your browser to localhost:3000/about to view the second response.

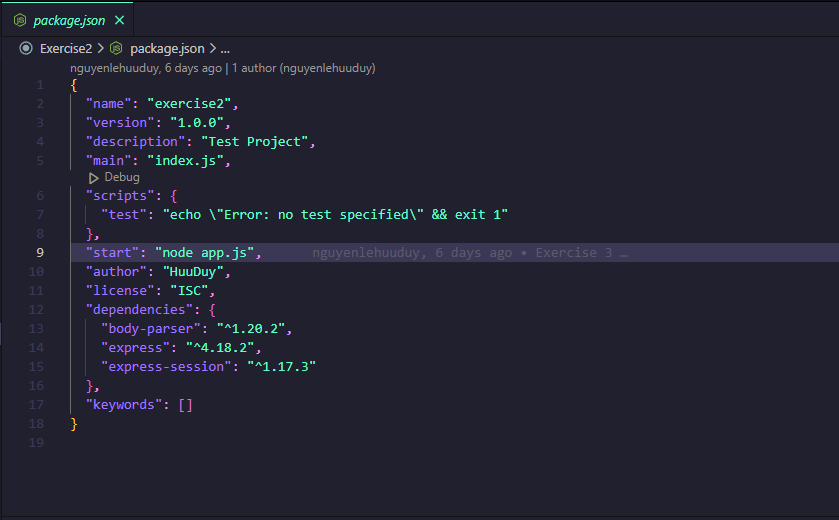


## **Day 3 :** **Node modules and NPM.**

In this day , I learnt about Node modules and how to install it .

We can install node modules using node package manager or called npm.Node modules are managed by the file called package.json. This file contains the list of the packages your project is using along with the version number etc.

A typical package.json file looks like this:



We also specify details such as project name, version, entry file of the

project in the package.json. All the dependencies that are installed using npm install command is listed in the package.json. You should always maintain the updated

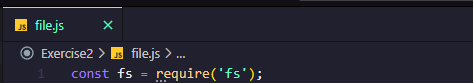
version of package.json while dealing with the packages.

## **Day 4 : File system module.**

The fs module provides an API for interacting with the file system of

your operating system. To use this module, require it in your code like

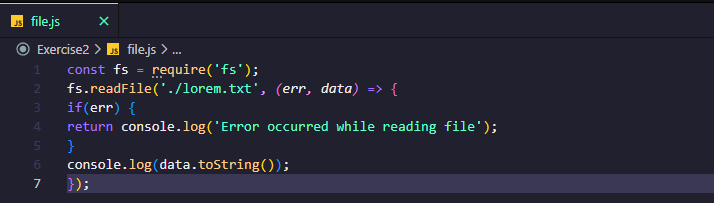
this:



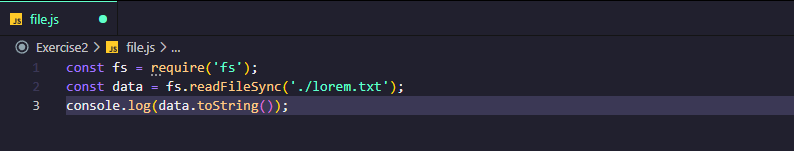
There are lots of methods provided under this node module to perform various tasks such as creating files, writing data into file, reading data from files etc.You can use fs.readFile() or fs.readFileSync() method to read files in Node.

For example:

Using the readFile() method.

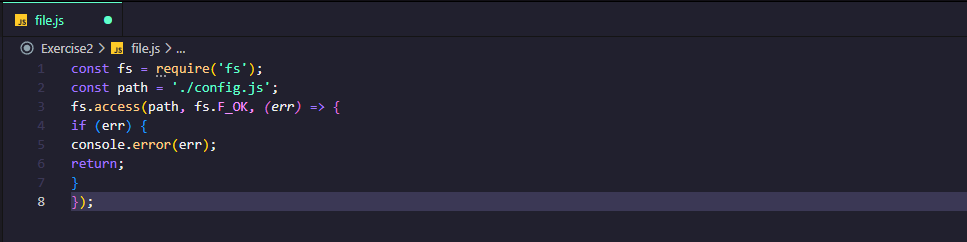


Using the readFileSync() method.



The simplest approach to check whether the file exists or not is by using the readFile() function.However, this function does open the file descriptor and occupies some memory too.If you just want to check the file existences in the system, I highly

recommend the access() function. Here is the code:



The file system module provides three methods to create files:

1. fs.open()

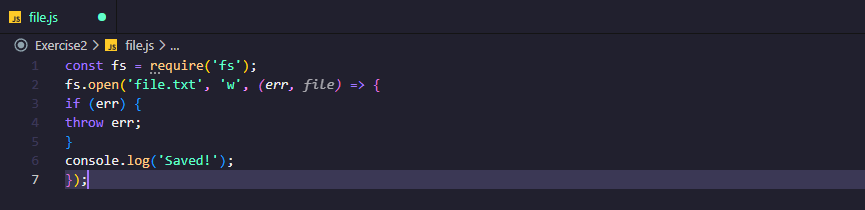
2. fs.writeFile()

3. fs.appendFile()

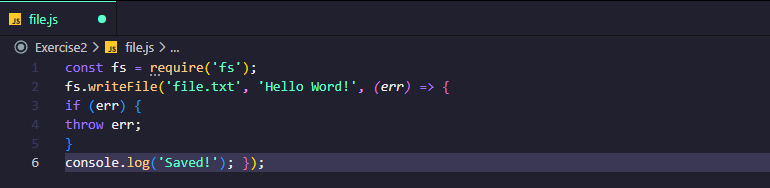
**fs.open()** method opens a new file or creates a new empty file if it does

not exist in the specified path. It takes the second parameter which acts as a flag such as w for writing, w+ for reading and writing etc.

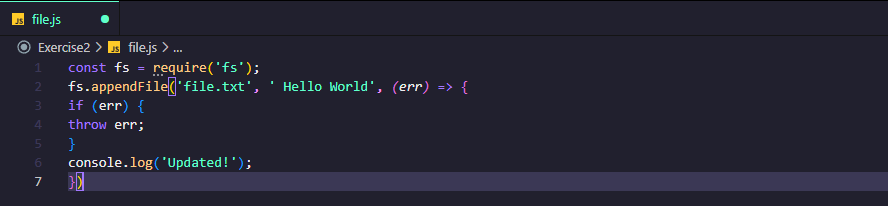
Code:



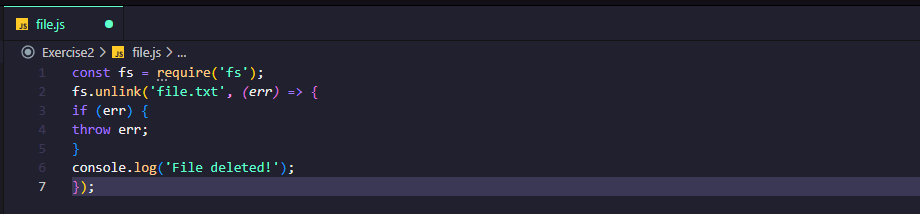
fs.writeFile() method allows you to create or replace files with the content.If a file exists, it will replace the content with the provided content and if the file does not exist, it will create it.



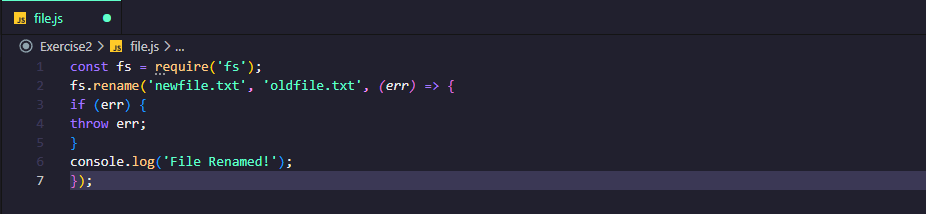
fs.appendFile() method appends the provided content at the end of the file.



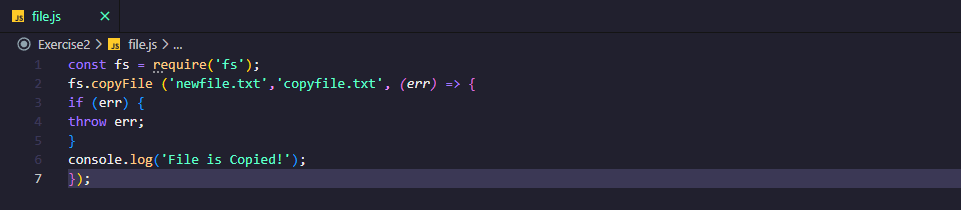
To delete a file, we can use fs.unlink() method.



To rename a file, we can use the fs.rename() method.



You can also copy files using the fs.copy() method.



## **Day 5 : Express framework.**

In this section , I learn about what is express? So I know Express is a popular framework to develop web applications in Node.Express is widely used and can be used to develop web applications such as Web Server, REST API Server, Streaming engine, etc.

And I am going to learn about express and we will create a simple web server with different routes.

Terminologies I will use in this section are:

● route: This means an endpoint. for example: facebook.com/profile

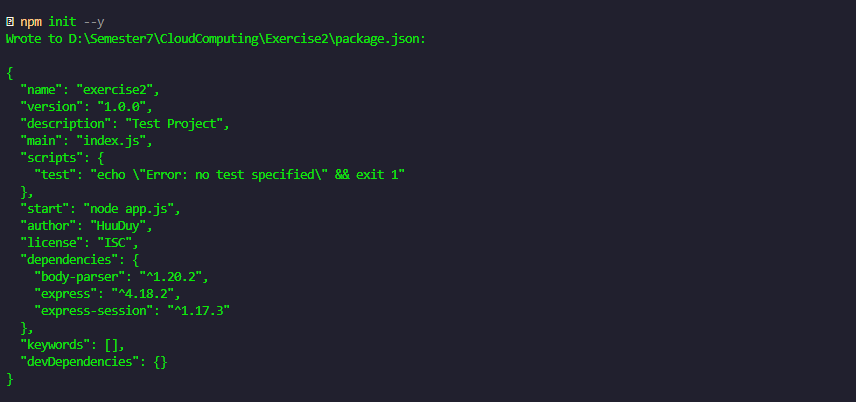
so profile is a route.

● middleware: A set of functions that will be executed in the chosen

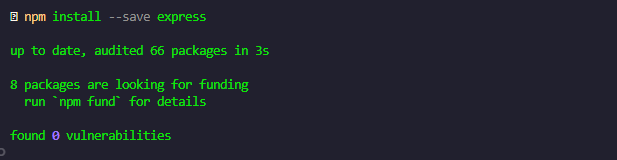
order.

So Let’s get started with me

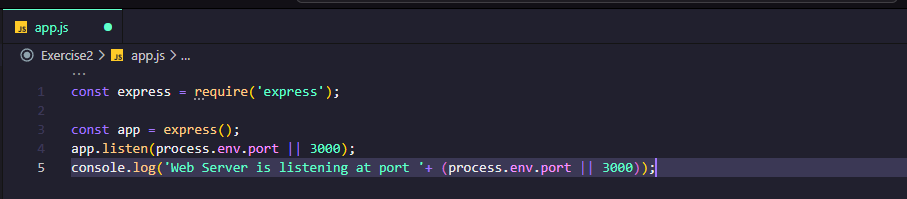
Create a new node project using this command **“ npm init –y “** .



This will create a sample package.json for your project. To install the express module, run the following this command **“ npm install --save express “** .



The latest version of the express framework will be installed in your project. Now create a new file and name it app.js. In this file, we will write our web server using express.



In the code shown above, we have required the express module and created a new instance of it. In the end, we have started our Server using the listen() function. Routers are simply an endpoint of a server.

For example, facebook.com/codeforgeek, here the codeforgeek is a route.

I need to create routers in our web application to serve different requests. We will create the following routes in our web application.

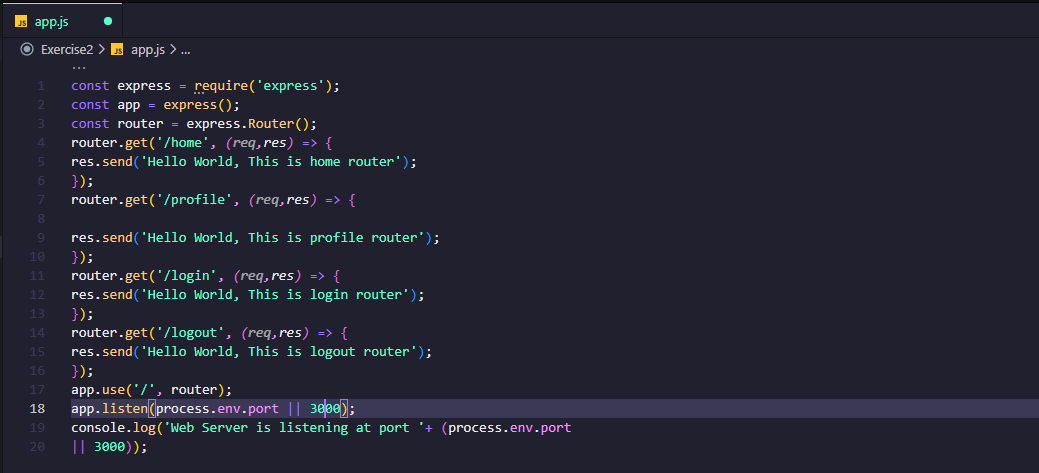
● home

● profile

● login

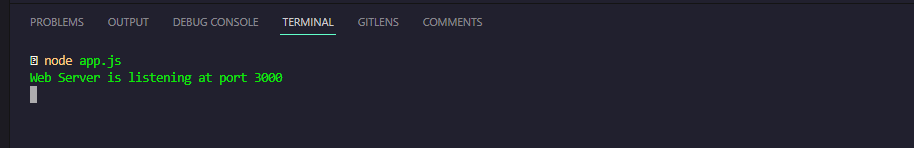
● logout

Express routers allow us to serve different HTTP methods such as GET, POST, PUT, DELETE, HEAD. Here is how to create a router.

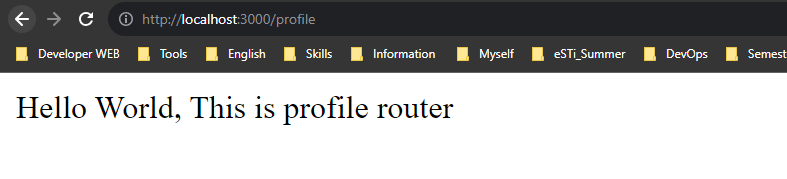


Let's run our application, save the file and run the code using the following this command : “node app.js”

After that , You should see the following message in the terminal.



Open your browser and visit the routes and you can see here is the profile page.



You can also send HTML/JSON/XML as a response.

Middleware functions as the name suggests can be used to make changes in the request/response lifecycle of the express. There are five types of middleware functions in the express.

1. Application middleware

2. Router middleware

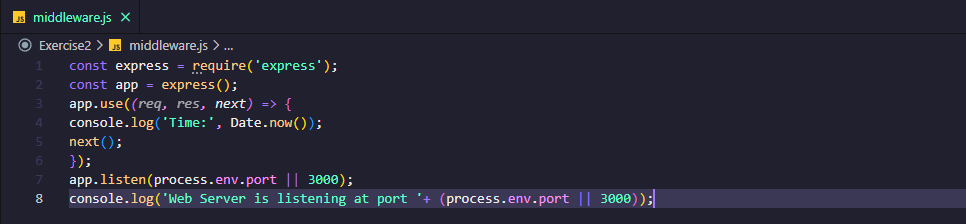
3. Error-handling middleware

4. Built-in middleware

5. Third-party middleware

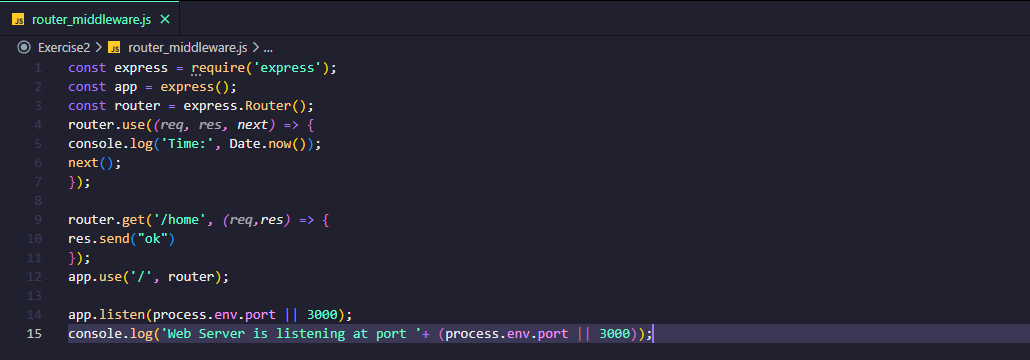
### **Application middleware**

I can use middleware in the application object of express. For example:



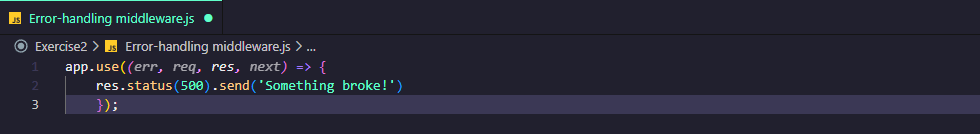
### **Router middleware**

In a similar way as application middleware, we can use router middleware. For example:



### **Error-handling middleware**

We can use this middleware to catch errors.



### **Built-in middleware**

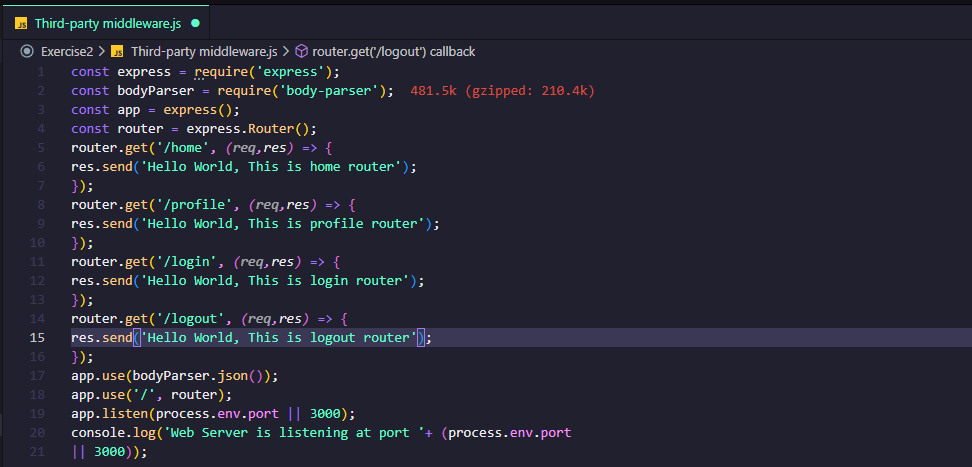
Express provides some middleware by default such as static, json etc.

**Third-party middleware**

We can use third party middlewares such as body-parser. Here is how

we can install third party middleware with command **: “npm install body-parser”**

To use this middleware, we need to require our code and load it.



We can handle sessions in Express using express-session middleware.We can use this middleware to create, track and delete sessions with command :

**“npm install --save express-session”**

To use this module in our code, load this as a middleware.

**app.use(session({secret: 'some secrets'}));**

Now, we can use req.session object to create, track and delete sessions.

## **Day 6 : Databases - MySQL, MongoDB, PostgreSQL and Redis.**

In this section , I learnt about integration of databases with Node.

I am going to cover the following databases:

● MySQL

● MongoDB

● PostgreSQL

● Redis

### **So let get started with MySQL.**

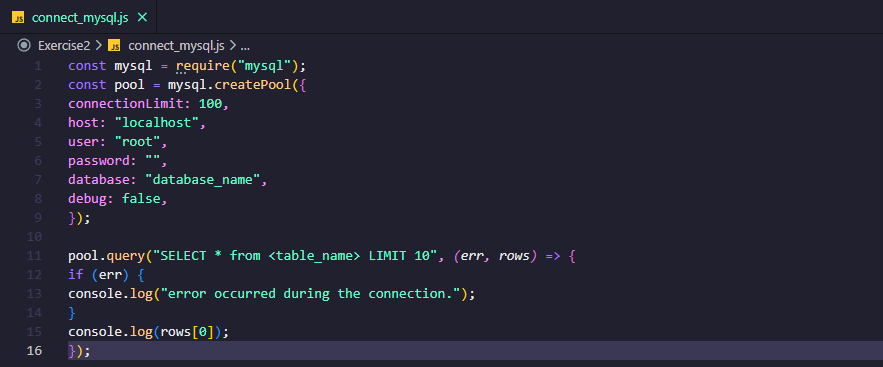
MySQL is a very popular SQL database. Learn how to connect, query

and use it with NodeJS.MySQL is a very popular database and has been used in millions ofapplications. We can use MySQL with Node as well.

This is a note important : *You need to have MySQL database installed in your system before proceeding.*

And after that , you install the module with command **: “npm install --save mysql”**

When I installed My SQL databaseand module in my system.Next , I can establish the connection to MySQL engine.



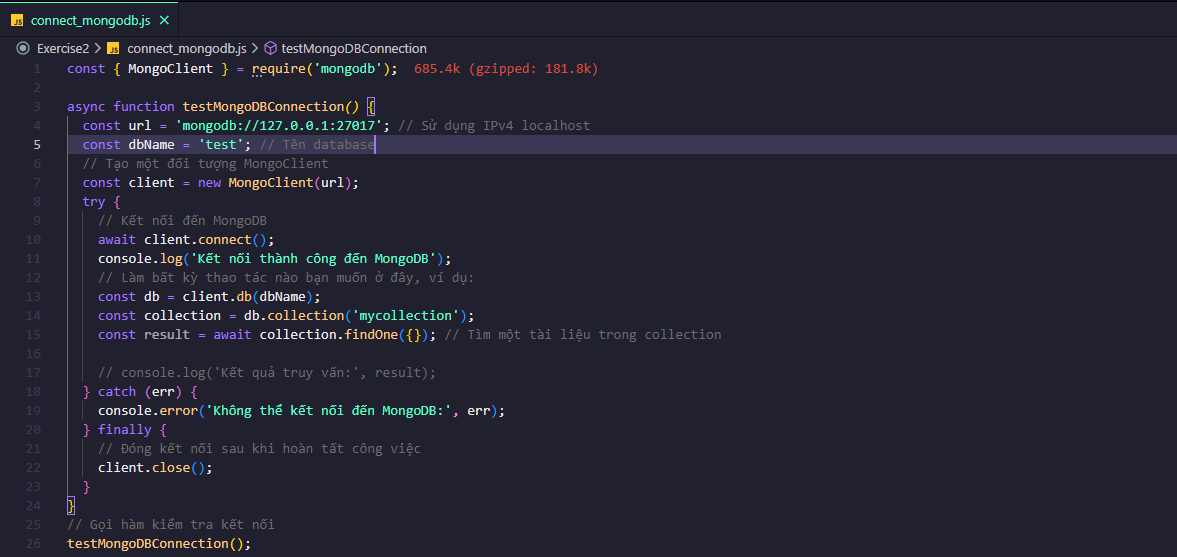
So I can execute the queries such as INSERT, UPDATE and DELETE in a similar fashion.

### **So Let’s learn about MongoDB.**

MongoDB is a very popular NoSQL database. Learn how to connect,query and use it with NodeJS. MongoDB is one of the most popular general purpose NoSQL database engines. You can use MongoDB to develop a wide range of applications.

This is a note important : *You need to have MongoDB database installed in your system before proceeding.*

And after that , you install the module with command **: “npm install --save mongodb”**

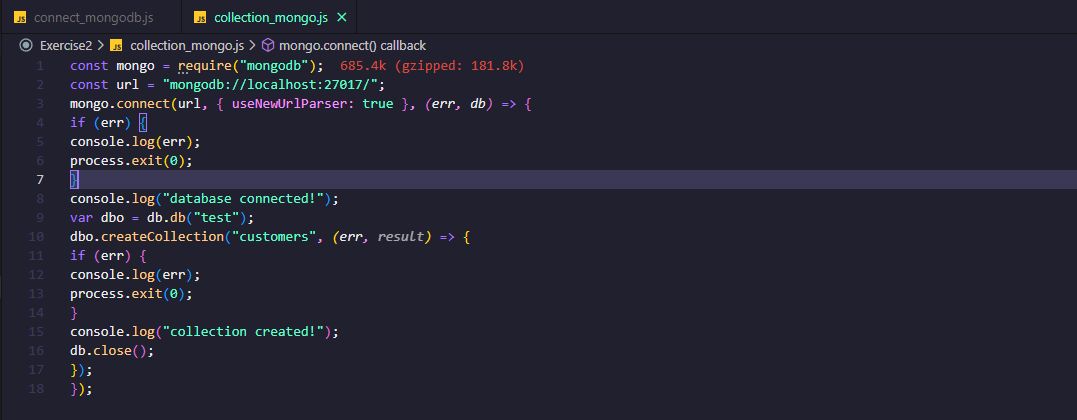


MongoDB runs on port 27017. We can connect to any database we like to work with.

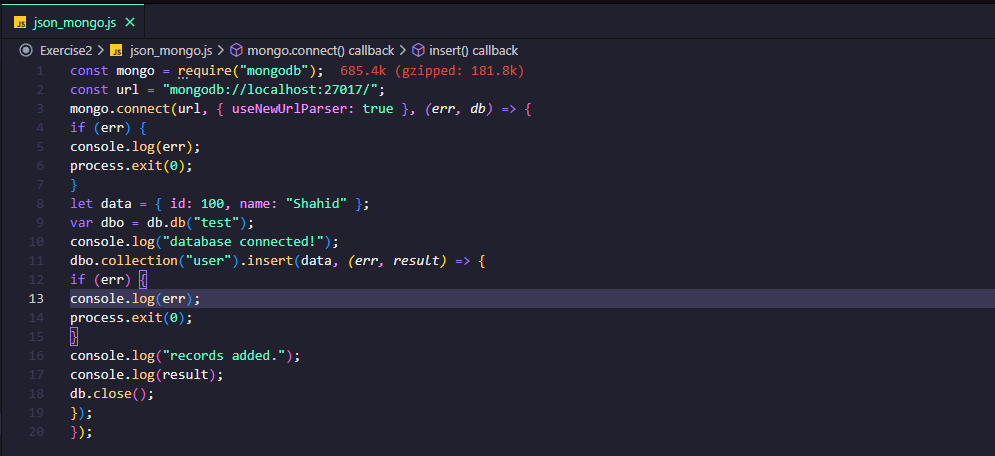
In the code shown above, we are connecting to the test database. We are using connect() method to establish a connection with the database. Save the code above in a file called connect\_mongodb.js' and run the file:



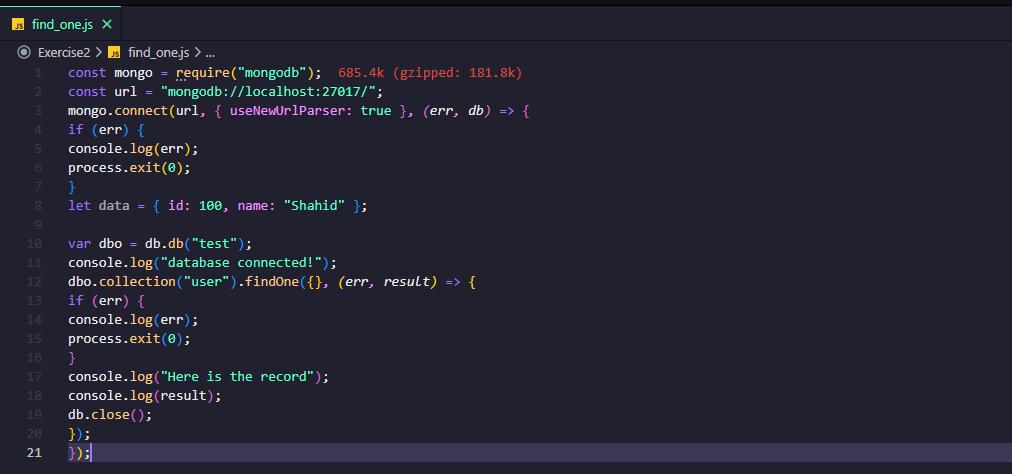
Think of collection as a table in SQL database. You can either create a collection in the MongoDB shell or you can do it in code. Here is how you can do it in code.



MongoDB stores data in a JSON format. JSON is a key-value based data format widely used across various layers of softwares.



You can also search for the data inside the collection using findOne() function.



### **So let get started with PostgreSQL.**

PostgreSQL is a very popular SQL database. Learn how to connect,

query and use it with NodeJS.PostgreSQL, also referred to as Postgres, is a free and popular relational database system. Postgres competes with relational databases like MySQL, SQL Server or MariaDB.

This is a note important : *You need to have Postgress database installed in your system before proceeding.*

We need to create a user credential in Postgres in order to connect to the database. Connect to Postgres using the default credentials. Run this command in your terminal.

* **“psql postgres”**

Create a new user.

* **CREATE ROLE codeforgeek WITH LOGIN PASSWORD 'somepassword'**

Give permission to the new user.

## **Day 7 : Deployment - deploying applications in Digitalocean**

## **Server.**