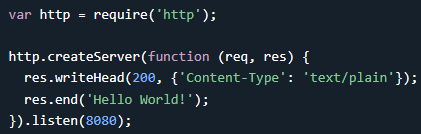
NODE.JS

Node.js is an open source server environment. Node.js allows you to run JavaScript on the server.

Example –

Node.js is free. Node.js files have extension ".js"

Node.js runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)

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 PHP or ASP handles a file request:

Sends the task to the computer's file system. Waits while the file system opens and reads the file.

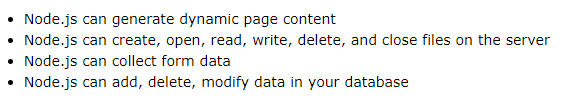
Returns the content to the client. Ready to handle the next request.

Here is how Node.js handles a file request:

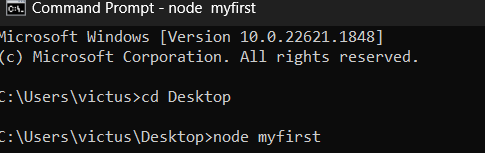
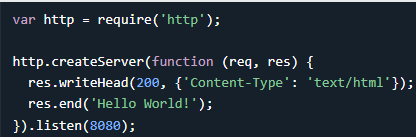
* Sends the task to the computer's file system.
* Ready to handle the next request.
* When the file system has opened and read the file, the server returns the content to the client.

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

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



Hello World in Node.js 🡪



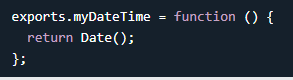
What is a Module in Node.js?

Consider modules to be the same as JavaScript libraries. A set of functions you want to include in your application.

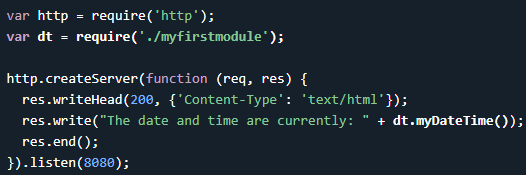
Node.js Build in modules - <https://www.w3schools.com/nodejs/ref_modules.asp>

To include a module, use the require() function with the name of the module:  Now your application has access to the HTTP module, and is able to create a server.

You can create your own modules, and easily include them in your applications. The following example creates a module that returns a date and time object –

Use the exports keyword to make properties and methods available outside the module file. Save the code above in a file called "myfirstmodule.js"

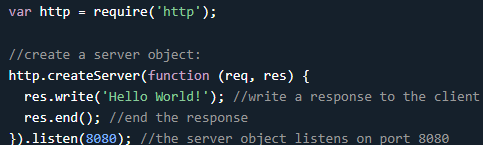
Use the module "myfirstmodule" in a Node.js file:



HTTP Module

Node.js has a built-in module called HTTP, which allows Node.js to transfer data over the Hyper Text Transfer Protocol (HTTP). The HTTP module can create an HTTP server that listens to server ports and gives a response back to the client.

Use the createServer() method to create an HTTP server. The function passed into the http.createServer() method, will be executed when someone tries to access the computer on port 8080.

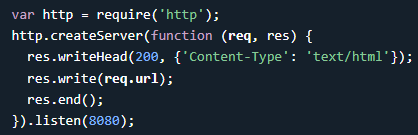


If the response from the HTTP server is supposed to be displayed as HTML, you should include an HTTP header with the correct content type. 

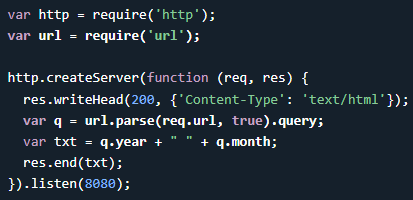
The first argument of the res.writeHead() method is the status code, 200 means that all is OK, the second argument is an object containing the response headers.

The function passed into the http.createServer() has a req argument that represents the request from the client, as an object (http.IncomingMessage object).

This object has a property called "url" which holds the part of the url that comes after the domain name:

localhost:8080/summer or /xyz

Split the Query String - There are built-in modules to easily split the query string into readable parts, such as the URL module.

 eg output - 

Whats the point of splitting Query string –

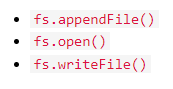
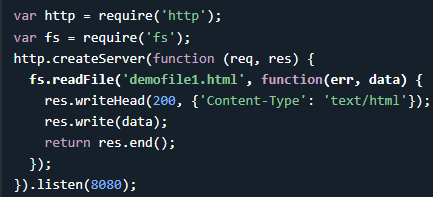
By splitting the query string, you can:

* Access and extract specific data.
* Perform server-side processing.
* Enable dynamic web page functionality.

The Node.js file system module allows you to work with the file system on your computer. To include the File System module, use the require() method:

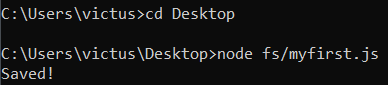
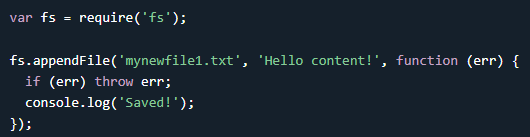


The fs.readFile() method is used to read files on your computer. Assume we have the following HTML file (located in the same folder as Node.js). With H1 and <p> tags.

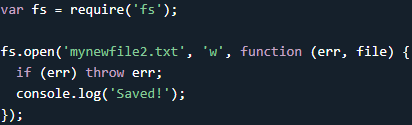
Create Files

The File System (fs) module has methods for creating new files (B).

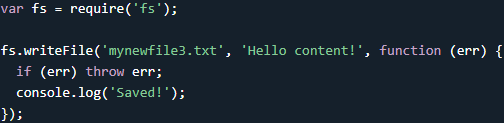
The fs.appendFile() method appends specified content to a file. If the file does not exist, the file will be created. Create a file using appendFile()



The fs.open() method takes a "flag" as the second argument, if the flag is "w" for "writing", the specified file is opened for writing. If the file does not exist, an empty file is created.



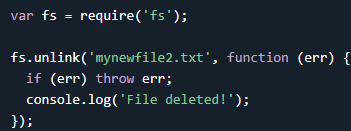
The fs.writeFile() method replaces the specified file and content if it exists. If the file does not exist, a new file, containing the specified content, will be created.



Update Files🡪The File System module has methods for updating files

 Same as Above

To delete a file with the File System module, use the fs.unlink() method. The fs.unlink() method deletes the specified file.

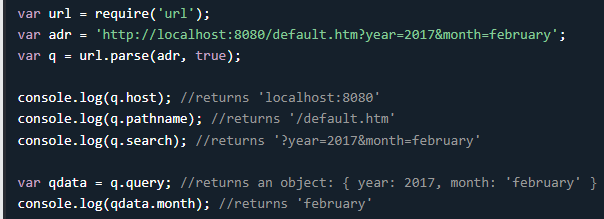


To rename a file with the File System module, use the fs.rename() method. The fs.rename() method renames the specified file.



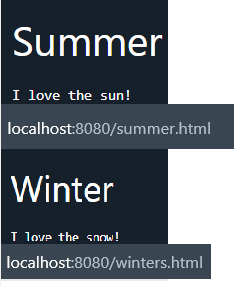
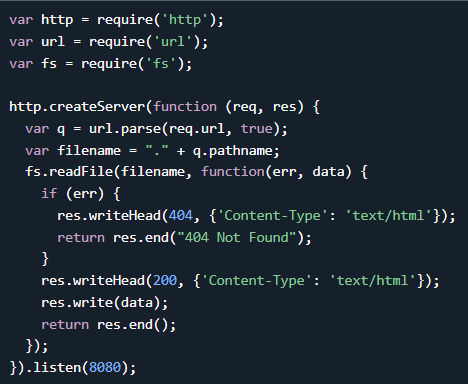
Node.js URL Module🡪The URL module splits up a web address into readable parts. To include the URL module, use the require() method.

Parse an address with the url.parse() method, and it will return a URL object with each part of the address as properties –



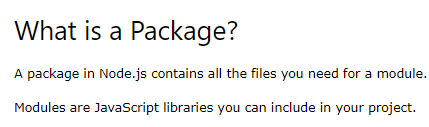
Node.js File Server

Now we know how to parse the query string, and in the previous chapter we learned how to make Node.js behave as a file server. Let us combine the two, and serve the file requested by the client. Create two html files and save them in the same folder as your node.js files.



Ref - <https://www.w3schools.com/nodejs/nodejs_url.asp>

NPM - NPM is a package manager for Node.js packages, or modules if you like. www.npmjs.com hosts thousands of free packages to download and use. The NPM program is installed on your computer when you install Node.js.

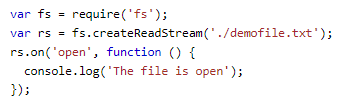


To download a package 🡪  (easy npm -i)

Now you have downloaded and installed your first package! NPM creates a folder named "node\_modules", where the package will be placed. All packages you install in the future will be placed in this folder.

My project now has a folder structure like this: 

Node.js Events🡪Every action on a computer is an event. Like when a connection is made or a file is opened. Objects in Node.js can fire events, like the readStream object fires events when opening and closing a file:

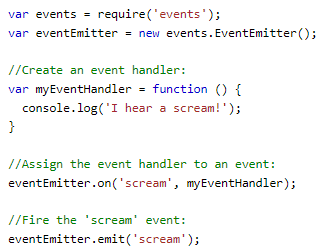


Node.js has a built-in module, called "Events", where you can create-, fire-, and listen for- your own events. To include the built-in Events module use the require() method. **In addition, all event properties and methods are an instance of an EventEmitter object.** To be able to access these properties and methods, create an EventEmitter object:



You can assign event handlers to your own events with the EventEmitter object. In the example below we have created a function that will be executed when a "scream" event is fired.

To fire an event, use the emit() method.

 Create 🡪Assign🡪FIRE

Node.js Upload Files🡪There is a very good module for working with file uploads, called "Formidable". The Formidable module can be downloaded and installed using NPM:  🡪 

Now you are ready to make a web page in Node.js that lets the user upload files to your computer: Lets see step by step –

Create a Node.js file that writes an HTML form, with an upload field:

 Create HTML form

Now, Include the Formidable module to be able to parse the uploaded file once it reaches the server. When the file is uploaded and parsed, it gets placed on a temporary folder on your computer.



When a file is successfully uploaded to the server, it is placed on a temporary folder. The path to this directory can be found in the "files" object, passed as the third argument in the parse() method's callback function. To move the file to the folder of your choice, use the File System module, and rename the file:



Instead of formidable we did it with muttler, as it was giving error with formidable. 

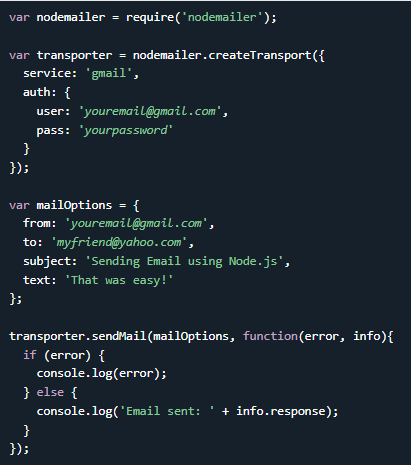
multer is a popular middleware for handling file uploads in Node.js web applications. It simplifies the process of accepting and processing multipart/form-data, which is the format used for file uploads in HTML forms.

Multer adds a multipart/form-data parser to the Express.js framework, allowing you to handle file uploads with ease. It provides several features and options for controlling the upload process, including file size limits, file type validation, and destination storage.



Node.js Send an Email🡪The Nodemailer module makes it easy to send emails from your computer. The Nodemailer module can be downloaded and installed using npm: 🡪

Now you are ready to send emails from your server. Use the username and password from your selected email provider to send an email.



And that's it! Now your server is able to send emails. To send an email to more than one receiver, add them to the "to" property of the mailOptions object, separated by commas.

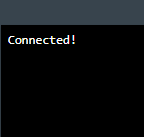
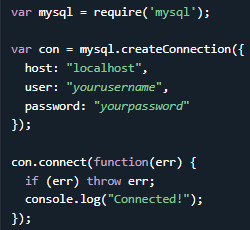
To send HTML formatted text in your email, use the "html" property instead of the "text" property: 

NODE and SQL

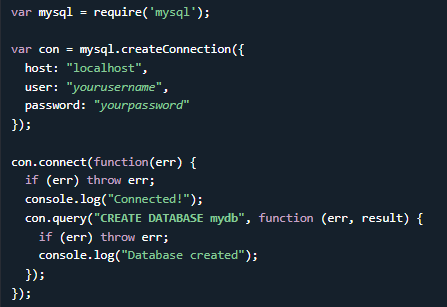
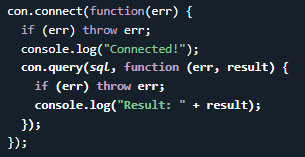
Node.js can be used in database applications. One of the most popular databases is MySQL.

Once you have MySQL up and running on your computer, you can access it by using Node.js. To access a MySQL database with Node.js, you need a MySQL driver. To download and install the "mysql" module, open the Command Terminal and execute the following: Now you have downloaded and installed a mysql database driver. Node.js can use this module to manipulate the MySQL database: 

Start by creating a connection to the database. Use the username and password from your MySQL database.

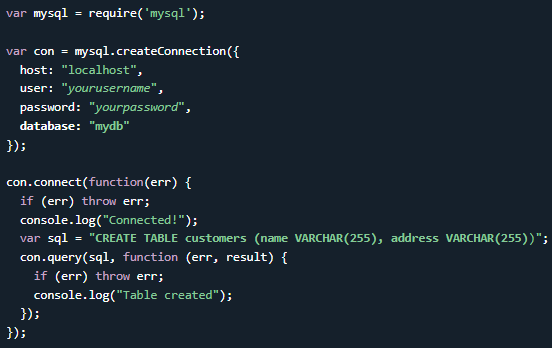
Save the code & 

Use SQL statements to read from (or write to) a MySQL database. This is also called "to query" the database. The connection object created in the example above, has a method for querying the database. The query method takes an sql statements as a parameter and returns the result.



To create a database in MySQL, use the "CREATE DATABASE" statement – (B) above |Save and run

To create a table in MySQL, use the "CREATE TABLE" statement. Make sure you define the name of the database when you create the connection so that table can be created in that database. Save the code and run.



When creating a table, you should also create a column with a unique key for each record. This can be done by defining a column as **"INT AUTO\_INCREMENT PRIMARY KEY**" which will insert a unique number for each record. Starting at 1, and increased by one for each record.



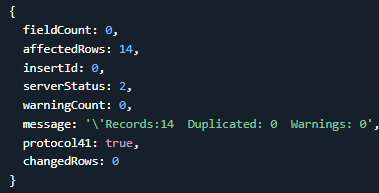
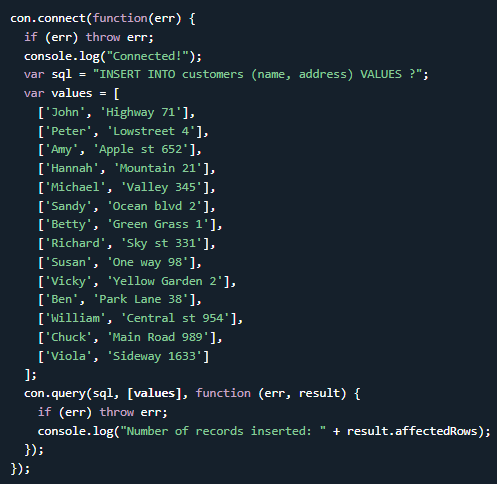
If the table already exists, use the ALTER TABLE keyword

To fill a table in MySQL, use the "INSERT INTO" statement.



To insert more than one record, make an array containing the values, and insert a question mark in the sql, which will be replaced by the value array:

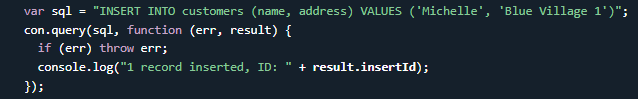
**INSERT INTO customers (name, address) VALUES ?**



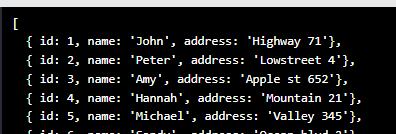
When executing a query, a result object is returned. The result object contains information about how the query affected the table. The result object returned from the example above looks like this.

// output - 

For tables with an auto increment id field, you can get the id of the row you just inserted by asking the result object.

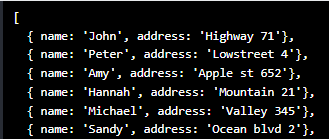


To select data from a table in MySQL, use the "SELECT" statement - This will display all the records of table customers in console. Note here output of function is err, result and fields also.

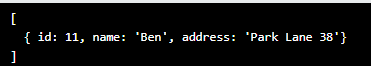
// Output

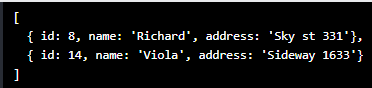
To select only some of the columns in a table, use the "SELECT" statement followed by the column name.



As you can see from the result of the example above, the result object is an array containing each row as an object. To return e.g. the address of the third record, just refer to the third array object's address property: Output -   
The third parameter of the callback function is an array containing information about each field in the result. You can try  with select of some columns from customers to see.

The fields object is an array containing information about each field as an object.

When selecting records from a table, you can filter the selection by using the "WHERE" statement: 

You can also select the records that starts, includes, or ends with a given letter or phrase. Use the '%' wildcard to represent zero, one or multiple characters: 

The reason is that if you enter something in the browser URL, it by default always is a GET request,which means it can't work because we have no get route that handles slash nothing, we just have a post route.