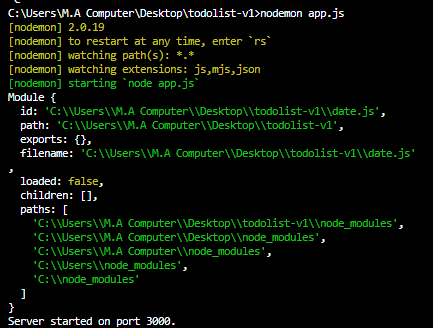
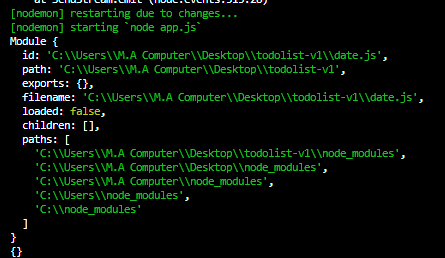
***10. Understanding Node Module Exports How to Pass Functions and Data between Files***

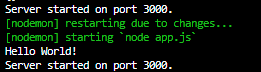
Create date.js file and paste date code

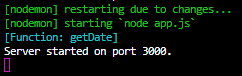
<http://www.nodejs.org/dist/latest-v10.x/docs/api/modules.html#modules_the_module_object>

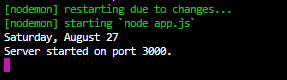
<https://nodejs.org/dist/latest-v18.x/docs/api/modules.html#the-module-object>











**Static methods:**

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object>

ways to declare JavaScript functions

<https://dmitripavlutin.com/6-ways-to-declare-javascript-functions/>

An example of function declaration:

// function declaration

function isEven(num) {

return num % 2 === 0;

}

isEven(24); // => true

isEven(11); // => false

**Way 2:**

// Function expression: starts with "const"

const isTruthy = function(value) {

return !!value;

};

// Function expression: an argument for .filter()

const numbers = ([1, false, 5]).filter(function(item) {

return typeof item === 'number';

});

// Function expression (IIFE): starts with "("

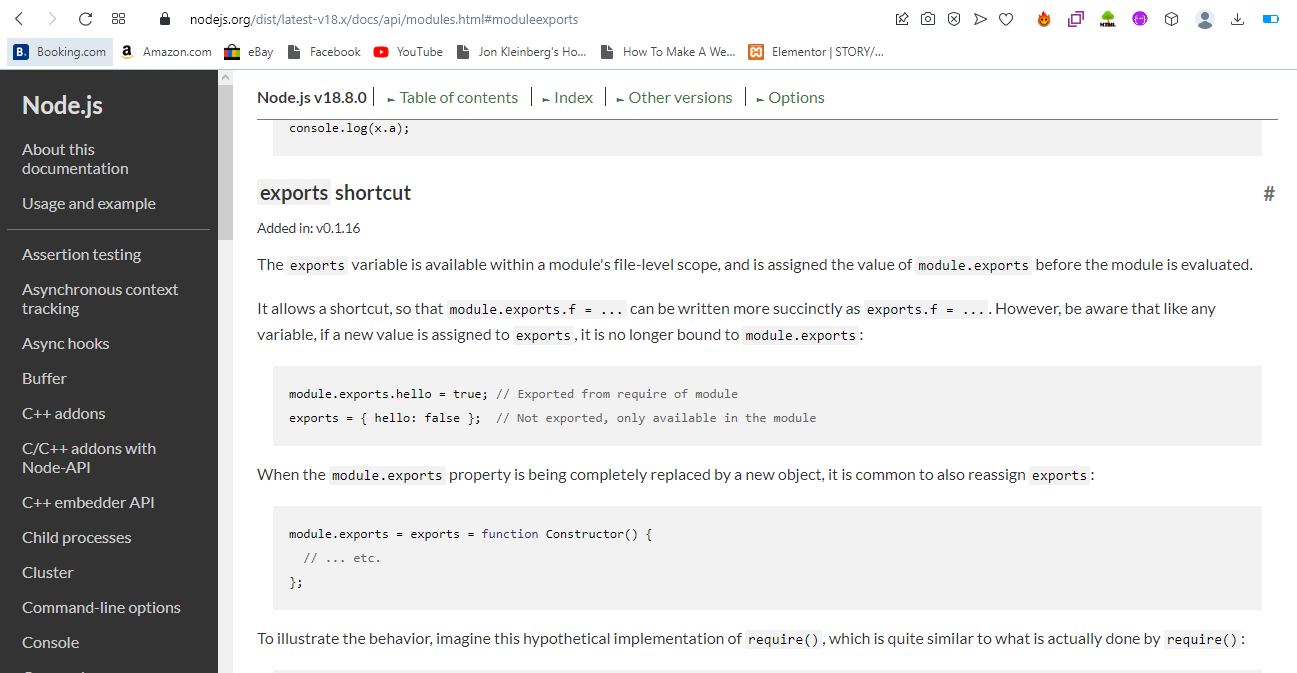
(function messageFunction(message) {

return message + ' World!';

})('Hello')

**exports shortcut:**

<https://nodejs.org/dist/latest-v18.x/docs/api/modules.html#moduleexports>



**Const**

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/const>

The same applies to arrays. Assigning a new array to the variable throws an error "Assignment to constant variable".

const MY\_ARRAY = [];

MY\_ARRAY = ['B'];

Still, it's possible to push items into the array and thus mutate it.

MY\_ARRAY.push('A'); // ["A"]

It’s not possible to add things const array



***Code:***

**Index.html:**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title>To Do List</title>

</head>

<body>

<h1>It's a weekday!</h1>

<p>Why are you not working.</p>

</body>

</html>

**App.js:**

//jshint esversion : 6

const express = require("express");

const bodyParser = require("body-parser");

const date = require(\_\_dirname + "/date.js");

//console.log(date);

//console.log(date);

const app = express();

const items = ["Buy Food", "Cook Food", "Eat Food"]; //let to const

const workItems = []; //let to const

app.set('view engine', 'ejs');

app.use(bodyParser.urlencoded({

extended: true

}));

app.use(express.static("public")); //use pulic folder as static

app.get("/", function(req, res) {

//let day = date();

const day = date.getDate(); //only day let to const

//let day = date.getDate();

res.render("list", {

listTitle: day, //replace the entire value of kind of day with current day

//newListItem: item //Its written to fix below problem of server

newListItems: items

});

});

app.post("/", function(req, res) {

const item = req.body.newItem;

if (req.body.list === "Work") {

workItems.push(item);

res.redirect("/work");

} else {

items.push(item);

res.redirect("/"); //simple form of above line

}

console.log(req.body);

//console.log(item);

//res.render("list",{newListItem:item});

});

app.get("/work", function(req, res) {

res.render("list", {

listTitle: "Work List",

newListItems: workItems

});

});

app.post("/work", function(req, res) { //post route

const item = req.body.newItem; //let to const

workItems.push(item);

res.redirect("/work");

});

//create about route

app.get("/about", function(req, res) {

res.render("about");

})

app.listen(3000, function() {

console.log("Server started on port 3000.");

});

**Date.js:**

//jshint esversion: 6

//console.log(module); //module object = https://nodejs.org/dist/latest-v18.x/docs/api/modules.html#the-module-object

//Above module is required to run all this code

//module.exports="Hello World!";

//module.exports=getDate;

//let's run all below functions

/\*

module.exports.getDate = getDate;

function getDate() {

let today = new Date();

let options = {

weekday: "long",

day: "numeric",

month: "long"

//year:"numeric"

};

//var day=today.toLocaleDateString("en-US"); //8/24/2022

let day = today.toLocaleDateString("en-US", options); //add date format in us english format

//Wednesday, August 24, 2022

return day;

}

module.exports.getDay = getDay;

function getDay() {

let today = new Date();

let options = {

weekday: "long"

};

//var day=today.toLocaleDateString("en-US"); //8/24/2022

let day = today.toLocaleDateString("en-US", options); //add date format in us english format

//Wednesday

return day;

}

console.log(module.exports);

\*/

/\*

//Function Expressions

module.exports.getDate = function() {

let today = new Date();

let options = {

weekday: "long",

day: "numeric",

month: "long"

//year:"numeric"

};

//var day=today.toLocaleDateString("en-US"); //8/24/2022

return today.toLocaleDateString("en-US", options); //add date format in us english format

//Wednesday, August 24, 2022

}

module.exports.getDay function() {

let today = new Date();

let options = {

weekday: "long"

};

//var day=today.toLocaleDateString("en-US"); //8/24/2022

return today.toLocaleDateString("en-US", options); //add date format in us english format

//Wednesday

}

\*/

//Function Expressions with export shortcuts

exports.getDate = function() {

const today = new Date(); //change all variables from let to const

const options = {

weekday: "long",

day: "numeric",

month: "long"

//year:"numeric"

};

//var day=today.toLocaleDateString("en-US"); //8/24/2022

return today.toLocaleDateString("en-US", options); //add date format in us english format

//Wednesday, August 24, 2022

}

exports.getDay = function() {

const today = new Date();

const options = {

weekday: "long"

};

//var day=today.toLocaleDateString("en-US"); //8/24/2022

return today.toLocaleDateString("en-US", options); //add date format in us english format

//Wednesday

}

**views\header.ejs:**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title>To Do List</title>

<link rel="stylesheet" href="css/styles.css">

</head>

<body>

**views\footer.ejs:**

</body>

<footer>

Copyright 2022 The GCU

</footer>

</html>

**views\about.ejs:**

<%- include("header") -%>

<h2>This is the about page</h2>

<%- include("footer") -%>

**views\list.ejs:**

<%- include("header") -%>

<div class="box" id="heading">

<h1><%= listTitle %></h1>

</div>

<!--

<li>Buy Food</li>

<li>Cook Food</li>

<li>Eat Food</li>

-->

<div class="box">

<% for(var i=0; i<newListItems.length;i++){ %>

<div class="item">

<input type="checkbox">

<p><%= newListItems[i] %></p>

</div>

<% } %>

<form class="item" action="/" method="post">

<input type="text" name="newItem" placeholder="New Item" autocomplete="off">

<button type="submit" name="list" value=<%= listTitle %>>+</button>

</form>

</div>

<%- include("footer") -%>

<!-- See all the things are same -->

**public\css\styles.css:**

html {

background-color: #E4E9FDD;

background-image: -webkit-linear-gradient(65deg, #A683E3 50%, #E4E9FD 50%);

min-height: 1000px;

font-family: 'helvetica neue'

}

h1 {

color: #fff;

padding: 10px;

}

.box {

max-width: 400px;

margin: 50px auto;

background: white;

border-radius: 5px;

box-shadow: 5px 5px 15px -5px rgba(0, 0, 0, 0.3);

}

#heading {

background-color: #A683E3;

text-align: center;

}

.item {

min-height: 70px;

display: flex;

align-items: center;

border-bottom: 1px solid #F1F1F1;

}

.item:last-child {

border-bottom: 0;

}

/\* All the paragraph elements that comes after checked. It allows purple line on paragraph when tick the checked box \*/

input:checked+p {

text-decoration: line-through;

text-decoration-color: #A683E3;

}

/\* Apply this to only those inputs that's have type checkedbox \*/

input[type="checkbox"] {

margin: 20px;

}

p {

margin: 0;

padding: 20px;

font-size: 20px;

font-weight: 200;

color: #00204a;

}

form {

text-align: center;

margin-left: 20px;

}

button {

min-height: 50px;

width: 50px;

border-radius: 50px;

border-color: transparent;

background-color: #A683E3;

color: #fff;

font-size: 30px;

padding-bottom: 6px;

border-width: 0;

}

input[type="text"] {

text-align: center;

height: 60px;

top: 10px;

border: none;

background: transparent;

font-size: 20px;

font-weight: 200;

width: 313px;

}

input[type="text"]:focus {

outline: none;

box-shadow: inset 0 -3px 0 0 #A683E3;

}

/\* It only target to placeholder like New Item in this case. \*/

::placeholder {

color: grey;

opacity: 1;

}

footer {

color: white;

color: rgba(0, 0, 0, 0.5);

text-align: center;

}