# Cách chạy : download node js, git

Tạo file app.js trong folder abc 🡺 git bash here vào folder

In ra màn hình : console.log(“Hieu” + “Tran”);

# Khai báo biến:

var x = "5";

var y = "7";

// chuyển đổi chuỗi sang số

x = parseInt(x);

y = parseInt(y);

console.log(x + y);

## Cấu Trúc Điều Khiển

### If else:

var diem = 5;

if (diem >= 5){

console.log ("Passed");

}

else{

console.log ("Failed");

}

### For:

for (var i= 1; i<=10; i++){

console.log("Dong thu" + i);

}

### While:

var dong = 1;

while(dong <=5){

console.log("Day la dong thu:" + dong);

dong ++;

}

### Switch case:

var day = "Saturday";

switch (day) {

case "Friday":

console.log("Thanks God it's Friday");

break;

case "Saturday":

console.log("Th? 7 máu ch?y v? tim");

break;

case "Sunday":

console.log("Ng? thôi!")

break;

defaut:

console.log("Xin m?i làm vi?c");

break;

}

### Array

**// mang rong var mang = [];**

var mang = ["buoi","cam","chuoi"];

**//xuất ra chữ cam**

console.log(mang[1]);

**// add thêm mảng**

mang.push("dua hau");

**// đếm phần tử**

console.log(mang.length);

# Function : có 3 cách để tạo hàm

## C1:

function tong(a,b){

return a + b;

}

**// khai bao 1 bien de lay ham return**

var x = tong(5,7);

console.log(x);

## C2:

**// goi ham khac thong qua ham**

function hello(){

console.log('Hieutran');

}

function goiHam(fn){

fn();

}

goiHam(hello);

## C3:

**// khai bao bien thong qua ham gì do**

var tong = function(){

console.log('Hieutran goi qua bien');

}

tong();

# Tạo module liên kết đến file khác:

## Step1: Tạo 1 file Hello.js

Var name = “Trung tam dao tao viettin”;

Module.exports = name

## Step2: Tạo 1 file khác export liên kết file Hello.js: export.js

**// “./ “cùng cấp để gọi ra**

Var x = require(./hello.js);

Console.log(x);

# OOP\_1:

## // khai bao class

var person = {

ho : "tran", **// thuoc tinh**

ten : "hieu", **// thuoc tinh**

**// phuong thuc**

chaomung: function (){

console.log("chao ban " + this.ho + " " + this.ten);

}

}

## // ham static de dùng đưa person ra

person.chaomung();

## // xuat ra tên

console.log(person["ten"]);

# OOP\_2:

## // tạo 1 class truyền tham số

function KhoaHoc(ten, hocphi){

// trỏ đến thuộc tính

this.Ten = ten;

this.HocPhi = hocphi;

}

## // phương thức

KhoaHoc.prototype.mota = function(){

console.log("Hello " + this.Ten + " " + this.HocPhi);

}

## // đối tượng

var kh = new KhoaHoc("Notejs", 800);

## // đối tượng dùng phương thức

kh.mota();

# Buffer:

## // chuyen chuoi "Hello" thanh buffer "<Buffer 48 65 6c 6c 6f>"

var buffer = new Buffer("Hello", "utf8");

console.log(buffer);

## // chuyen buffer thanh chuoi

console.log(buffer.toString());

## // chuyen buffer thanh JSON

console.log(buffer.toJSON());

# Xử lý file:

## // đọc file phải gọi đến module (require) fs : đọc file trên server or đọc file upload

var fs = require("fs");

## // hằng số \_\_dirname là node có sẵn để gọi đường dẫn mặc định

var noidung = fs.readFileSync( \_\_dirname + "/danhsach.txt", "utf-8");

console.log(noidung);

console.log(noidung.toString());

# Webserver (text):

**// gọi module**

var http = require("http");

**// khởi tạo 1 con server trong đó có req: từ client sẽ request server và khi đó res: trả về cho client mở ra những gì khác hàng xem vd: image, text**

http.createServer(function(req, res){

**//writeHead : truyền vào mã file vd : 404 là lỗi trả về, 200 là mã gì đó ở đây là chuỗi**

res.writeHead(200,{"Content-Type":"text/plain"});

**// trả về ở đây là chuỗi**

res.end("Hieutran");

**// mở port 8888 để trả về**

}).listen(8888);

* Mở chrome lên localhost: 8888

# Webserver (html):

## Step1: Tạo 1 file index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

</head>

<body>

<h1>{NAME}</h1>

</body>

</html>

## Step2: Tạo 1 file xuly.js

// goi module http cho webServer

var http = require("http");

// goi module cho xu ly file (doc file, upload file)

var fs = require("fs");

http.createServer(function(req, res){

res.writeHead(200, {"Content-Type":"text/html"});

// lay file

var data = fs.readFileSync(\_\_dirname + "/index.html", "utf-8");

data = data.replace("{NAME}", "Hieutran");

res.end(data);

}).listen(7777);

# Webserver (pipe):

## Step1: Tạo 1 file index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

</head>

<body>

<h1>{NAME}</h1>

</body>

</html>

## Step2: Tạo 1 file xuly.js

// goi module http cho webServer

var http = require("http");

// goi module cho xu ly file (doc file, upload file)

var fs = require("fs");

http.createServer(function(req, res){

res.writeHead(200, {"Content-Type":"text/html"});

// lay file va do vao pipe

fs.createStream(\_\_dirname + “/index.html”).pipe(res)

}).listen(7777);

# Webserver (JSON):

## Step1: Tạo 1 file index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

</head>

<body>

<h1>{NAME}</h1>

</body>

</html>

## Step2: Tạo 1 file xuly.js

var http = require("http");

var fs = require ("fs");

http.createServer (function(req, res){

res.writeHead(200, {"Content-Type": "application/json"});

var obj = {

ho : "tran",

ten : "hieu",

namsinh : 1987

};

// do ve cho khach hang nho ham stringify(obj) bien obj thanh json

res.end(JSON.stringify(obj));

}).listen(8888);

# Cài đặt express: module bên ngoài

## Step 1: Tạo 1 file json: package.json

{

// cùng tên với folder

"name" : "SERVER",

"version" : "0.0.1",

"private" : "true",

"dependencies" : {

"express":"\*"

}

}

## Step2:

Dùng lệnh npm install

## Dựng server với module express:

// goi module express

var express = require("express");

var app = express();

// khai bao server qua http tạo server từ thằng app

var server = require("http").createServer(app);

server.listen(3000);

// chạy có đường dẫn localhost:3000/

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

// trả về cho khách hang là 1 chuỗi dung send

res.send("<font color=red>Hello Hieutran</font>");

});

# Trả về 1 file : sendfile

## Step1: Tạo 1 file index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

</head>

<body>

<h1>{NAME}</h1>

</body>

</html>

## Step2: tạo 1 file app.js

var express = require("express");

var app = express();

var server = require("http").createServer(app);

server.listen(3000);

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

res.sendFile(\_\_dirname + "/index.html");

});

# Routing:/ hello đường dẫn

var express = require("express");

var app = express();

var server = require("http").createServer(app);

server.listen(3000);

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

res.sendFile(\_\_dirname + "/index.html");

});

------------------------------------------------------------------------------------------------------------------------------------------

# \*\*\*\*Nodejs + Express + Postgresql\*\*\*\*

# Cách tạo project nodejs

npm init

Sau đó phần auth: hieutran

Chú ý là ko dung unikey để làm

Cơ chế:

- Triệu gọi freedomcode.cf lên server (request) sau đó server sử lý và response tải về cho khách hang xem

- Trên server node cài đặt module express để hổ trợ html, truy vấn database …

# Cài đặt goi module express:

npm install express

## tạo 1 file index.js match với file confile

## import express

var express = require("express");

var app = express();

var server = require("http").createServer(app);

server.listen(8000);

## ham get

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

res.send("<font color=red>getting</font>");

});

## ham post : dung phần mềm postman để kiểm tra

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

res.send("<font color=red>postting</font>");

});

# Truyền parameter ở dạng get

var express = require("express");

var app = express();

var server = require("http").createServer(app);

server.listen(3000);

app.get("/tinhtong/:so1/:so2",function(req, res){

## // lay req cua kh

var n = req.params.so1;

## // ep chuoi thanh integer

n = parseInt(n);

var m = req.params.so2;

m = parseInt(m);

var tong = n + m;

res.send("<h1>" + tong + "</h1>");

});

# Truyền parameter ở dạng post:

lấy tham số của post ko hổ trợ trong express phải dung body parser nodejs

**cài đặt body parser: npm install body-parser**

## import express

var express = require("express");

var bodyParser = require('body-parser');

var urlencodedParser = bodyParser.urlencoded({ extended: false });

var app = express();

var server = require("http").createServer(app);

server.listen(8000);

## // username password

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

var u = req.body.username;

var p = req.body.password;

res.send("username: " + u + " password: " + p );

});

Dùng postman: <http://localhost:8000/login> chọn post 🡺 body 🡺 x-www-form-urlencoded

Username :hieutran

Password: 123456

🡺 username: hieutran password: 123456

# EJS template engine: để xuất ra nhiều dòng trong get

Tải về 1 trang web giống html

# Cài đặt EJS :

npm install ejs

## step1:index.js

var express = require("express");

var bodyParser = require('body-parser');

var urlencodedParser = bodyParser.urlencoded({ extended: false });

var app = express();

var server = require("http").createServer(app);

server.listen(8000);

**// config ejs**

**// view engine : công thức; ejs: là một đuôi định dạnh của ejs**

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

**// khai bao layout nằm trong views để biết**

app.set("views", "./views");

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

## // render tạo ra giao diện vào lấy đúng tên file ejs

**res.render("hieutran");**

});

## Step2: hieutran.ejs

<!-- khai bao bien ejs-->

<% var n = "<font color=green>Green</font>"

var tuoi = 18;

%>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

</head>

<body>

<h1>Hello Hieu tran</h1>

<!-- <%= n%> cho bao mat ejs-->

<h2><%= n%></h2>

<!-- <%- n%> bao mat kem ejs-->

<h2><%- n%></h2>

<!-- cho if else-->

<% if(tuoi >=18){ %>

<h2>Tren 18 tuoi</h2>

<% }else{ %>

<h2>Duoi 18 tuoi</h2>

<% } %>

<!-- cho for-->

<% for(i=0; i<=5; i++ ){ %>

<h1>Day la dong thu: <%= i%></h1>

<% }%>

</body>

</html>

# Truyền tham số và mang

## Step1 : index.js

// import express

var express = require("express");

var bodyParser = require('body-parser');

var urlencodedParser = bodyParser.urlencoded({ extended: false });

var app = express();

var server = require("http").createServer(app);

server.listen(8000);

// config ejs

// view engine : công thức; ejs: là một đuôi định dạnh của ejs

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

// khai bao layout nằm trong views để biết

app.set("views", "./views");

## // truyền tham số

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

## // tạo 1 tên chitiet.ejs trong views

res.render("namsinh", {namsinh:1989});

});

## //truyền mảng

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

res.render("chitiet", {solieu:[1989, 1888, 7777, 4444]});

});

## Step 2: chitiet.ejs

<!--forEach để duyệt qua các mảng -->

<% solieu.forEach(function(i){ %>

<h1>Day la gia tri <%- i%></h1>

<%});%>

# Upload ảnh: npm install express multer ejs

## Step 1:index.js

var express = require("express");

var multer = require("multer");

var app = express();

// connect web port 3000

app.listen(3000, function(){

console.log('Connected server');

});

**//ejs**

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

**// khai bao layout nằm trong views để biết**

app.set("views", "./views");

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

res.render("form");

});

**// noi luu tru file tai len**

var storage = multer.diskStorage({

destination: function (req, file, cb) {

**// noi luu tru uploads folder**

cb(null, './uploads');

},

filename: function (req, file, cb) {

**// dat ten file ảnh dung voi ten goc**

cb(null, file.originalname);

}

});

**// khai bao bien luu tru**

var upload = multer({storage:storage});

// tai 1 file ==> upload.single('giatri ten trong form')

app.post('/upload', upload.single('file'), function (req, res) {

**// tai len va thong tin cua file**

console.log(req.file);

**//thong bao message thanh cong**

res.send('UPLOAD thanh cong');

});

## Step2:form.ejs

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Upload</title>

</head>

<body>

**<!-- them thuoc tinh multipart/form-data dể dùng cho multer ko thì ko xử lý-->**

<form action="http://localhost:3000/upload" method="POST" enctype="multipart/form-data">

<input type="file" name="file">

</br>

<input type="submit" value="upload">

</form>

</body>

</html>

------------------------------------------------------------------------------------------------------------------------------------------

# \*\*\*\*Mogodb\*\*\*\*

Là 1 ct csdl mã nguồn mở thiết kế theo hướng đối tượng, cấu trúc trong bảng ko tuân theo nào hết

Mỗi bảng ghi đc gán them id đc tự động xuất hiện 🡺 bảng ghi sẽ ghi đệm lên bộ nhớ ram k cần đọc từ ổ cứng từ truy vấn sau đợi 60s mới xuống HDD

* Điểm mạnh : hiệu suất tang, dễ mở rộng
* Điểm yếu : dễ mất dữ liệu khi bị cúp điện, ko rang buộc toàn vẹn mô hình giao dịch độ chính xác cao, k có cơ chế transition

Cài mongodb(mongodb.org) trong đó có ssl bảo mật. Tạo folder data trong ổ C

Add path C:\Program Files\MongoDB\Server\3.4\bin vào trong environment 🡺 mở cmd gõ mongod để chạy

Robomongo.org/download 🡺 create server 🡺create db 🡺 collection 🡺 insert document

{

"type" : "cupcakes",

"banh" : [],

"\_\_v" : 0

}

## Run mongo trong project:

**npm install express ejs multer mongoose body-parser**

Mongoose : module này giúp ta giao tiếp xử lý mongodb với nodejs nhanh hơn

Xây dựng schema, mỗi schema sẽ match với collection trong mongodb xác định mỗi document trong collection

## Step 1: đưa dữ liệu vào db index.js

var express = require("express");

var app = express();

app.listen(3000, function(){

console.log('Connected mongodb');

});

**// su dung ejs module**

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

app.set("views","./views");

**// post truyền tham số**

var bodyParser = require('body-parser');

var urlencodedParser = bodyParser.urlencoded({ extended: false });

**// su dung multer module**

var multer = require("multer");

var mongoose = require("mongoose");

**// connect db mongodb trong đó cookies là tên của db**

mongoose.connect('mongodb://localhost/cookies');

var dbMongo = mongoose.connection;

**// lắng nge kết nối db có 2 trường hợp connect : 1. Err; 2. connect**

dbMongo.on('err',console.error.bind(console,'connect error'));

dbMongo.once('onpen', function(){

console.log("Mongodb connect");

});

**// tạo schema , mỗi schema sẽ match với collection trong mongodb xác định mỗi document trong collection**

var Schema = new mongoose.Schema({

type : String,

banh : [{

name : String,

image : String,

mota : String

}]

});

**// sử dụng csdl sẽ gọi product**

var product = mongoose.model('sanpham', Schema);

**// upload**

var storage = multer.diskStorage({

// bo vao folder uploads

destination : function(req, file, cb){

cb(null,'./public/uploads')

},

**// dat ten file theo file gốc của image**

filename : function(req, file, cb){

cb(false, file.originalname)

}

});

**// single file upload len**

var upload = multer({

storage: storage

}).single('file');

**// chuyen den produc\_types cho post nguoi dung nhap vao để đưa vào db**

app.post("/product\_types",urlencodedParser, function(req, res){

upload(req, res, function(err){

**// kiem tra err**

if(err){

console.log(err);

res.send("Ket noi that bai")

}

**// insert csdl vao db**

else{

**// theo dan json trong ten : loai banh nguoi dung nhap vao; image: upload len; mota: do người dùng nhập vào**

var sp = {"name": req.body.ten, "image": req.file.originalname, "mota": req.body.mota};

**// sẽ đc upload qua trường product đã gọi trên**

**// trong đó type do người dùng chọn, để insert "banh"**

product.findOneAndUpdate({type: req.body.loai}, {$push:{banh:sp}}, function(err, result){

console.log(result);

});

**// xuất ra bên các loại bánh cakes, cookies, cupcakes, macarons**

res.render("type");

}

});

});

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

res.render("form");

});

## Step 2: xây dựng row cho từng loại sản phẩm add them step 1

**// hàm global cho /public**

app.use(express.static(\_\_dirname + '/public'));

//xây dựng row cho từng loại sản phẩm

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

**// du lieu trong 1 loai sẽ nằm trong cái filter là bánh**

var mang = [];

**// lấy ra các sp của 1 loại**

product.find({type:"cakes"}, function(err, result){

result.forEach( function(sp) {

mang = sp.banh;

});

console.log(mang);

res.render("sp", {mang});

});

});

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

**// du lieu trong 1 loai sẽ nằm trong cái filter là bánh**

var mang = [];

**// lấy ra các sp của 1 loại**

product.find({type:"cupcakes"}, function(err, result){

result.forEach( function(sp) {

mang = sp.banh;

});

console.log(mang);

res.render("sp", {mang});

});

});

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

**// du lieu trong 1 loai sẽ nằm trong cái filter là bánh**

var mang = [];

**// lấy ra các sp của 1 loại**

product.find({type:"macarons"}, function(err, result){

result.forEach( function(sp) {

mang = sp.banh;

});

console.log(mang);

res.render("sp", {mang});

});

});

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

**// du lieu trong 1 loai sẽ nằm trong cái filter là bánh**

var mang = [];

**// lấy ra các sp của 1 loại**

product.find({type:"cookies"}, function(err, result){

result.forEach( function(sp) {

mang = sp.banh;

});

console.log(mang);

res.render("sp", {mang});

});

});

## Step 3: form.ejs

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Upload</title>

</head>

<body>

**<!-- them thuoc tinh multipart/form-data dể dùng cho multer ko thì ko xử lý-->**

<h1>ADD BANH VAO DB VA MO TA </h1>

<form action="http://localhost:3000/product\_types" method="POST" enctype="multipart/form-data">

<select name="loai">

**<!-- cupcakes, cakes, macarons, cookies lấy trong db-->**

<option value="0">--chon loai banh--</option>

<option value="cupcakes">CUPCAKES</option>

<option value="cakes">CAKES</option>

<option value="macarons">MACARONS</option>

<option value="cookies">COOKIES</option>

</select>

<br>

<input type="text" name="ten" placeholder="Nhập tên bánh"></br>

<input type="file" name="file"></br>

<textarea name="mota"></textarea></br>

<input type="submit" value="upload">

</form>

</body>

</html>

## Step4: type.ejs

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Upload</title>

</head>

<body>

<a href="http://localhost:3000/cakes">CAKES</a></br>

<a href="http://localhost:3000/cupcakes"/>CUPCAKES</a></br>

<a href="http://localhost:3000/macarons">MACARONS</a></br>

<a href="http://localhost:3000/cookies">COOKIES</a>

</body>

</html>

## Step 5: sp.ejs

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Upload</title>

</head>

<body>

<% mang.forEach(function(item){ %>

<div>

<div style="color: red">

**<!-- lay ra phan tu image phai dung dau bang <%= item.name%>-->**

<%= item.mota%>

</div>

**<!-- lay ra phan tu image phai dung dau bang <%= item.image%>-->**

<img width="300px" height="300px" src="/uploads/<%= item.image%>" alt="">

<div style="color: blue">

**<!-- lay ra phan tu image phai dung dau bang <%= item.name%>-->**

<%= item.name%>

</div>

</div>

<% }); %>

</body>

</html>

# \*\*\*\* Loginapp\*\*\*\*

## Npm int : tạo project ở dạng package json

{

"name": "loginapp",

"version": "1.0.0",

"description": "",

"main": "app.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"author": "Hieutran",

"license": "ISC",

"dependencies": {

"bcryptjs": "\*", // password băm

"body-parser": "\*",

"connect-flash": "\*", // flash message

"cookie-parser": "^1.4.1",

"express": "\*",

"express-handlebars": "\*", // our view engine

"express-messages": "\*", // flash messaging express

"express-session": "\*", // flash messaging express

"express-validator": "\*", // validate

"mongodb": "\*",

"mongoose": "\*",

"passport": "\*",

"passport-http": "\*", // password fb, youtube ..

"passport-local": "\*"

}

}

## Npm install : để chạy package json

Mở mongod 🡺 mongo

show dbs : xem thử đã có db tồn tại chưa

use loginapp : dung db loginapp

db.dropDatabase(); : xóa db

db.createCollection('users'); : tạo collection tên là users

show collections