# SSR & CSR

SSR (sever side rendering):

* ra đầy đủ src code, dễ dàng cho SEO tim kíêm gg.
* Mỗi lần chuyển site phải tại lại toàn bộ trang.
* Thời gian phát triển ứng dụng nhanh hơn CSR.
* Lượt truy cập đầu tiên nhanh hơn CSR.

CSR (Client side rendering):

* ít mã code, khó khăn việc SEO tìm kiếm gg.
* Mỗi lần chuyển site không phải tải lại toàn bộ, chỉ tải lại những cái mới.
* Phải giao tiếp API.

# Cài đặt thư viện cần thiết

Cài đặt **express**  // trước tiên phải cài nodejs trước

B1: cd nameFolder

B2: npm init // khởi tạo dự án (tạo ra file **packasge.json**)

B3: npm i express // tải express (tạo ra file **package-lock.json**, folder node\_modules)

B4: tạo file index.js

B5: vào đường link <https://expressjs.com/en/starter/hello-world.html> để copy code hello world đầu tiên.

B6: node index.js // để run với <http://localhost:3000/> (\*) (cách này bất tiện, cần cài thư viện Nodemon).

B6.1: npm i --save-dev nodemon

B6.2: vào file **packasge.json** vào phần key **‘’script’’** thêm key-value sau:

"start":"nodemon --inspect app.js", // value nodemon –inspect đừng dẫn file .js

B6.3: npm start // để run với <http://localhost:3000/> (\*)

Cài đặt thư viện **Morgan** // để xem thông tin Logger HTTP

B1: npm i -–save-dev morgan

B2: const morgan = require('morgan')

app.use(morgan('combined'))

Cài đặt thư viện **handlebars**

Note: search google ‘***express handlebars’*** để tìm hiểu thêm

B1: npm install express-handlebars

B2: const handlebars = require('express-handlebars')

const path = require('path')

app.engine('hbs', handlebars.engine({extname: '.hbs'}));

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

app.set('views', path.join(\_\_dirname, 'resources', 'views'));

B3: cấu hình theo đúng format layouts (see more: <https://github.com/express-handlebars/express-handlebars>)

B4: tham khảo format layouts đã tự làm trước đó:

<https://github.com/Hoangk69/teamplate-engin>

**Satatic file** (để trong folder public)

app.use(express.static(path.join(\_\_dirname, 'public')))

Path folder: public\img\Capture.PNG

* Truy cập: <http://localhost:3000/img/Capture.PNG>

Cài đặt thư viện **node-sass** (search gg: npm node-sass)

B1: npm install node-sass -–save-dev

B2: Tạo folder file **scss\app.scss** trong folder **resources**. Và tạo folder **css** trong folder **public**

B4: Vào file packge.json thêm cặp “key”:”value” này nào phần “scripts”.

"watch": "node-sass --watch src/resources/scss --output src/public/css"

B5: npm run watch

B6: Viết style cho các thẻ element trong file **app.scss** thì tự tạo ra file **app.css** có style y hệt.

B7: <link rel="stylesheet" href="css/app.css"> // thêm vào file **main.hbs**

Tham khảo code đã làm trước đó: <https://github.com/Hoangk69/template-node-sass>

# Query parameters

Thường hiển thị chi tiết lên thanh URL, phần lớn sử dụng method **get**.

[http://localhost:3000/search**?q=laptrinh&name=f8**](http://localhost:3000/search?q=laptrinh&name=f8)

**?key1=value1&key2=value2...&keyn=valuen**

*Đối với method get*

console.log(req.query) // { q: 'laptrinh', name: 'f8' }

console.log(req.query.q) // laptrinh

*Đối với method post*

console.log(req.body) // { q: 'laptrinh', name: 'f8' }

console.log(req.body.q) // laptrinh

\*để hiện thị đc với method post trước khi lấy data body thì cần thêm:

app.use(express.urlencoded({extended: true}));

app.use(express.json());

// reading more

Sự khác biệt giữa req.query và req.params:

|  |  |  |
| --- | --- | --- |
|  | req.query | req.params |
| Định nghĩa router | https://localhost:3000/user?userId=5896544 | https://localhost:3000/user/:userId |
| Truyền params |  | https://localhost:3000/user/5896544 |
| Sau khi in ra | {userId : 5896544} | {userId : 5896544} |
| Nhận biết | Sau dấu ?key=value&key=value... | :slug |

# Form Default behavior

Mặc định form ko truyền method thì mặc định là method get, ko truyền action mặc định là tại form đó.

Sẽ gửi req theo thuộc tính name:

<input type="text" name="q" class="form-control" >

# Mô hình MVC (Model-View-Controller)

Cấu hình router-controller tham khảo link sau:

<https://github.com/Hoangk69/config-router-controller.git>

Cài đặt thư viện **mongoose**:

B1: npm install mongoose

B2: trong folder src tạo folder file config/db/index.js

B3: thêm const mongoose = require('mongoose'); vào file vừa tạo.

B4: tạo hàm để connect.

B5: đọc thêm doc: <https://www.npmjs.com/package/mongoose> phần connect to db.

B6: **Defining a Model** (định nghĩa một model).

Tạo foloder file modles/Course.js trong folder app.

Export xem trong mục **Accessing a Model** của doc <https://www.npmjs.com/package/mongoose>

# CRUD

Read more: <https://mongoosejs.com>

1. Hiển thị list khóa học (sử dụng hàm find({},callback) để lấy ra tất cả các *document/bản ghi* trong môt *collection/bảng*. )

// by callback

Model.find({}, function(err, datas){

If(!err){return res.jon(datas)}

Else{next(err)}

})

// by promise

Model.find({})

.then(datas => res.json(datas))

.catch(next)

1. Lấy một bản ghi (hiển thị detail sử dụng hàm findOne({key: value}, callback)).
2. Creat một document/bản ghi (xem thêm phần **Constructing Documents** của tài liệu).

More: cài đặt thêm **mongoose-slug-generator**

1. Update (method PUT).

Cần cài thêm thư viện hỗ trợ method PUT

B1: npm install method-override

B2: const methodOverride = require('method-override')

B3: **override using a query value** (đọc doc phần này tại <http://expressjs.com/en/resources/middleware/method-override.html>)