**BÀI THỰC HÀNH 4**

**Làm quen NodeJS, ExpressJS**

## Cài đặt:

Git: <https://git-scm.com>

NodeJS: <https://nodejs.org/en/download/>

MongoDB: <https://www.mongodb.com/download-center?jmp=nav#community>

Code Editor:

* SublimeText: <https://www.sublimetext.com/3>
* Atom: <https://atom.io>

Google Chrome extension:

- Postman: <https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop?hl=vi>

- Or ARC:

<https://chrome.google.com/webstore/detail/advanced-rest-client/hgmloofddffdnphfgcellkdfbfbjeloo>

*\* Lưu ý: Các bạn chuẩn bị cài đặt các phần mềm trên trước khi đến lớp thực hành, hướng dẫn cài đặt có sẵn trong các link hãy chọn đúng phiên bản phù hợp với OS mà mình đang sử dụng*

## Bắt đầu làm quen

Sau khi đã cài đặt xong các Software cần thiết:

Mở code editor mà các bạn đã cài đặt:

**Save lại tên bt1.js:**

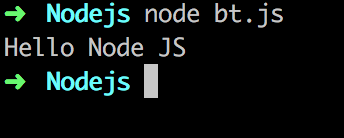


**Sau đó mở console lên:**

Đối với Window thì sau khi đã cài Git-scm thì chuột phải ngay tại thư mục lưu file bt1.js vừa tạo, chọn Open Git bash here.

Gõ lệnh:

$ node bt1.js

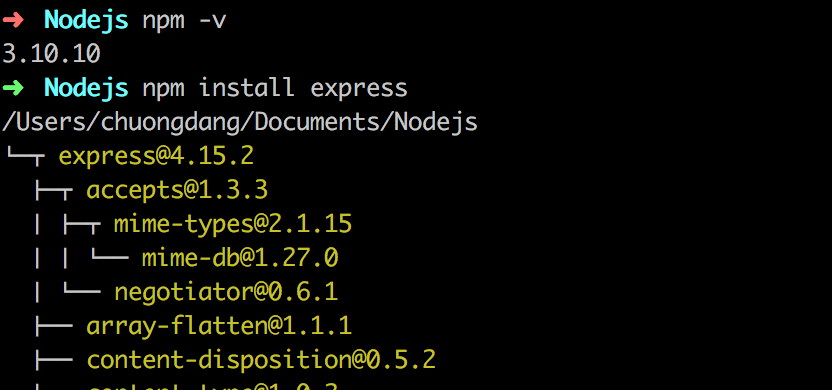


Ngoài ra **NodeJS** còn có khả năng xử lý tương tự như console các bạn có thể gõ lệnh *node* để vào console của nodeJS



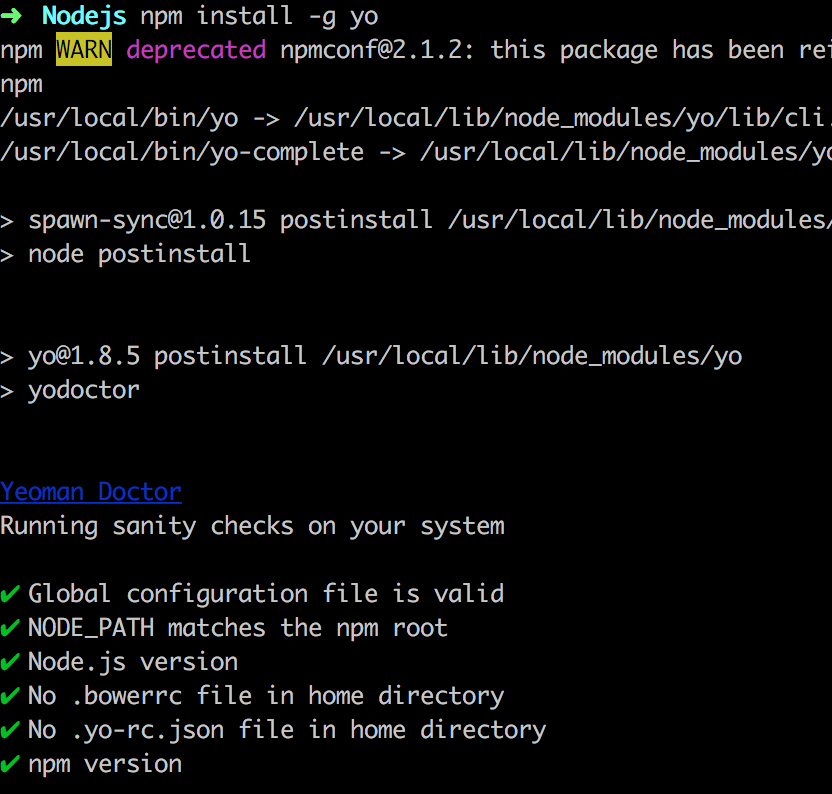
## Node Package Manager & Yeoman

NodeJS đi kèm với một package manager (**npm)** cho phép cài đặt & quản lý các gói thư viện (module theo định nghĩa của nodejs) một cách tiện lợi và nhanh chóng. Tiếp tục với giao diện console để install một module.



Mặc định module sẽ được cài đặt theo mode ***local*** ngay tại thư mục hiện tại, các bạn có thể kiểm tra thư mục hiện tại xem có thư mục ***node\_modules****,* việc này hạn chế chỉ ứng dụng (source code \*.js) được việc trong thư mục hiện hành mới gọi được những *node modules này*. Nhưng một vài trường hợp các bạn sẽ muốn cài đặt những phần mềm ***global*** để bất kì ứng dụng nodeJS nào các bạn viết đều sử dụng được nó.

Cài đặt **Yeoman** ở mode global thêm tham số -g:



**Yeoman** là một app generator, cho phép bạn tạo nhanh nhiều template source code project có liên quan tới các công nghệ phát triển Web như nodeJS-express-mongodb stack, boostrap-nodejs-express-angular và rất nhiều bộ template khác có thể tham khảo ở đây: <http://yeoman.io/generators/>

**Bower:** là package manager cho các Framework Web app, một tiện ích tương tự npm nhưng với mục tiêu nhỏ hơn là cung cấp các gói thư viện liên quan tới công nghệ web. Cài đặt bằng **npm:**

**$ npm install -g bower**

## Ứng dụng Web server với expressJS

ExpressJS là một Web Application Framework đơn giản viết bằng nodeJS nhằm hỗ trợ viết ứng dụng web một cách nhanh chóng và dễ dàng nhất:

Sử dụng npm cài đặt các gói sau:

**$ npm install body-parser -g**

**$ npm install cookie-parser -g**

**$ npm install multer -g**

Tạo 1 file bt2.js với nội dung như sau:

var express = require('express');

var app = express();

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

res.send('Hello World');

})

var server = app.listen(8081, function () {

var host = server.address().address

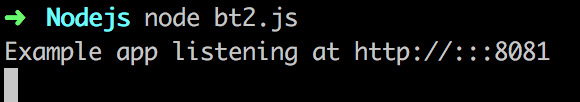
var port = server.address().port

console.log("Example app listening at http://%s:%s", host, port)

})

Save lại và chạy lệnh:

**$ node bt2.js**



Mở trình duyệt localhost:8081 và xem kết quả,

Tiếp tục với bt3.js với nội dung như sau:

var express = require('express');

var app = express();

// This responds with "Hello World" on the homepage

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

console.log("Got a GET request for the homepage");

res.send('Hello GET');

})

// This responds a POST request for the homepage

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

console.log("Got a POST request for the homepage");

res.send('Hello POST');

})

// This responds a DELETE request for the /del\_user page.

app.delete('/del\_user', function (req, res) {

console.log("Got a DELETE request for /del\_user");

res.send('Hello DELETE');

})

// This responds a GET request for the /list\_user page.

app.get('/list\_user', function (req, res) {

console.log("Got a GET request for /list\_user");

res.send('Page Listing');

})

// This responds a GET request for abcd, abxcd, ab123cd, and so on

app.get('/ab\*cd', function(req, res) {

console.log("Got a GET request for /ab\*cd");

res.send('Page Pattern Match');

})

var server = app.listen(8081, function () {

var host = server.address().address

var port = server.address().port

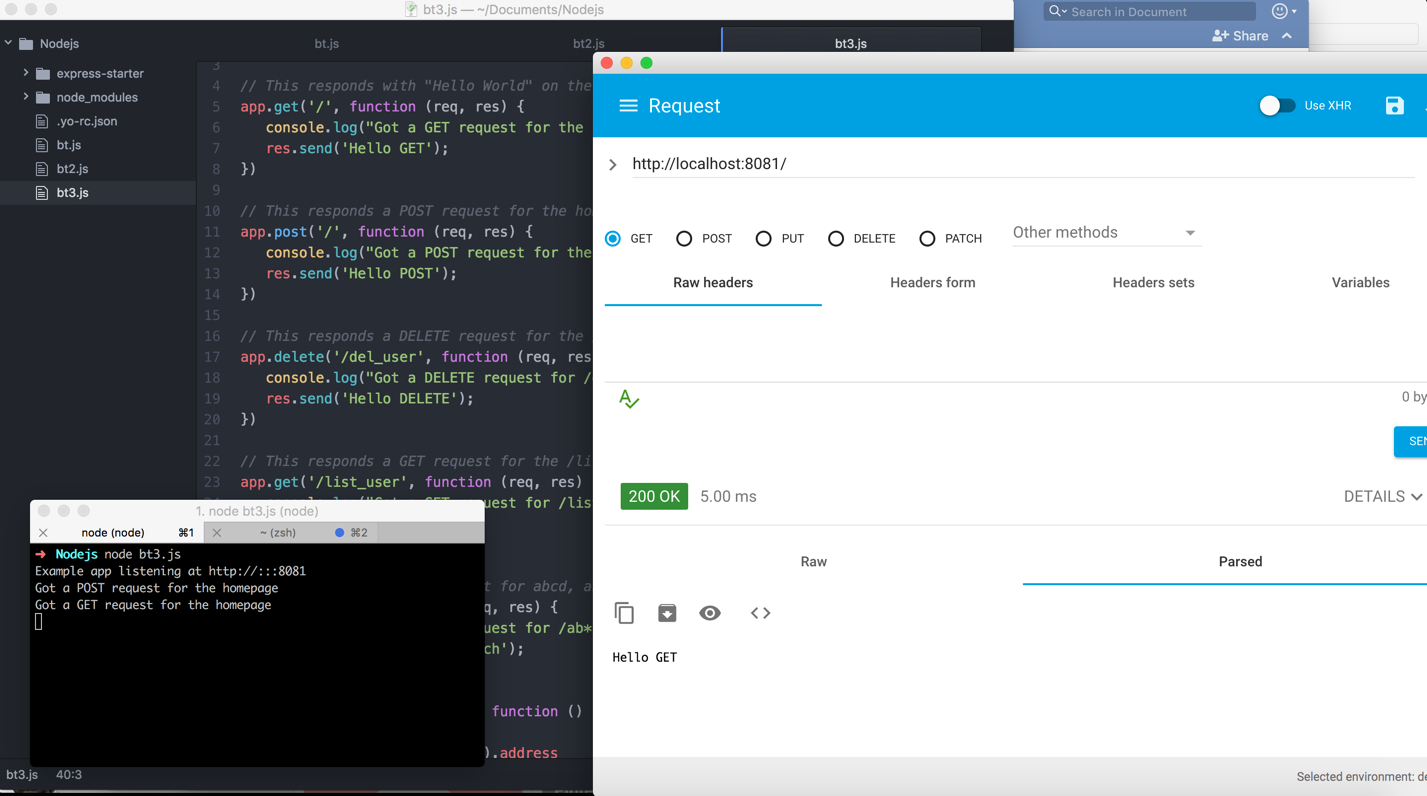
console.log("Example app listening at http://%s:%s", host, port)

})

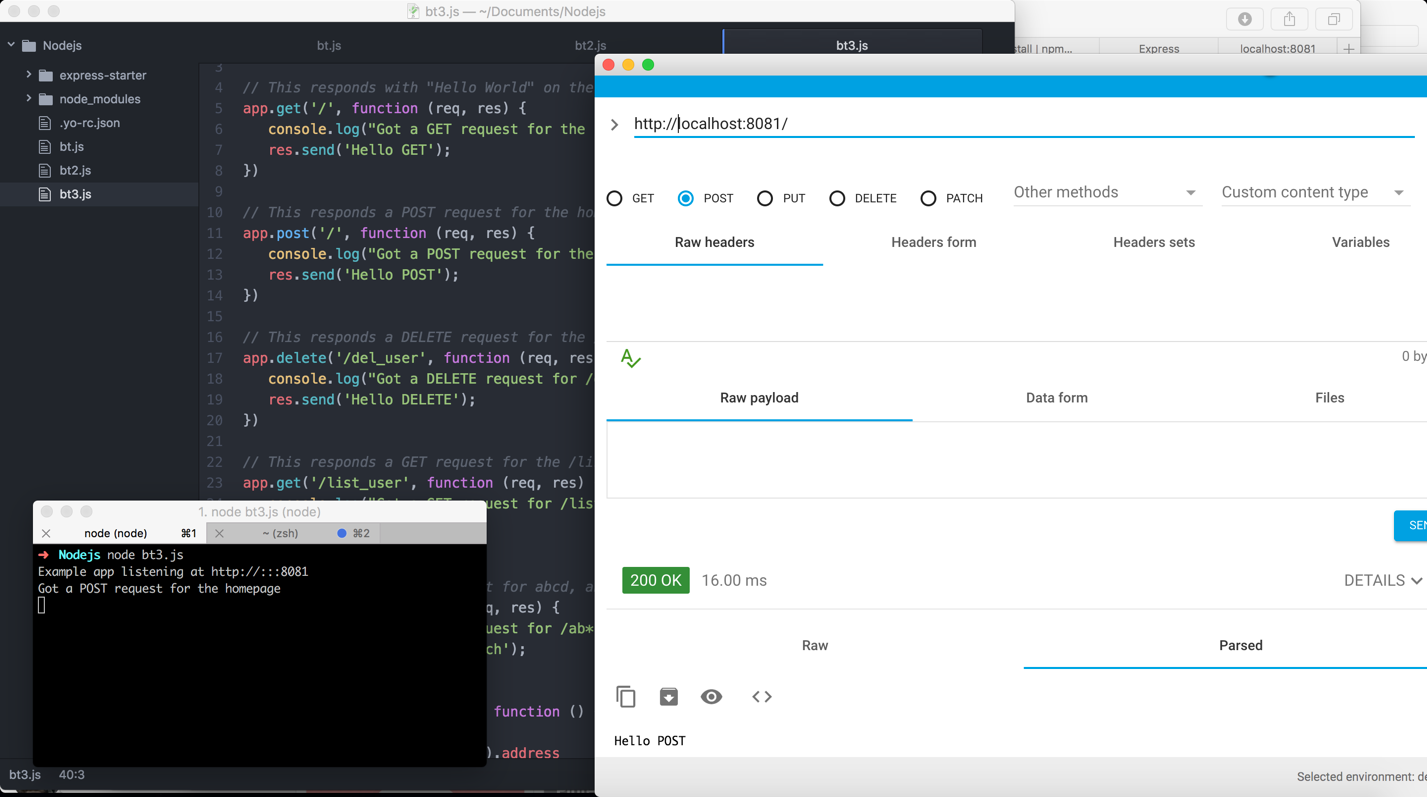
Để kiểm tra các Web method request và route trên có trả về đúng không, chạy ứng dụng bằng lệnh:

**$ node bt3.js**

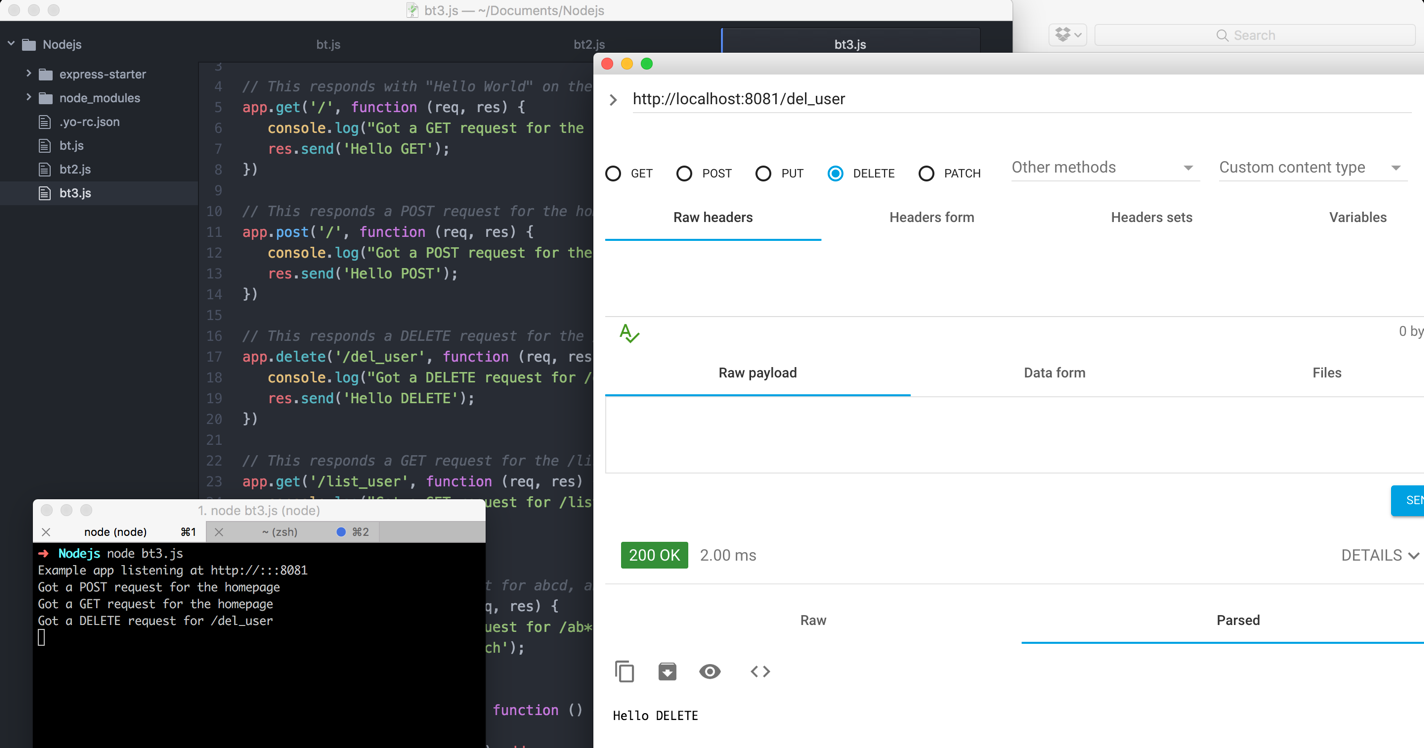
**Test GET /**

****

**Test POST /**



**Test DEL /del\_user**



Thực hành tiếp các ví dụ còn lại:

Tham khảo thêm:

<https://www.tutorialspoint.com/nodejs/nodejs_express_framework.htm>

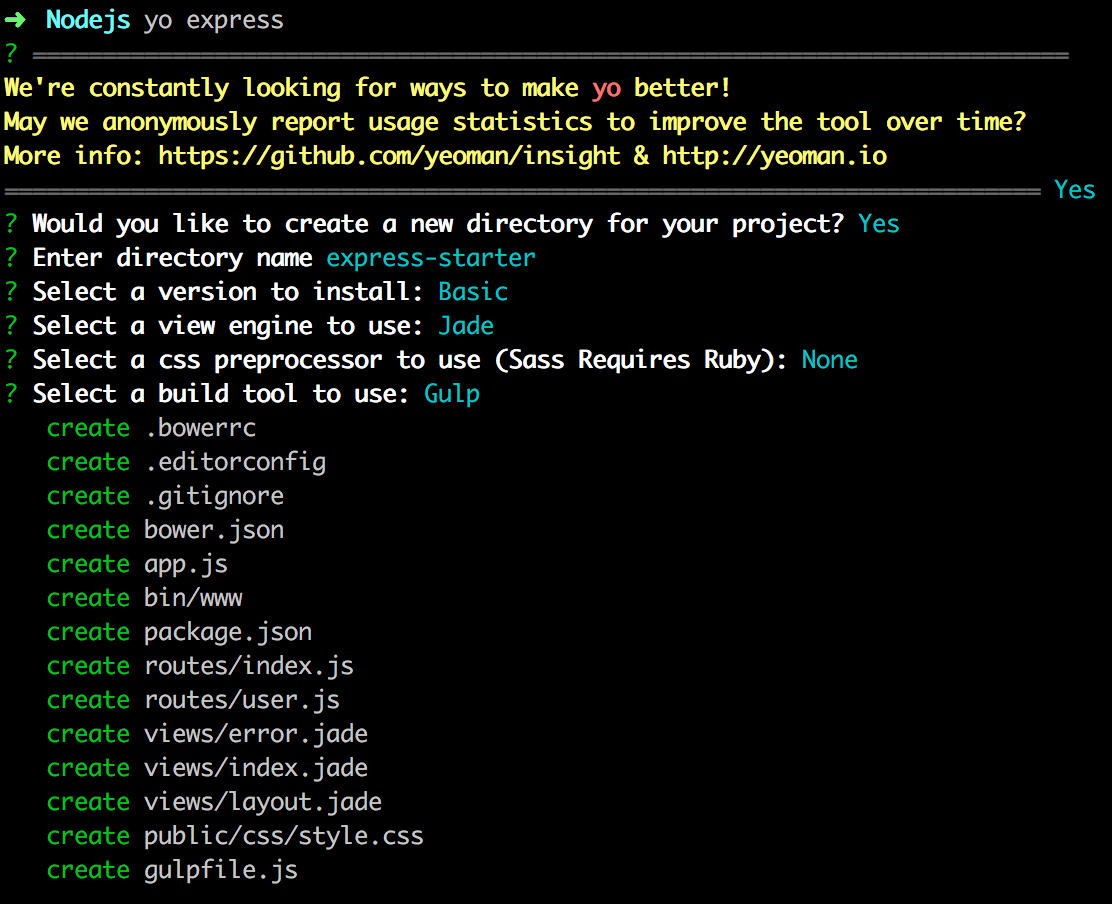
## Sử dụng Yeoman, Bower để tạo ra một khung project đơn giản:

Cài đặt express-generator sử dụng **npm**:

*$ npm install -g generator-express*

Sử dụng yeoman để tạo mẫu project sử dụng ExpressJS:

*$ yo express*

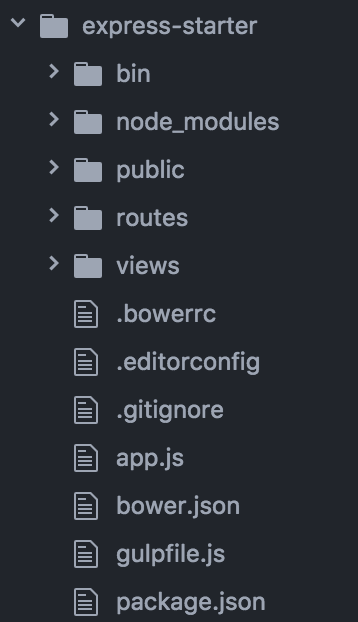


Install build tool Gulp:

**$ npm install -g gulp**

Sau khi tạo xong các bạn chuyển vào tên thư mục các bạn vừa nhận cho project ở đây là “express-starter”, gõ :

Giải thích cấu trúc thư mục:



**Node\_modules:** các module của nodejs

**Route:** chứa định nghĩa về các route/đường url của ứng dụng web này

**Views:** chứa các file để generate ra các file html cho client

**Public:** chứa các file css, js phía client

**Package.json:** chứa định nghĩa về các module được sử dụng trong ứng dụng, khi gõ “npm install” tại thư mục hiện hành sẽ install tất cả các module được khai báo trong package.json

**Gulpfile.js/Grunt.js:** Build runtime để chạy/test/deploy ứng dụng node.

## Yêu cầu

Sử dụng nodeJS, Boostrap, JavaScript ở phía client (jQuery) để xây dựng một trang login/signup đơn giản (Không cần sử dụng database) dữ liệu có thể được hardcoded trong source hoặc chưa trong file \*.json ở server. Cụ thể ứng dụng sẽ có các View sau:

**Route: /login**

Sẽ hiện thị một khung cho người dụng nhập username (dạng email) /password đã tạo và 2 nút login, signup (nếu chưa có tài khoản), thông tin nhập vào gửi lên server theo phương thức POST, server nhận được sẽ kiểm tra xem có đúng thông tin chưa và trả về Response dạng JSON cảnh báo nhập sai username/password nếu sai, nếu đúng người dùng sẽ được redirect đến route / với một message “Welcome <tên\_user>”

**Route: /signup**

Sẽ có 3 khung username(dạng email)/password/confirmed password, thông tin nhập vào khi nhấn nút signup sẽ được gửi lên expressJS server theo phương thức POST, server sẽ lưu lại tài khoản mới bằng 1 biến trong JS hoặc bằng file \*.json

**Bắt buộc: validate format email ở phía server.**

**Điểm cộng:** validate format email và kiểm tra so khớp 2 mật khẩu có khác nhau lúc signup và login ở phía client và cảnh báo người dùng (gợi ý: alert(“…”)) trước khi gửi lên server.

Các view html phải được xây dựng bằng Bootstrap

Gợi ý: các bạn có thể sử dụng generator

<https://www.npmjs.com/package/node-express-bootstrap3-template>

<https://www.npmjs.com/package/generator-jquery-bootstrap-simple>

## Tham khảo và thực hành thêm

RestfulAPI: <https://www.tutorialspoint.com/nodejs/nodejs_restful_api.htm>

Thực hành: <https://github.com/maxogden/art-of-node>