

บทที่ 7 (crash course) เริ่มต้นกับ vuejs

ทำไมต้อง vuejs

vuejs เป็น javascript framework เอาไว้ทำหน้าบ้าน หรือ Front End ตัวนึง เมื่อ่อน react และ angularJS แล้วทำไมถึงต้องเป็น vuejs ไม่เป็นตัวอื่น ผมตอบว่า มันเล็ก มันไม่กินทรัพยากร เครื่องมากนัก syntax ของมัน มีการแทรก data ไปใน tag HTML darmณ์คล้าย PHP ไม่เขียน syntax วุ่นวายเหมือน React ทำให้ง่าย ต่อการเรียนรู้และเข้าใจได้มากกว่า

work หรือ คนใช้ react เยอะนะครับ ยื่ห้อ facebook ด้วยอีก

องกรค ระดับโลกเค้าก็ใช้กันนะครับ เช่น Alibaba, Adobe, IBM, NASA, Nintendo, Trivago และอีกเยอะครับ แต่ผมไม่ได้สนใจว่าใครจะใช้นะครับ ผมสนใจว่า feature ที่มันมีตอบโจทย์ งานที่เราจะทำหรือเปล่า แค่นั้นเอง สำหรับผม vuejs ตอบโจทย์มากครับ ตัวอย่างเช่นผมเขียน webblog ของผมด้วย PHP เป็นเดือน ๆ แต่ผมใช้ vuejs + nodejs ทำเว็บໄว์ต gooddev.me ของผมเสร็จใน 10 วัน ผม apply ไปทำ web อื่นอีก ใช้เวลาเพียงแค่ 4-5 วันเองครับ จะมี เหตุผลอะไร ที่จะไม่ใช้ vuejs

การใช้งานผ่าน CDN

เราสามารถใช้งาน vuejs CDN ได้เหมือน jQuery เลยครับ สำหรับคนที่ไม่ทำการใช้งานแบบ CDN คืออะไร การใช้งานหรือติดตั้งแบบ CDN คือ เอา Code ของ library หรือ framework ที่เราใช้มาแปะครับ เช่น

```
<script src="https://cdn.jsdelivr.net/npm/vue"></script>
```

พอແປແລ້ວ ເຮັດຈະການເຂົ້າມ ເພື່ອໃຊ້ງານ vuejs ໄດ້ທັນທີເລຍ ສໍາຮັບ CDN ຂ້າງບົນ ດືວ່າ ສໍາຮັບ Development ອີ່ວິ້ວ ຕອນພັດນາ ສ່ວນສໍາຮັບ Production ອີ່ວິ້ວ ໃຊ້ງານບົນ Server ເຄົາໃຫ້ໃຊ້ຕົວນີ້

```
<script  
src="https://cdn.jsdelivr.net/npm/vue@2.6.7/dist/vue.js"></script>
```

ແຕ່ເອາຄວາມຈິງເລຍຄັບ ມີໄມ່ເຄີຍໃຊ້ທັງສອງຕົວ ມີໃຊ້ແຕ່ webpack ຄັບ

ເຮັມສ້າງ project vuejs ໂດຍໃຊ້ webpack

ການສ້າງ project vuejs ໂດຍໃຊ້ webpack ອີ່ວິ້ວ ມີເຮົາກວ່າ vue-cli ນັ້ນ ຊັດຕືບອັນ ດືວ່າ ຈັດການໂປຣເຈຄຂອງເຮົາເປັນໄດ້ຢ່າງຮະບນ ແລະ ຕິດຕັ້ງ library ອີ່ວິ້ວ package ທີ່ຈໍາເປັນມາໃໝ່ ເວລາເຮົາພັດນາໂປຣແກຣມ ໄມຕ້ອນມາຄອຍກົດ refresh ພອກດ save ຮະບນທີ່ເຮົາພັດນາ ອີ່ວິ້ວ Front End ຂອງເຮົາ ຈະ update ໃຫ້ທັນທີ ລວມຖິກການສັ່ງ build ເພື່ອ deploy ອີ່ວິ້ວຕິດຕັ້ງບົນ Server ຈິງ ໄດ້ຢ່າງ ປັບ config ຕ່າງໄດ້ຢ່າຍ

ທຳການຕິດຕັ້ງ vue-cli ກັນກ່ອນເລຍຄັບ ເປີດ Terminal ຂອງເຮົາຂຶ້ນມາ ໄມຕ້ອນກັງລ ເຊິ່ງໂຟລ්ເດວරනະຄັບ ເພົ່າການຕິດຕັ້ງແບນ -g ດືວ່າ gobal ອີ່ວິ້ວຕິດຕັ້ງໃໝ່ເຮົາໃຊ້ງານ ໄດ້ທຸກທີ່ນີ້ເອງ

```
npm install -g vue-cli
```

ທຳການຕິດຕັ້ງເສົ່າງແລ້ວ ເຮັດເຂົ້າໄປທີ່ໂຟල්ເດວර client ຂອງເຮົາກັນຄັບ ເຊັ່ນ

```
d:/NodeRootBook/nv-webblog/client
```

ຈາກນັ້ນທຳການພິມພົມພຳສັ່ງ ເພື່ອສ້າງ project ຂອງເຮົາກັນຄັບ

```
vue init webpack .
```

ໂດຍປົກຕິ “.” ທີ່ຕ່ອງຈາກ webpack ມັກຈະເປັນຊື່ໂປຣເຈຄຂອງເຮົາ ແຕ່ໃນການນີ້ພົມ ຕ້ອງການໃສ່ມັນໄວ້ໃນໂຟල්ເດວර client ນີ້ລະຄັບ ຈະໄດ້ໄມ່ສັບສນ ເມື່ອທຳການຮັນພຳສັ່ງ ເຮັດຈະທຳການປົ່ວໂມງ ຂໍ້ມູນໃໝ່ມັນຕາມກາພຄັບ

```
[korns-MacBook-Pro:~ kornchayapon$ cd noderootbook/nv-webblog
[korns-MacBook-Pro:nv-webblog kornchayapon$ ls
client server
[korns-MacBook-Pro:nv-webblog kornchayapon$ cd client
[korns-MacBook-Pro:client kornchayapon$ vue init webpack .

? Generate project in current directory? Yes
? Project name nv-webblog-client
? Project description nodejs vuejs weblog client
? Author chayapon chandra
? Vue build standalone
? Install vue-router? Yes
? Use ESLint to lint your code? No
? Set up unit tests No
? Setup e2e tests with Nightwatch? No
? Should we run 'npm install' for you after the project has been created? (recommended) npm

vue-cli · Generated "client".
```

ในการพัฒนาติดตั้ง vuejs ด้วย vue-cli เพื่อทำการเขียนโปรแกรมในส่วน Front End ของเรารather ไฟล์เดอร์ client โดยเราต้องเข้าไปที่ไฟล์เดอร์ client ในโปรเจคของเรา ก่อนจะรับ ถึงใช้คำสั่งติดตั้ง vue-cli

จากนั้นก็ทำการใส่ข้อมูลโปรเจคของเราเข้าไป แล้วกด Enter ครับ จะมีส่วน ESLint ซึ่งเป็นแจ้ง warning การเขียนโปรแกรมในรูปแบบของ ES6 ผิดไม่ใช้ เพราะเขียนใหม่ ๆ จะแยกไม่ออกว่าอันไหน error จาก code ผิดพลาด หรือพิมพ์ code ไม่สวยงามตามรูปแบบของ ES6 (ESLint สำหรับผู้ที่ต้องเช็ค code สวยงาม) และ พวก Unit Test ต่าง ๆ ผิดไม่ได้ทำการติดตั้งด้วย เช่นกัน เพราะไม่ค่อยได้ใช้ ถ้าต้องการ UnitTest เราเขียนเองทำงานได้ดีและยืดหยุ่นมากกว่าครับ แต่เราจะไม่ได้เรียนในหนังสือเล่มนี้นะครับ แต่ไม่ต้องห่วงครับ ทำงานเยอะ ๆ จะ Debug เก่งลงครับ และสุดท้ายผมเลือก npm เป็นคำสั่งในการติดตั้ง node package ของเราจะครับ

เมื่อเราทำการติดตั้งเสร็จแล้ว ให้เราทำการพิมพ์

```
npm run dev
```

Terminal ของเราจะรัน และโหลด อญญาติ แล้วจะไปค้างอยู่ที่

```
DONE           Compiled      successfully      in      3002ms
12:03:00

| Your application is running here: http://localhost:8080
```

แสดงว่า ตอนนี้เรารับรู้ว่ามี Vue.js กันแล้วคับ

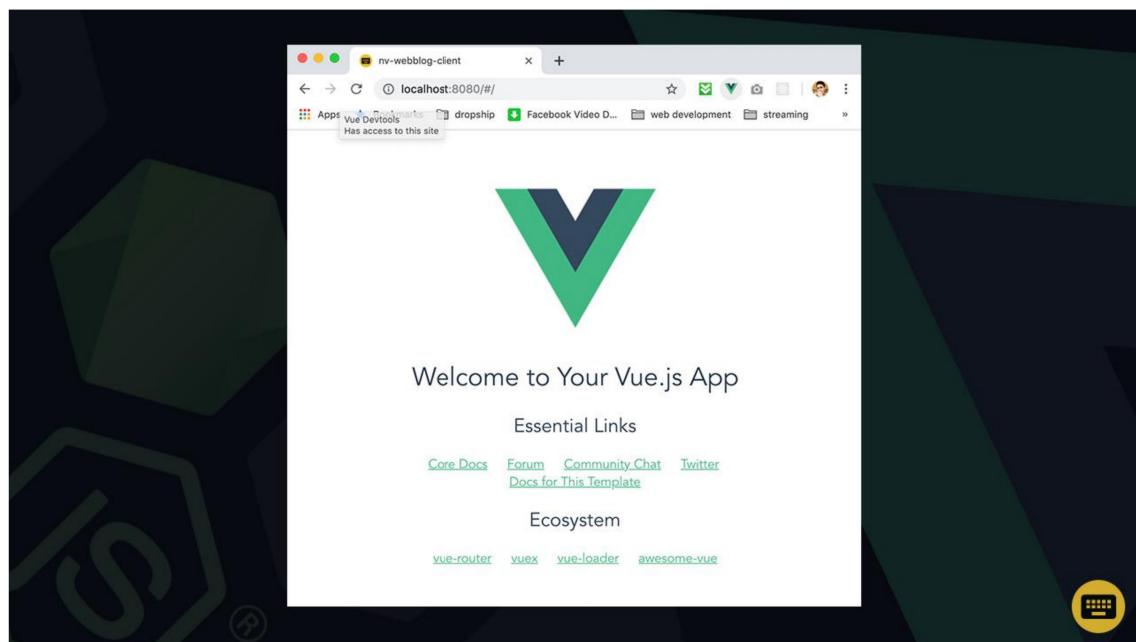
เปิด Browser ของเรามา แล้วทำการ URL ของเราดังนี้

```
http://localhost:8080/
```

Browser ของเราจะถูก Redirect ไปที่

```
http://localhost:8080/#/
```

แล้ว Browser ของเรา แสดงผลดังภาพ นั่นคือ Vue.js ของเราทำงานแล้วนั่นเอง ครับ



หลังจากเราทำการติดตั้ง Vue.js ด้วย vue-cli กันเรียบร้อยแล้ว เราทำการเปิดโปรแกรม VS Code ขึ้นมา และเปิดเข้าที่โฟล์เดอร์ nv-webblog ของเราจะเห็นว่า vue-cli ทำการติดตั้งอะไรมาให้เราอย่างละเอียด เราไม่ต้องไปสนใจทั้งหมดให้ปวดหัวครับ เราสามารถดูได้ในตัวอย่าง ๆ ดังภาพ ส่วนอื่น ๆ เราปล่อยไว้ก่อนครับ

The screenshot shows a code editor with a dark theme. On the left is a file tree for a Vue.js project:

```

client
  build
  config
  node_modules
  src
    assets
    components
      HelloWorld.vue
      router
      index.js
      App.vue
      main.js
    static
  .babelrc
  .editorconfig
  .gitignore
  .postcssrc.js
  index.html
  package-lock.json
  package.json
  README.md

```

The files `index.html`, `main.js`, `App.vue`, and `router/index.js` are highlighted with red boxes. The `App.vue` file is open in the editor, showing its template and script code:

```

3 import Vue from 'vue'
4 import App from './App'
5 import router from './router'
6
7 Vue.config.productionTip = false
8
9 /* eslint-disable no-new */
10 new Vue({
11   el: '#app',
12   router,
13   components: { App },
14   template: '<App/>'
15 })

```

ความสัมพันธ์ระหว่าง `index.html`, `main.js`, `App.vue`, `router` และ `components`

ส่วนประกอบที่สำคัญที่เราสนใจ คือ `index.html`, `main.js`, `App.vue`, `router/index.js` โดย การทำงาน จะเริ่มที่ `index.html` และ `main.js`

เมื่อ `main.js` ทำงานจะทำการสร้าง vue object หรือสร้าง vue ของเราขึ้นมาด้วยคำสั่ง `new` โดยส่ง พารามิเตอร์ หรือ ตัวแปรเข้าไป เพื่อบอกเงื่อนไขในการสร้าง object นั้นเอง โดย

```

el: '#app', -> ให้หนังสือ id="app" ใน index.html แล้ว render
router, -> เซ็ตมื้อยังกับ router ที่โหลดมา
components: { App }, -> ใช้ component นั้นๆไป render
template: '<App/>' -> บอกว่า App คือ component นั้น

```

```

index.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
6     <title>nv-weblog-client</title>
7   </head>
8   <body>
9     <div id="app"></div>
10    <!-- built files will be auto injected -->
11  </body>
12 </html>
13

main.js
1 // The Vue build version to load with the `import` command
2 // (runtime-only or standalone) has been swapped for compatibility with Node
3 import Vue from 'vue'
4 import App from './App'
5 import router from './router'
6
7 Vue.config.productionTip = false
8
9 /* eslint-disable no-new */
10 new Vue({
11   el: '#app',
12   router,
13   components: { App },
14   template: '<App/>'
15 })

```

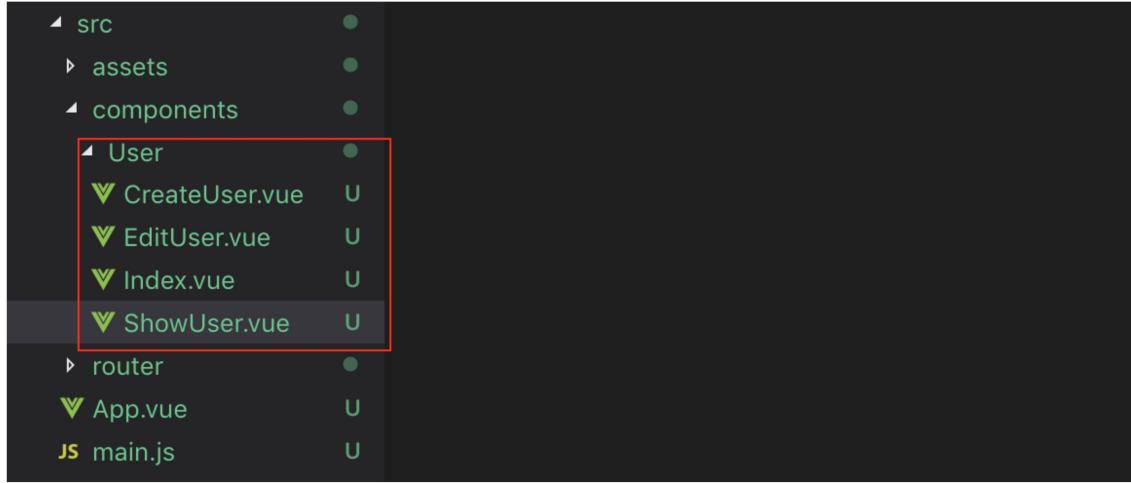
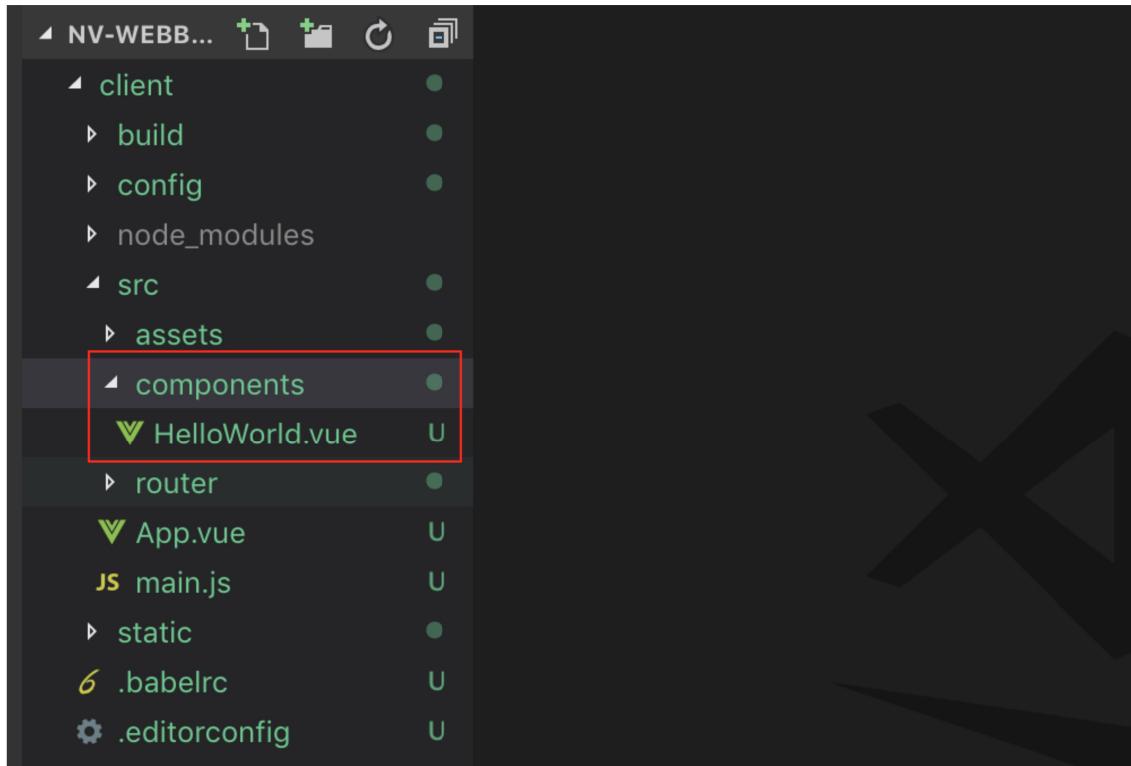
เป็นการบอกว่า App ของเราเป็น Vue Component

ในหัวข้อนี้ผมเพียงอธิบายว่ามันโหลดอะไรมาบ้าง ทำงานตอนไหน และเชื่อมโยงกันอย่างไร ถึงตอนนี้ถ้ายัง งงๆ ยังไม่ต้องไปกังวลนะครับ เข้าใจบ้างไม่เข้าใจบ้างก็ปล่อยผ่านกันไปก่อน เพราะตอนนี้เรายังไม่ได้เริ่มเขียนอะไร์กัน พอเราเขียนๆ ไปจะเริ่มค่อยๆ คุ้นเคยและเข้าใจไปเองครับ เพราะจริงๆ เราคุ้นเคยกับไฟล์เหล่านี้ไม่มาก จะไปแก้ไขมากหน่อยก็ที่ router/index.js เท่านั้นครับ

เริ่มต้นทำ back office

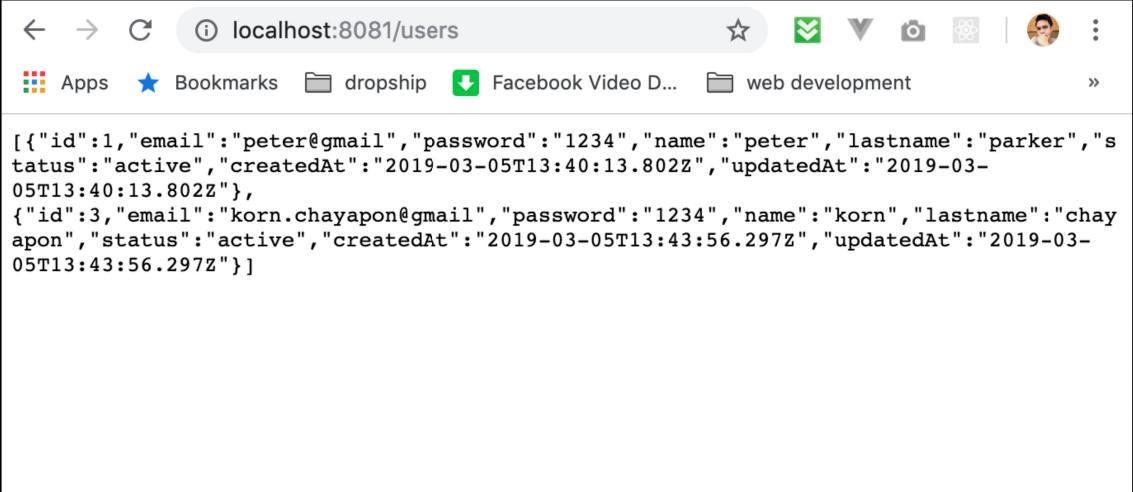
มาถึงตอนนี้เราจะเริ่มต้นทำ Front End ในส่วนของ Back Office เพื่อใช้จัดการ Back End ในส่วนของ Controller กันก่อนเลย

เปิด code ของเราขึ้นมาด้วย VS Code ครับ และเปิดไปที่ folder ตามภาพ ทำการลบ HelloWorld.vue ออก และสร้างโฟล์เดอร์ชื่อ Users ใน โฟล์เดอร์ Components จากนั้นทำการสร้าง Index.vue, CreateUser.vue, EditUser.vue, ShowUser.vue ดังภาพ



เรามาเริ่มกันที่ API ที่เราเขียนนะครับ เราทำการทดสอบ API ของเราโดยการพิมพ์ url ของเราดังนี้ (อย่าลืมรัน node server ของเรา ก่อนนะครับ)

```
http://localhost:8081/users
```



The screenshot shows a browser window with the URL `localhost:8081/users`. The page content is a JSON array of user objects:

```
[{"id":1,"email":"peter@gmail.com","password":"1234","name":"peter","lastname":"parker","status":"active","createdAt":"2019-03-05T13:40:13.802Z","updatedAt":"2019-03-05T13:40:13.802Z"}, {"id":3,"email":"korn.chayapon@gmail.com","password":"1234","name":"korn","lastname":"chayapon","status":"active","createdAt":"2019-03-05T13:43:56.297Z","updatedAt":"2019-03-05T13:43:56.297Z"}]
```

นั่นคือ API ที่เราทำไว้ พร้อมใช้งานแล้วครับ

เราเปิดไฟล์ Client ซึ่งอว่า

```
src/components/Users/Index.vue
```

ขึ้นมาครับ

แล้วทำการใส่ Code ดังนี้ครับ

```
<template>
</template>
<script>
export default {
}
</script>
<style scoped>
</style>
```

เราจำง่าย ๆ code จะแบ่งเป็นสามส่วนคือ template, script และ style

template เราเอาไว้ใส่ code html ของเรา script เราเอาไว้ใส่ vue code ที่เราจะเขียนด้วย javascript ส่วน style คือ css เอาไว้แต่งหน้าทากปาก เว็บไซต์ของเรา scoped คือ ให้ style นี้ใช้ภายใน components นี้เท่านั้น ไม่ไปมีว่ากับคนอื่น

ผมจะใส่ ข้อความไว้ว่า ที่นี่คือ get all user ไว้ก่อนนะครับ เวลาดึง component นี้มาใช้งานจะได้รู้ว่ามาจาก component นี้ครับ ผมเพิ่ม code เข้าไปใน template ด้วยนี่ครับ

```
<template>
  <div>
    <h1>Get All Users</h1>
  </div>
</template>
<script>
export default {

}
</script>
<style scoped>

</style>
```

ผมทำการใส่ Code ใน CreateUser.vue, EditUser.vue และ ShowUser.vue ตามลำดับดังนี้นะครับ

CreateUser.vue

```
<template>
  <div>
    <h1>Create User</h1>
  </div>
</template>
<script>
export default {

}
```

```
</script>
<style scoped>
</style>
```

EditUser.vue

```
<template>
  <div>
    <h1>Edit User</h1>
  </div>
</template>
<script>
export default {

}

</script>
<style scoped>

</style>
```

ShowUser.vue

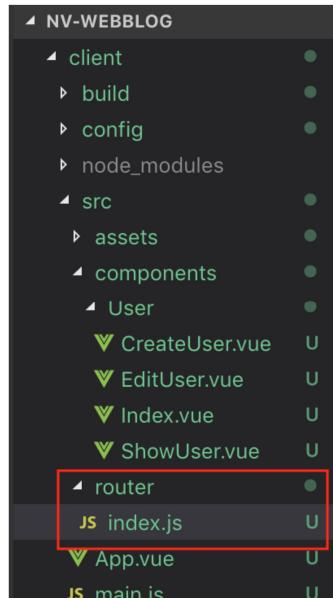
```
<template>
  <div>
    <h1>Get User By Id</h1>
  </div>
</template>
<script>
export default {

}

</script>
<style scoped>

</style>
```

เมื่อเราสร้าง Vue Component ในส่วนของ User ไว้เรียแล้ว ที่นี่เราจะ Link ทุกอย่างเข้าด้วยกันครับ โดยทำการเปิด router/index.js ขึ้นมาครับ



```
2 import Router from 'vue-router'
3 import HelloWorld from '@/components/HelloWorld'
4
5 Vue.use(Router)
6
7 export default new Router({
8   routes: [
9     {
10       path: '/',
11       name: 'HelloWorld',
12       component: HelloWorld
13     }
14   ]
15 })
16
```

จากนั้นทำการแก้ไข code ที่ router/index.js เราเป็นดังนี้ครับ

```
import Vue from 'vue'
import Router from 'vue-router'

// Users
import UserIndex from '@/components/Users/Index'

Vue.use(Router)

export default new Router({
  routes: [
    {
      path: '/users',
      name: 'users',
      component: UserIndex
    }
  ]
})
```

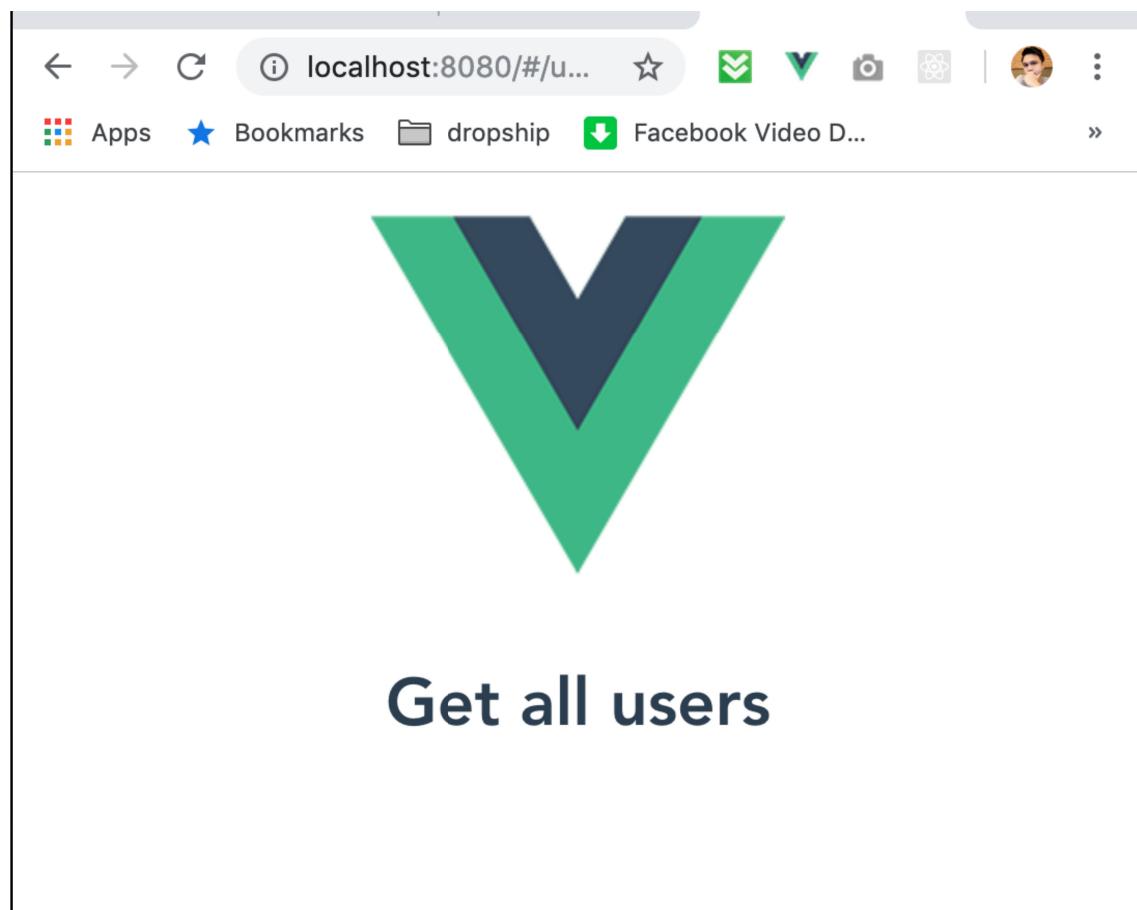
ทำการรัน client ของเราโดยครับ ไม่ลืมใช้มั่ยครับ เข้าไปที่ nv-webblog/client จากนั้น พิมพ์คำสั่ง

```
npm run dev
```

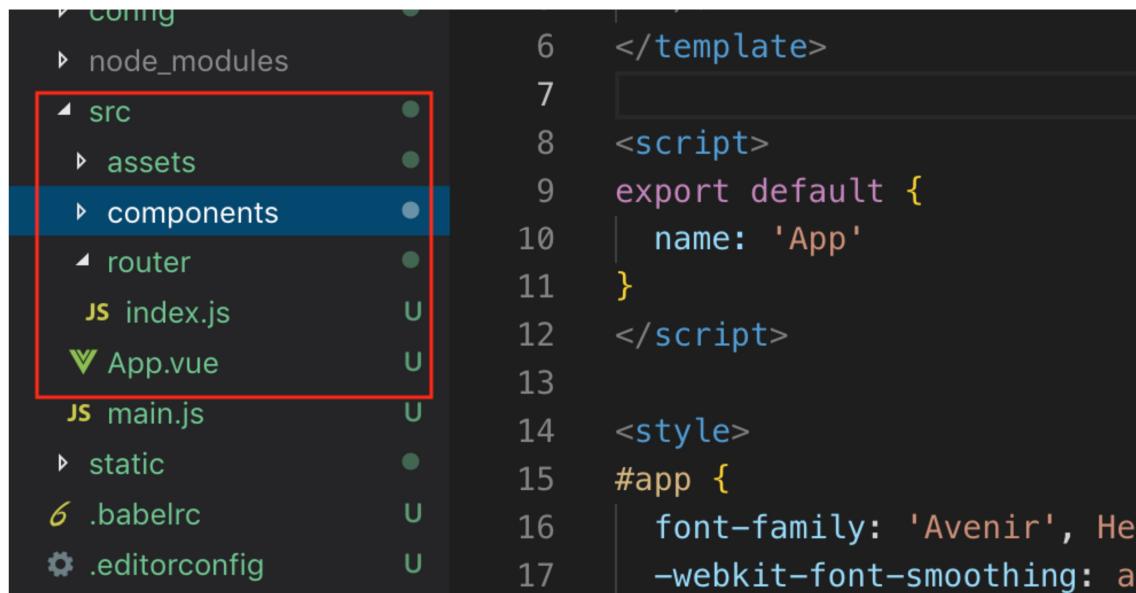
เมื่อ client หรือ front end ของผมทำงานแล้ว ผมเปิด Web Browser ของผมไปที่

```
http://localhost:8080/#/user
```

สังเกตว่าผมรัน Back End ของผมไว้ที่ port 8081 และ Front End ของผมที่ port 8080 ครับ เมื่อผมทำการใช้ Web Browser ของเปิดไปที่ url ด้านบนสิ่งที่ได้ คือ



ก่อนผมจะอธิบาย code ผมจะไปเอา Logo ออกก่อนนะครับ เห็นแล้วชอบหุ่นยนต์
 เพราะเราไม่ได้แสดงผลอย่างที่เราต้องการนั่นเอง เปิด src/App.vue ขึ้นมาครับ



The screenshot shows a file explorer on the left and a code editor on the right. The file explorer displays the project structure:

- config
- node_modules
- src
 - assets
 - components (highlighted with a red box)
 - router
 - index.js
- App.vue
- main.js
- static
- .babelrc
- .editorconfig

The code editor shows the content of App.vue:

```
</template>
<script>
export default {
  name: 'App'
}
</script>

<style>
#app {
  font-family: 'Avenir', Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  color: #2c3e50;
}
```

จากนั้นผมจะลบ Logo และ style ที่เค้าให้มาที่อยากไม่ใช้ออกครับ เหลือไว้เพียง style ที่ผมชอบ โดยผมทำการแก้ code ดังนี้

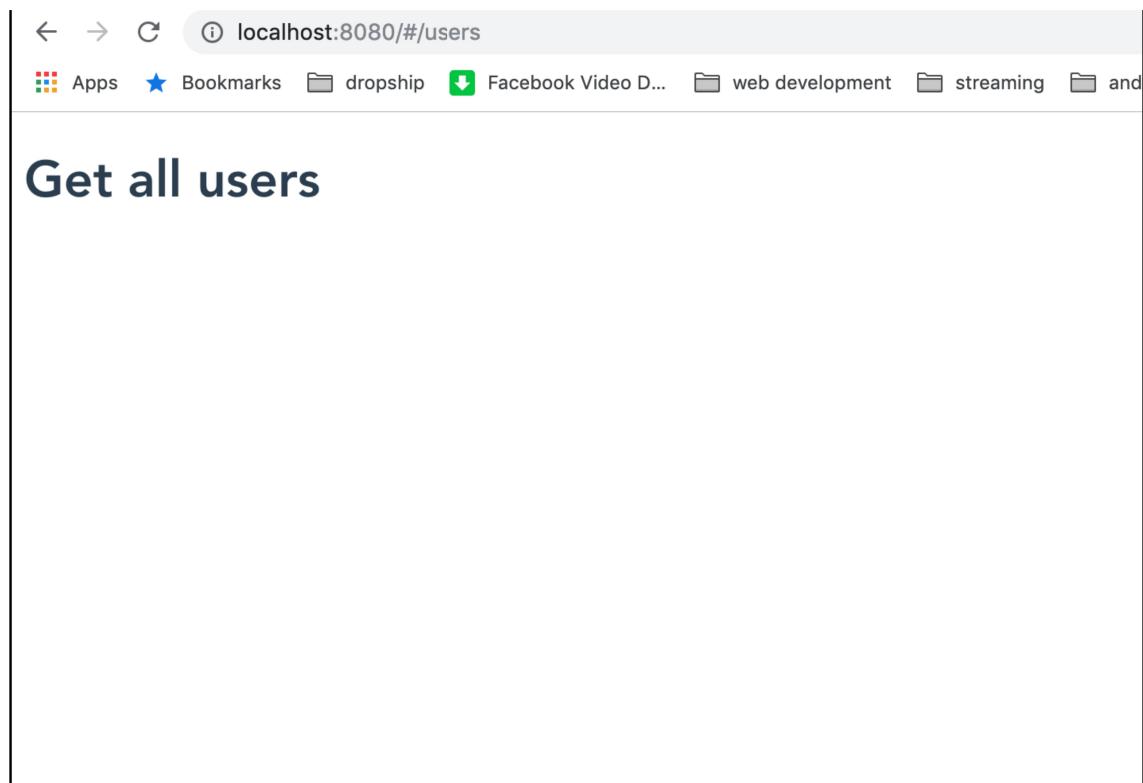
```
<template>
<div id="app">
  <router-view/>
</div>
</template>

<script>
export default {
  name: 'App'
}
</script>

<style>
#app {
  font-family: 'Avenir', Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  color: #2c3e50;
}
```

```
</style>
```

ผลการเปิดผ่าน Web Browser ของผมจะเป็นดังนี้ครับ



เปิด code ในส่วนของ router/index.js ขึ้นมาครับ จะมีสองส่วนที่เราต้องสนใจ คือ

```
import UserIndex from '@/components/Users/Index'
```

คือ โหลด component Users/Index.vue ของเรามาใช้งาน และ มาดูส่วนนี้กันครับ

```
{  
  path: '/users',  
  name: 'users',  
  component: UserIndex  
}
```

path หมายถึง ใช้ที่เรียกต่อจาก url หลัก ซึ่งในที่นี่คือ <http://localhost:8080> นั่นเอง แต่เวลาเรา Deploy หรือเราเอาขึ้น Server เราจะไม่แสดง Port ต่อท้ายนะครับ ไม่ต้องห่วง เช่น <http://sanook.com/users> จะเป็นทำนองนี้ครับ

name ก็คือชื่อ ตอนนี้เรายังไม่ได้ใช้ แค่รู้ว่าเอาไว้ อ้างอิงครับ เสร็จแล้ว เมื่อถูกเรียกมาตาม path และ เราถูกส่ง component ของเราไปให้ครับ

สังเกตว่า ก่อนหน้านี้ เราไปลบ logo ออก มีผลกระทบเว็บไซต์ของเราทันทีเลย นั่นหมายความว่า App.vue มันทำงานร่วมด้วย

ถูกต้องครับ App.vue เป็นตัวแสดง Component ของเราผ่าน <router-view> นั่นเองครับ

```
<router-view>
```

เปิด App.vue ขึ้นมาประกอบนะครับ tag นี้ล่ะครับ จะเป็นตัวเลือกการแสดงผล และ Component จาก path ที่ส่งมา ซึ่งเค้าเรียกว่า vue-router แต่เราไม่ต้องเข้าไปลึกอะไรมันมาก รู้แค่ว่ามันแสดงผลตรงนี้ ส่ง component มาแสดงตรงนี้ตาม path ครับ

จากนั้น เราทำการใส่ path และ component ที่เราเตรียมไว้ทั้งหมดให้ครับ โดยแก้ไข code เป็น ดังนี้ครับ

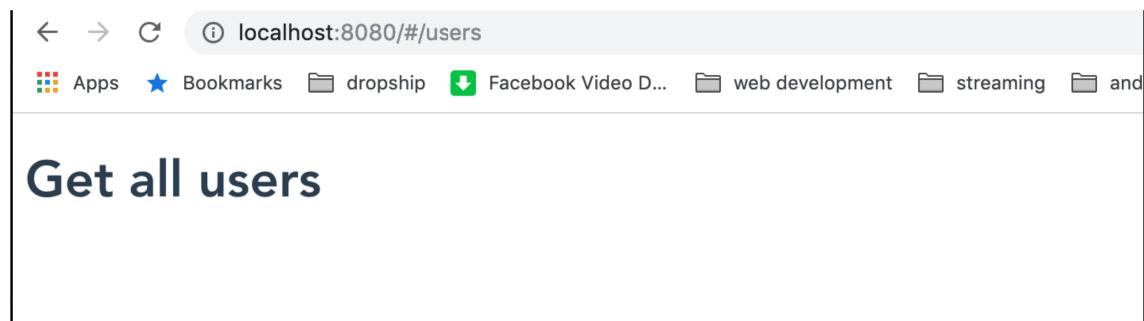
```
import Vue from 'vue'  
import Router from 'vue-router'  
  
// Users  
import UserIndex from '@/components/Users/Index'  
import UserCreate from '@/components/Users/CreateUser'
```

```
import UserEdit from '@/components/Users/EditUser'
import UserShow from '@/components/Users>ShowUser'

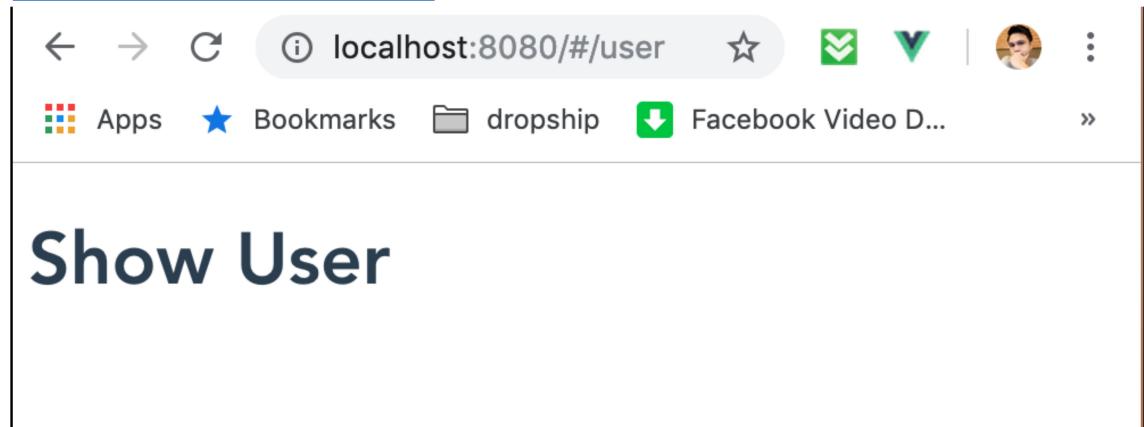
Vue.use(Router)

export default new Router({
  routes: [
    {
      path: '/users',
      name: 'users',
      component: UserIndex
    },
    {
      path: '/user/create',
      name: 'users-create',
      component: UserCreate
    },
    {
      path: '/user/edit',
      name: 'user-edit',
      component: UserEdit
    },
    {
      path: '/user',
      name: 'user',
      component: UserShow
    },
  ]
})
```

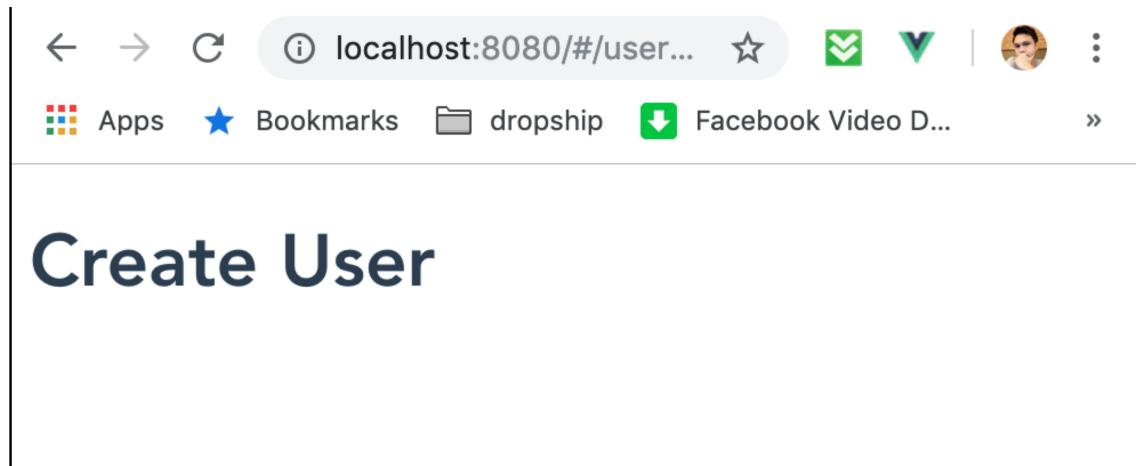
จากนั้นทำการทดสอบไปที่ URL ต่าง ๆ ดังนี้ จากตอนกลับมาตามรูปครับ
<http://localhost:8080/#/users>



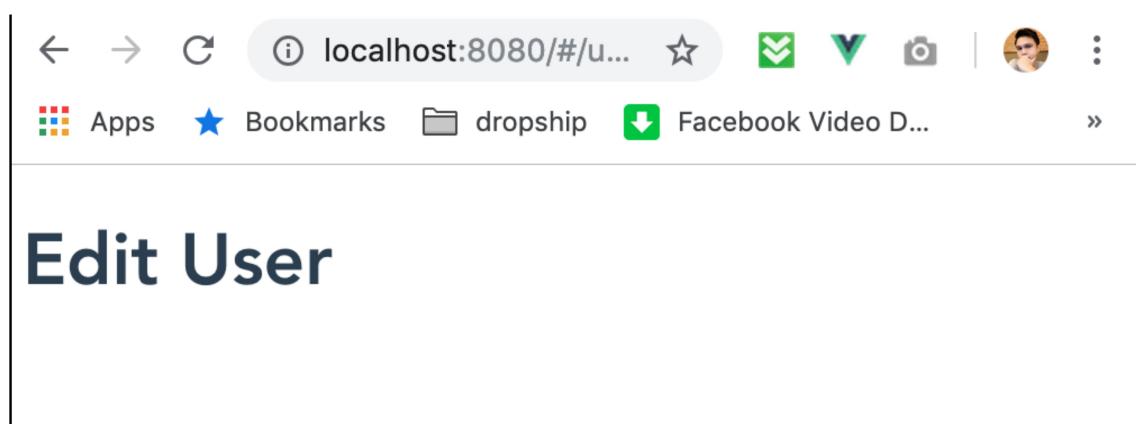
<http://localhost:8080/#/user>



<http://localhost:8080/#/user/create>



<http://localhost:8080/#/user/edit>



มาถึงตอนนี้เรา สามารถสร้าง Vue Component และ สามารถ เรียกใช้งานผ่าน url ที่เราต้องการได้เรียบร้อยแล้วนะครับ ในบทต่อไป เราจะดึงข้อมูล และสั่งงาน Server หรือ Back End ของเราให้ทำงานคำสั่งนะครับ