

Vue Cli 3.xx project

백성애

Vuejs.org

<https://cli.vuejs.org/>

Vue cli 설치

- 개발환경
- [1] node.js 설치해두기
- nodejs.org에서 다운로드 <http://nodejs.org>
- => LTS (안정화버전) Current(최신버전)
- - 10버전 이상 LTS로 설치하자.
- 버전 확인
- Node -v => 10 버전 이상 권함
- Npm -v => 6 버전 이상

Vue project 생성 및 서버 실행

- Windows의 경우
- 콘솔에서
- `npm i -g @vue/cli`
- 친 뒤 아래를 작성하자 (현재의 최신 버전 설치됨)
- `vue create myvue-app`
- `cd my-app`
- `npm run serve` 로 실행함

Npm이란

nodejs를 이용해 만든 여러 앱들을 명령어 환경에서 쉽게 설치할 수 있도록 도와주는 도구. node.js계의 앱스토어 역할을 하는 SW임

애플리케이션 실행

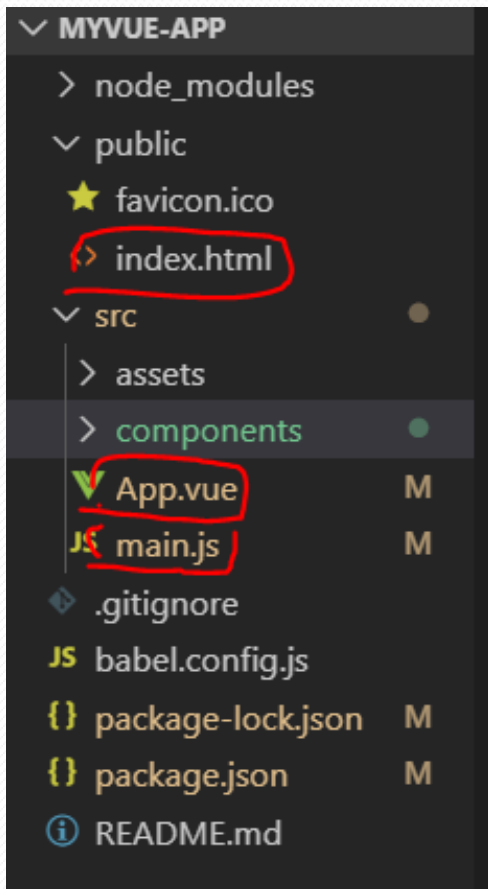
- 터미널에서
- **npm run serve**하면 브라우저가 실행되면서 <http://localhost:8080> url로 화면이 뜨는 것을 확인할 수 있음

개발환경

- [2] Vue-router 설치
- 참고: <https://router.vuejs.org/installation.html>
- Myvue-app 디렉토리에서
- `npm i --s vue-router`

- [3] jquery, bootstrap등도 설치하여 사용해보자.
- `npm i --s jquery bootstrap popper.js`

프로젝트 구조



- main.js => 진입 파일 (entry파일)
- 이 안에서 App컴포넌트를 index.html의 div 아이디가 app인 곳에 보여주도록 함
- public/index.html
- App.vue 파일들을 살펴 보자.

Single File Component

- 싱글파일 컴포넌트는 **html, css, script**를 한 파일로 관리하겠다는 것. 재사용성, 가독성을 높이기 위해...
- Vue를 통해 컴포넌트들을 만들면 **public**안에 **index.html**에 있는 **id가 app인 div**태그 안에 들어가도록 약속되어 있음
- **app**안에 들어갈 컴포넌트들은 **src/components**안에 둔다.
- 또한 **src**안의 **main.js**파일이 진입 파일임(entry file)
- **main.js**에서 기술된 **App**이 컴포넌트임
- **App.vue**에 구성되어 있음

프로젝트에서 사용할 bootstrap, vue-router설치

- `npm i bootstrap --s`
- `npm i vue-router --s`

main.js 에서 vue-router와 bootstrap import

```
import Vue from 'vue'
import App from './App.vue'
import VueRouter from 'vue-router';

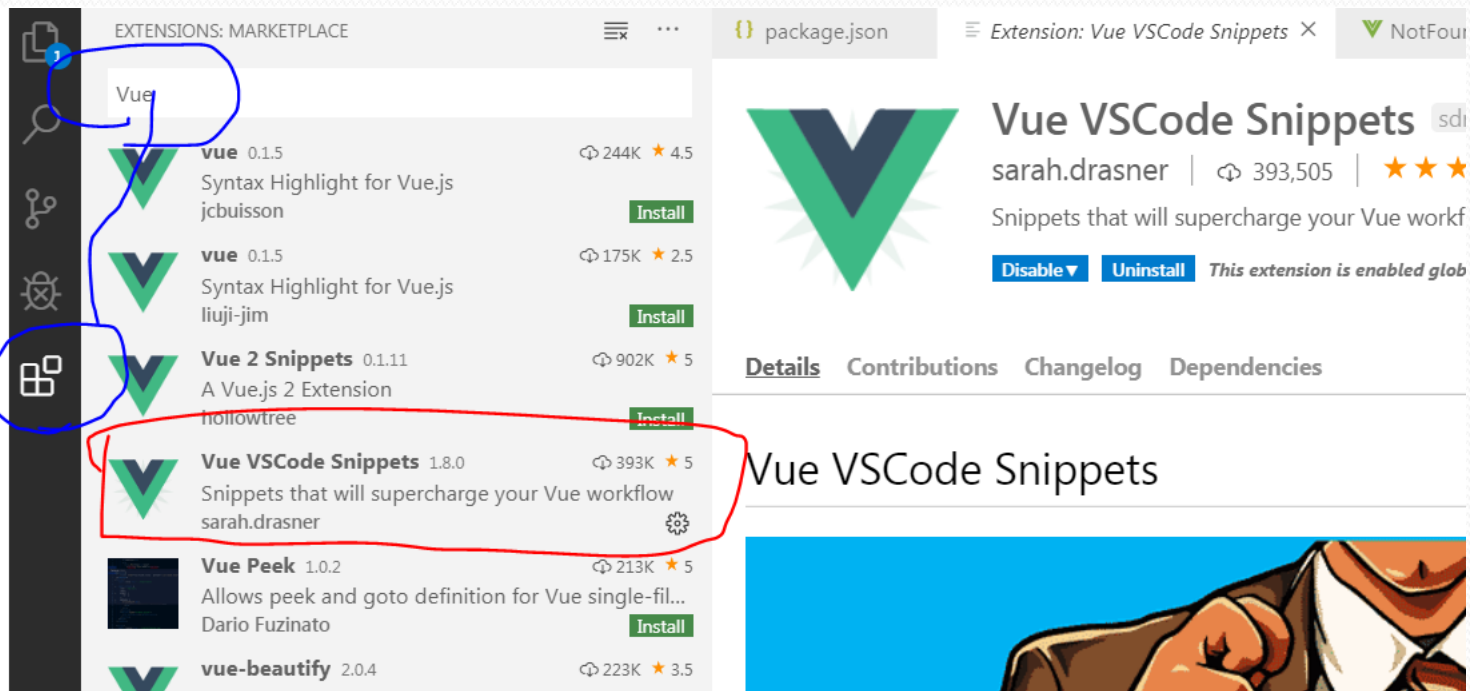
import 'bootstrap'
import 'bootstrap/dist/css/bootstrap.min.css'

Vue.config.productionTip = false
Vue.use(VueRouter);

new Vue({
  render: h => h(App),
  //vue내부적으로 사용하는 함수
  //템플릿을 정의하면 기본적으로 render함수가 동작함
}).$mount('#app')
```

VSCode 에디터에서 확장 플러그인 설치

- Vscode에서 extension에서 vue를 검색하여 **Vue VScode Snippets**을 설치하자.



프로젝트에서 사용할 JS라이브러리 설치

- 프로젝트 진행중 필요한 라이브러리를 npm으로 설치하자. --s 옵션을 주어 설치하면 package.json파일에 해당 라이브러리가 저장된다.
- [1] **axios** : ajax 요청 처리시 필요한 라이브러리
- **npm i axios -s**
- [2] **querystring** : JSON객체를 요청파라미터 문자열로 만들 때 필요한 라이브러리
- **npm i querystring -s**

package.json 파일 확인해 보기

```
package.json X NotFound.vue JS main.js
package.json > {} dependencies
4   "private": true,
5   "scripts": {
6     "serve": "vue-cli-service serve",
7     "build": "vue-cli-service build",
8     "lint": "vue-cli-service lint"
9   },
10  "dependencies": {
11    "axios": "^0.19.0",
12    "bootstrap": "^4.3.1",
13    "core-js": "^3.3.2",
14    "jquery": "^3.4.1",
15    "popper.js": "^1.16.0",
16    "querystring": "^0.2.0",
17    "vue": "^2.6.10",
18    "vue-router": "^3.1.3"
19  },
20  "devDependencies": {
21    "@vue/cli-plugin-babel": "^4.0.0",
22    "@vue/cli-plugin-eslint": "^4.0.0",
23    "@vue/cli-service": "^4.0.0",
24    "babel-eslint": "^10.0.3",
25    "eslint": "^5.16.0",
26    "eslint-plugin-vue": "^5.0.0",
27    "vue-template-compiler": "^2.6.10"
28  },
29  "eslintConfig": {
30    "root": true,
31    "env": {
```

Ajax연동시 proxy설정

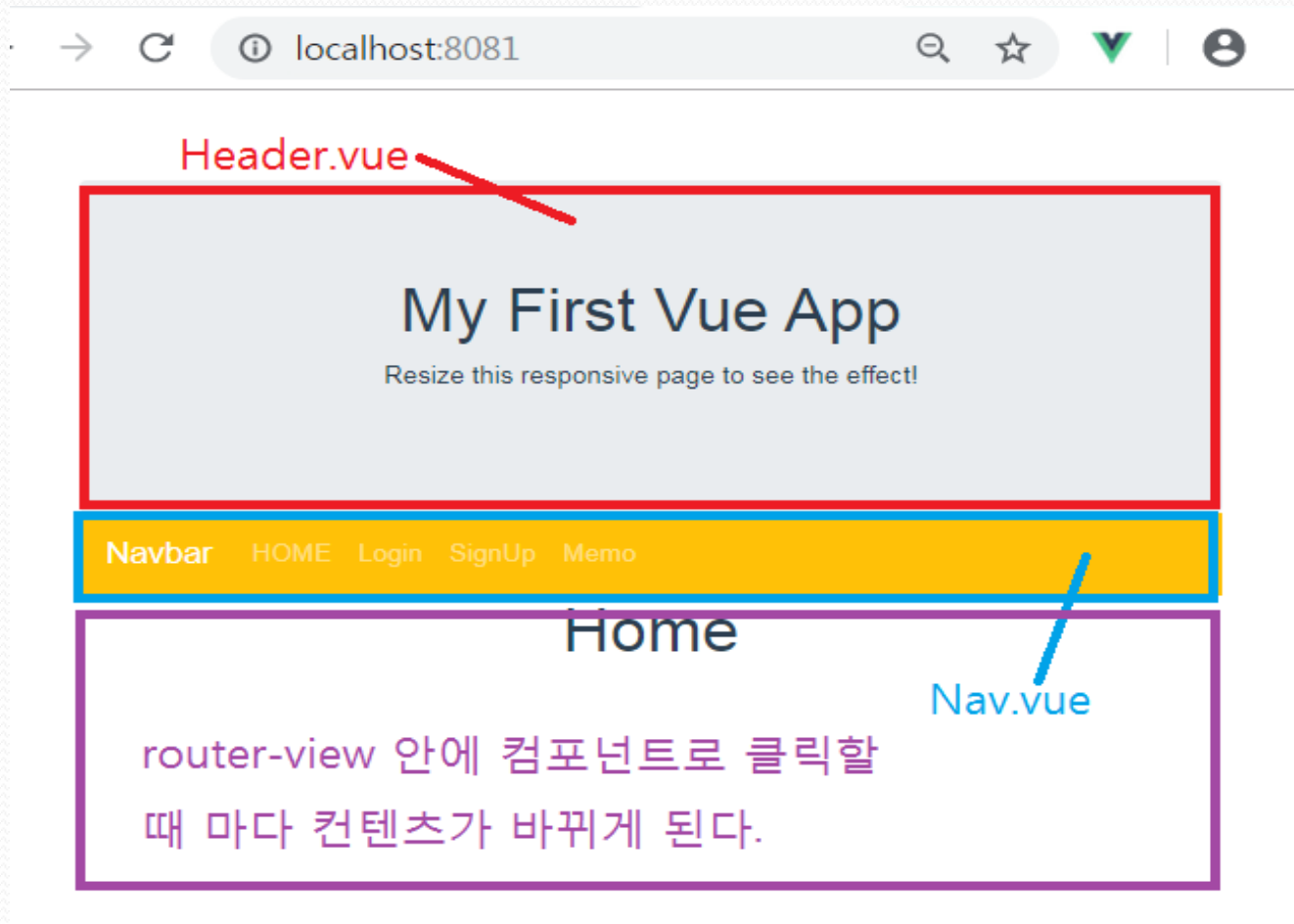
- **proxy란?**
- proxy 는 '대리' 라는 의미로 서로 다른 공간에 있는 두 서버의 중계 기능을 담당한다.
- 우리가 프론트 엔드를 개발할 때, 일반적으로 **http://localhost:8080** 같은 로컬 주소를 사용하게 된다.
- 백 엔드 또한 **port** 를 사용하게 되는데, 기본은 **80** 이나 개발용으로는 **8080** 을 사용하지 못한다. 이미 **8080** 포트를 프론트 엔드 서버에서 사용 중 이므로 백엔스 서버는 다른 포트를 사용하게 되는 것.
- (우리는 **9090 port**를 백엔드서버에서 사용 중...)
- 그래서 **proxy** 로 프론트엔드 서버와 백엔드 서버를 연결해 줄 예정이다.
- 프론트엔드 개발이 끝나고 배포할 경우에는 **proxy** 가 필요하지 않게 되지만(**path** 가 같아지게 되기 때문) 개발과정에서는 프론트엔드와 백엔드를 분리해서 작업하므로 이를 위해 **proxy**설정을 해서 테스트하도록 하자.

Vue.config.js 파일 작성

- 프로젝트 root 디렉토리에 vue.config.js 파일을 작성하자

```
module.exports = {  
  devServer:{  
    proxy:{  
      '/':{  
        target:'http://localhost:9090',  
        changeOrigin:true  
      }  
    }  
  }  
}
```

프로젝트 UI



프로젝트UI-회원가입 페이지

Navbar HOME Login **SignUp** Memo

SingUp

Name:

Nick Name:

아이디는 4자이상 6자 이하 입니다.

PASSWORD:

비밀번호는 4자 이상 8자 이내입니다.

PASSWORD Confirm :

비밀번호가 일치하지 않아요

Tel :

Address :

프로젝트UI-로그인 페이지

My First Vue App

Resize this responsive page to see the effect!

Navbar HOME Login SignUp Memo

Login

아이디가 틀려요!

User ID:

PASSWORD:

Login

프로젝트 UI-한줄 메모장

My First Vue App

Resize this responsive page to see the effect!

Navbar HOME 김길동님 로그인중... Logout SignUp Memo



한줄 메모장

Have a Happy Day~

+

Add Memo

제 목

메모 내용

글쓰기

Memo List

2019-06-01

aaa

2019-11-22

test

2019-11-22

Hi

2019-11-22

제목

메모내용

작성일: a

글수정

★ READ ME

my-app/router/index.js

```
import VueRouter from 'vue-router';
import Vue from 'vue';
import Home from '../components/Home.vue'
import NotFound from '../components/NotFound.vue'
import MemoApp from '../components/MemoApp.vue'
import SignUp from '../components/user/SignUp.vue'
import Login from '../components/user/Login.vue'
import LogOut from '../components/user/LogOut.vue'
Vue.use(VueRouter);
```

```
export default new VueRouter({
```

```
  mode: 'history',
```

```
  base: '/',
```

```
  routes: [
```

```
    {
```

```
      path: '/',
```

```
      name: 'Home',
```

```
      component: Home
```

```
    },
```

```
    {
```

```
      path: '/memo',
```

```
      name: 'MemoApp',
```

```
      component: MemoApp
```

```
    },
```

```
    {
```

```
      path: '/signup',
```

```
      name: 'SignUp',
```

```
      component: SignUp
```

```
    },
```

```
    {
```

```
      path: '/login',
```

```
      name: 'Login',
```

```
      component: Login
```

```
    },
```

```
    {
      path: '/login',
      name: 'Login',
      component: Login
    },
    {
      path: '/logout',
      name: 'LogOut',
      component: LogOut
    },
    {
      path: '/*',
      name: 'NotFound',
      component: NotFound
    }
  ]
})
```

main.js 에서 router/index.js

router의 index.js
참조후 Vue인스
턴스에 router 등
록

```
JS main.js  X  App.vue  MemoApp.vue  IntroContent.v
src > JS main.js > ...

1  import Vue from 'vue'
2  import App from './App.vue'
3  import router from './router';
4
5  import 'bootstrap'
6  import 'bootstrap/dist/css/bootstrap.min.css'
7
8  Vue.config.productionTip = false
9
10 new Vue({
11   render: h => h(App),
12   //vue내부적으로 사용하는 함수
13   //템플릿을 정의하면 기본적으로 render함수가 동작함
14   /**
15    * render:function(h){
16    *   return h(App)
17    * }
18    *
19    */
20   router
21 }).$mount('#app')
22
```

App.vue 에 컴포넌트 등록 <router-view />삽입

```
1 <template>
2   <div id="app" class="container">
3     <header-comp></header-comp>
4     <nav-comp></nav-comp>
5     <router-view> </router-view>
6   </div>
7 </template>
8
9 <script>
10 import Header from './components/Header.vue'
11 import Nav from './components/Nav.vue'
12 export default {
13   name: 'app',
14   data(){
15     return {
16       status:'',
17       loginFlag:false, //로그인 여부를 가질 데이터
18       loginUser:'' //로그인한 사람 닉네임을 가질 데이터
19     }
20   },
21   components: {
22     'header-comp':Header,
23     'nav-comp':Nav
24   },
25 }
```

Props의 특징

- 상위 컴포넌트가 하위 컴포넌트에 값을 전달할 때는
- **props**를 통해 전달한다.
- 하지만 하위 컴포넌트가 상위 컴포넌트의 값을 바꾸고자 할 때는 **event**를 통해 함.
- 그 이벤트가 실행되었을 때
- 상위 컴포넌트의 **data**를 변경하는 것을 통해 영향을 미침

참고 사이트

- vue.js 공식 문서: <https://vuejs.org/v2/guide/>
- vue.js 스타일 가이드: <https://vuejs.org/v2/style-guide/>
- Vue.js 쿡북: <https://vuejs.org/v2/cookbook/>

- vuex 공식 문서: <https://vuex.vuejs.org/>
- vurouter 공식 문서: <https://router.vuejs.org/>
- vue cli 공식 문서: <https://cli.vuejs.org/>

- Ajax 테스트 데이터 서비스 사이트:
<https://jsonplaceholder.typicode.com/users>