

- SPA : F5(새로고침)을 해도 전체 내용을 reloading 하지 않고 변경된 내용만 적용
- Component : HTML, CSS, JS가 세팅되어 있음
- V8 + node의 등장으로 Java 를 Server 단에서 실행
- 새로운 프로젝트 디렉토리 생성시
  - 해당 디렉토리로 이동 c:/work/vue/vueexm
  - Terminal : npm install
  - npm run serve

## 1) App.vue

```
vueexm > src > App.vue > ...
1 |
2 | <template>
3 |   <div id="appPage">
4 |     <router-view/> <!-- 템플릿파일을 웹서버에 출력하기위
5 |   </div>
6 | </template>
```

1. 전체 화면 출력하는 것이 app.vue다  
모든 것에 시작

시작위치라고 알려주는 것이 router-view

## 2) index.vue : 페이지에 대한 규격

```
vueexm > src > layout > index.vue > Vuter > template
1 <template>
2   <section id = "warp">
3     <section>상단</section>
4     <section id = "container">
5       <router-view /><!--메뉴에 따른 콤포너트 삽입 할 곳 -->
6     </section>
7     <section>하단</section>
8   </section>
9 </template>
```

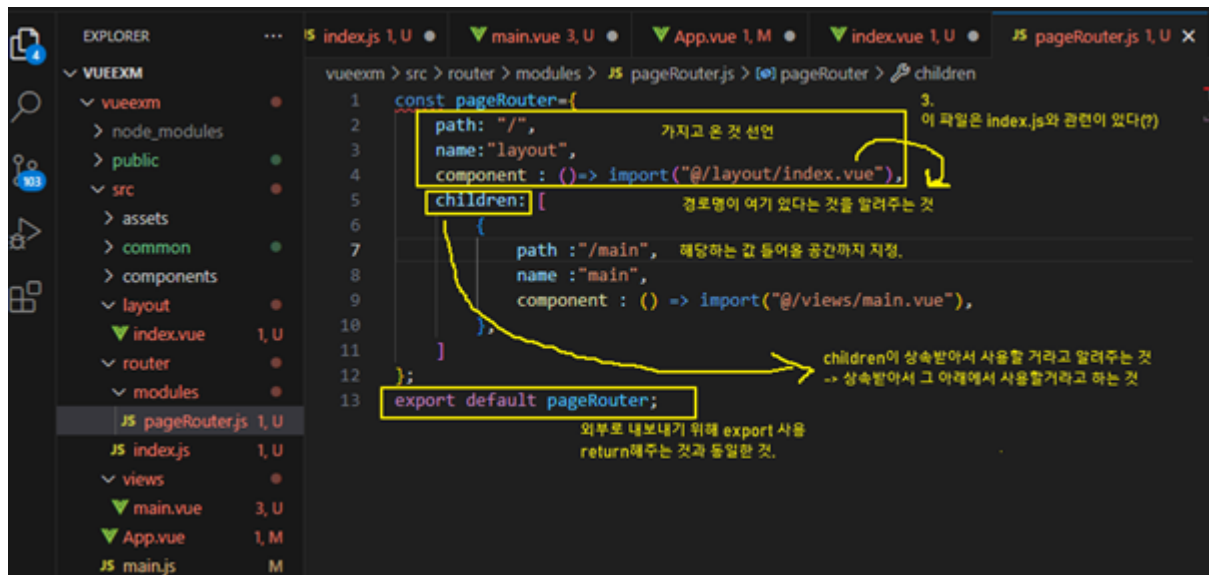
layout이 전체 내용을 의미.

2. 그 다음에 index.vue를 만들었다.(layout 안에)  
웹사이트에 대한 큰 골격을 만든 것

상단 - 헤더  
하단 - 푸터  
router-view - main 부분의 자리

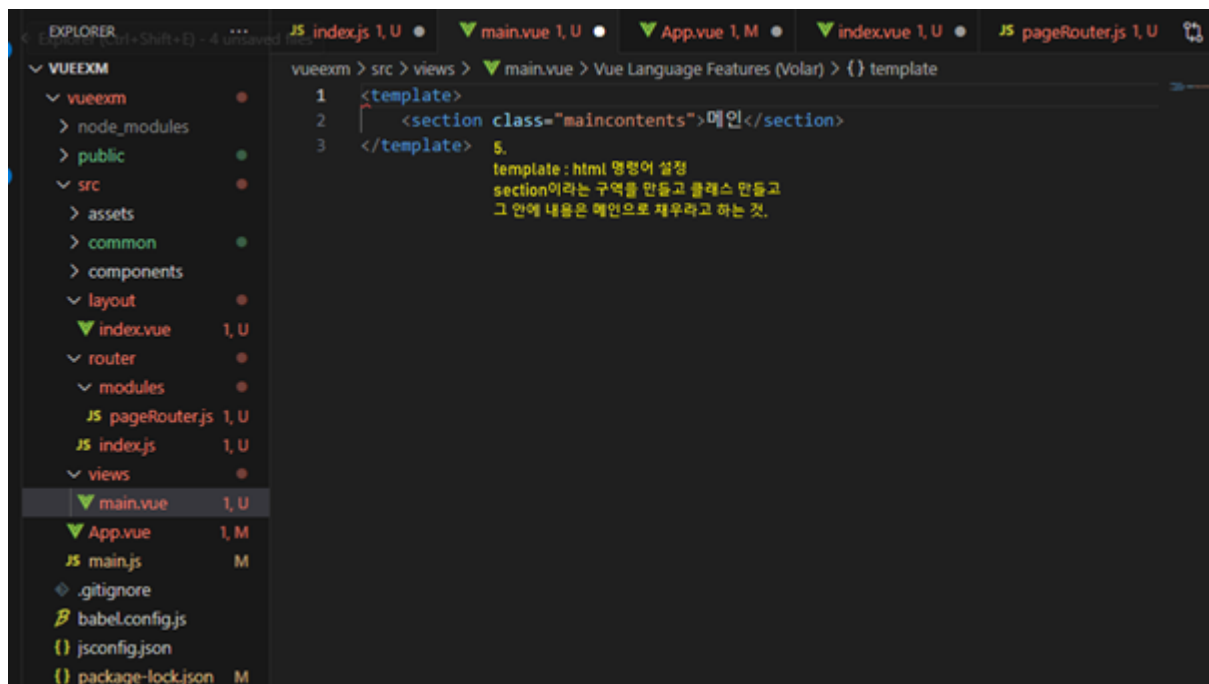
- router : 컴포넌트 전체 경로설정에 대한 큰 틀
  - modules 에서도 뭔가 세팅??

### 3) pageRouter



- export : 함수, 변수, 클래스 등 내보낼 때
  - export {식별자}
- import : 자바스크립트 파일에서 가져올 때
  - import {식별자} from '경로'
- router(경로설정) 설정은 경로설정할게 많아질수록 필요, 코드수정시 파일위치확인 번거로움

### 5) main.vue



## 6) index.js

```
vueexm > src > router > JS index.js > [🔍] router
1  import Vue from "vue";
2  import VueRouter from "vue-router";
3  import pageRouter from "../modules/pageRouter";
4  Vue.use(VueRouter);
5  const router = new VueRouter({
6    mode: "history",
7    routes: [pageRouter],
8  });
9  export default router;
```

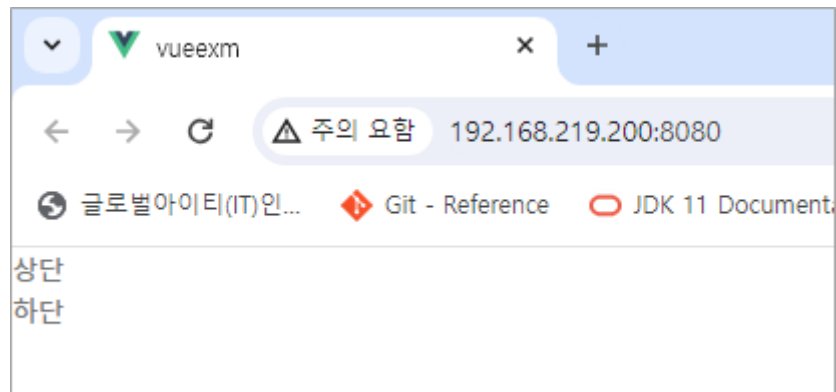
```
vueexm > src > router > modules > JS pageRouter.js > [🔍] default
1  const pageRouter = {
2    path: "/",
3    name: "layout",
4    component: () => import("@/layout/index.vue"),
5    children: [
6      {
7        path: "/main",
8        name: "main",
9        component: () => import("@/views/main.vue"),
10      },
11    ],
12  };
13  export default pageRouter;
```

- component : () => import("@/layout/index.vue") 이 index.vue 의 내용을 의미

```
vueexm > src > layout > index.vue > Vuter > [🔍] template
1  <template>
2    <section id = "warp">
3      <section>상단</section>
4      <section id = "containar">
5        <router-view /><!--메뉴에 따른 콤포너트 삽입 할 곳 -->
6      </section>
7      <section>하단</section>
8    </section>
9  </template>
```

이 내용이다!!!!

- local



7) 중간 정리 : 해당 순서로 진행됨

7-1) App.vue : 모든 것의 시작점 (1)

- SPA 의 특징 : vue 는 node 상에서 구동되므로 필요한 값만 browser에게 전달하므로 경로 설정이 필요, 다소 느려짐
- App.vue 는 모든 조각을 조립, 다른 component 들의 부모 component
- <router-view> : 경로 지정, vue 컴포넌트들을 호출 root 라 설정 불필요

7-2) index.vue : main.vue 와 구별, 최종적으로 HTML 프레임의 정의 (2-1)

- <template> : html 이 작성됨, 각각의 컴포넌트로 구성됨(vue 파일로 만들어짐)
  - header , footer
- <router-view /> : 경로 지정, index에서만 호출
- <style> <script> : CSS, JS

7-3) index.js : index.vue의 자바스크립트 파일, (2-2)

- const router = new VueRouter({ routes: [ pageRouter ] })
  - routes 에서 pageRouter 경로가 연결됨
- import VueRouter from "vue-router"; : VueRouter 전체 권한 필수, 필요한 부분만 호출하기 위해, 빠른 실행을 위한 권한 설정

7-4) main.vue : (3-1)

- 

7-5) main.js : (3-2)

- 

7-6) pageRouter.js : SPA 계열에서의 경로 설정, JS 로 제작 (4)

- path : "/" 는 root를 의미 (현재 c:/vue/vueexm/src)
- redirect : "/main"
  - index.vue 를 부를때 main 호출이 아니며 main이 열리지 않으므로 강제로 열리게 함
  - children 이 상속받음
  - path: "/main" : 상속 받을 경로가 /main
  - 현재 main.vue 는 index.vue 에 종속됨
- component : 경로 설정, 이 부분에서 index.vue 와 main.vue가 연결됨
  - (2) 와 (3) 의 연결지점

## 8) v-for.html , v-if.html

```
<!-- Vue.js CDN 추가 -->
<script src="https://cdn.jsdelivr.net/npm/vue@2"></script>
</head>
<body>
<div id="app">
  <ul>
    <!-- 배열의 각 항목을 반복하면서 리스트 아이템을 생성 -->
    <li v-for="item in items">{{ item }}</li>
  </ul>
</div>
<script>
  new Vue({ // Vue 인스턴스 생성
    el: '#app',
    data: {
      items: ['Apple', 'Banana', 'Orange', 'Grapes'] // 배열 데이터
    }
  });
</script>

</body>
</html>
```

- `el: '#app'` 을 마운트 지정
  - `#app` 는 `<div id="app">`

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Vue.js v-if 예제</title>
  <script src="https://cdn.jsdelivr.net/npm/vue@2"></script>
</head>
<body>

<div id="app">
  <!-- 조건에 따라 엘리먼트를 표시하거나 숨기기 -->
  <p v-if="showMessage">이 메시지는 표시됩니다.</p>
  <p v-else>이 메시지는 숨겨집니다.</p>
</div>
<script>
  new Vue({ // Vue 인스턴스 생성
    el: '#app',
    data: { // 조건을 나타내는 데이터
      showMessage: true
    }
  });
</script>

</body>
</html>
```

- vue 는 node 는 js를 실행하기 위한 runtime(일시적 운영체제)이자, jvm (가상), webserver 처럼 보이지만 아니다. 실질적으로는 사용하지만 동일한 것은 아님
- CSS : common > style > .css
  - Terminal : npm install bootstrap-vue 로 설치 가능