SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

삼성 청년 SW 아카데미

Vue.js

SAMSUNG SW ACADEMY FOR YOUTH

FrontEnd Framework

Vue Router

- Vue Router

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Vue Router

Vue Router

- 라우팅 : 웹 페이지 간의 이동 방법
- Vue.js 공식 라우터
- 라우트(route)에 컴포넌트를 매핑한 후, 어떤 주소에서 렌더링할 지 알려줌.
- SPA 상에서 라우팅을 쉽게 개발할 수 있는 기능을 제공
- URL에 따라 컴포넌트를 연결하고 설정된 컴포넌트를 보여줌.

https://v3.router.vuejs.org/kr/

Vue Router 설치

- CDN 방식

  <script src="/path/to/vue.js"></script>

  <script src="/path/to/vue-router.js"></script>

- NPM 방식

  npm install vue-router

- Vue CLI

  vue add router

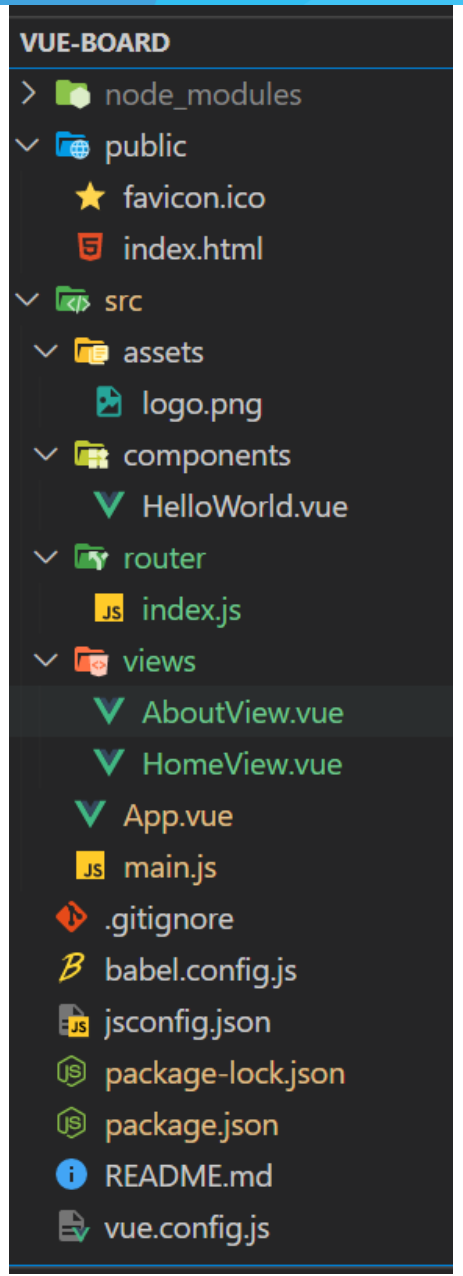
  (프로젝트를 진행하던 중에 추가를 하게 되면 App.vue를 덮어쓰므로 백업을 해두고 추가할 것)

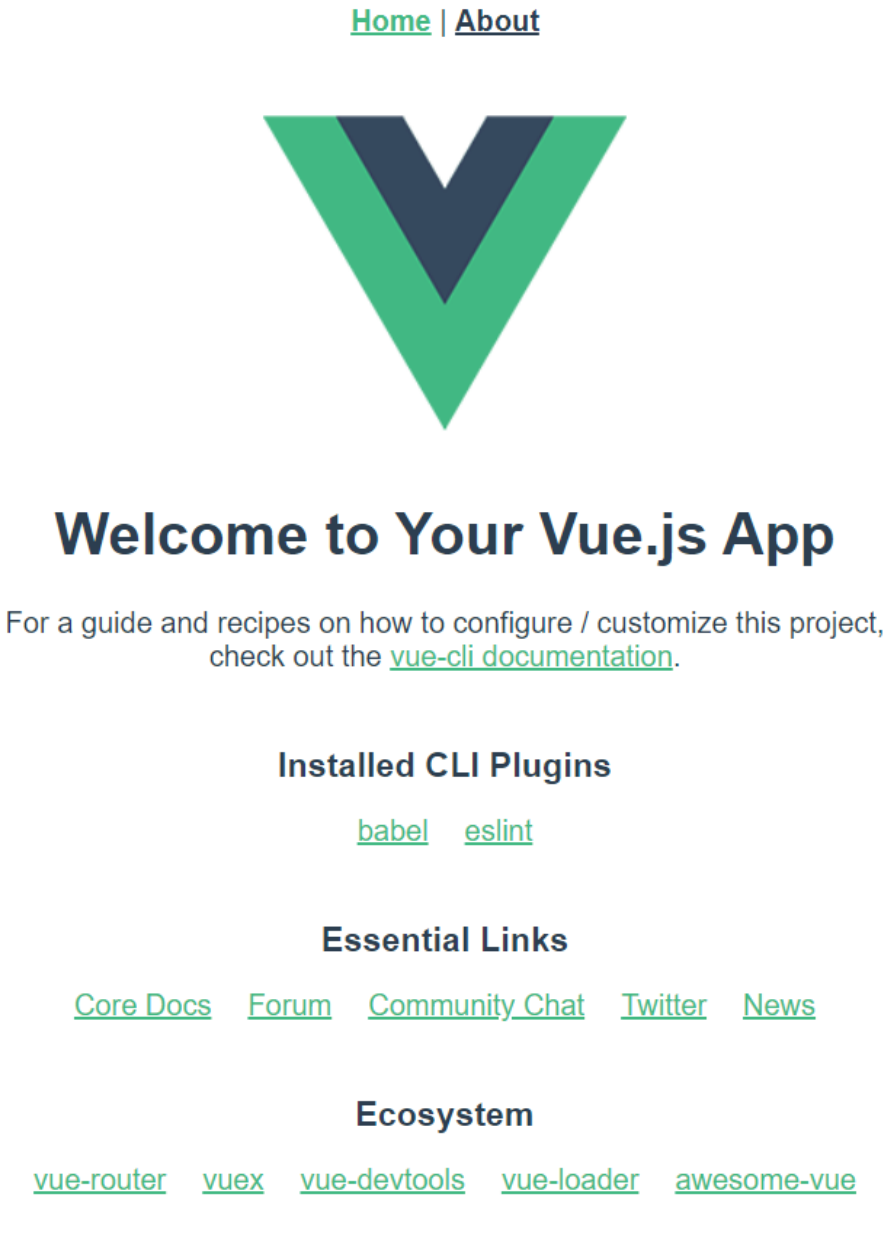
Vue Router 설치 CLI

```
$ vue add router

📦  Installing @vue/cli-plugin-router...


up to date, audited 927 packages in 3s

96 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
✔  Successfully installed plugin: @vue/cli-plugin-router

? Use history mode for router? (Requires proper server setup for index fallback in production) Yes

🚀  Invoking generator for @vue/cli-plugin-router...
📦  Installing additional dependencies...


added 1 package, and audited 928 packages in 2s

96 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
⚓  Running completion hooks...

✔  Successfully invoked generator for plugin: @vue/cli-plugin-router
```

Vue Router 프로젝트 변화

- App.vue 코드
- router/index.js 생성
- views 디렉토리 생성





components 와 views

- 두개의 폴더는 각기 SFC 파일을 저장함.
- 어느 곳에 저장해도 상관은 없지만 주로 아래와 같이 작성.
- App.vue : 최상위 컴포넌트
- views/ : router(index.js)에 매핑되는 컴포넌트를 모아두는 폴더
- components/ : router에 매핑된 컴포넌트 내부에 작성하는 컴포넌트를 모아두는 폴더

Vue Router - index.js

- 라우트에 관련된 정보 및 설정 작성

```
import Vue from 'vue'
import VueRouter from 'vue-router'
import HomeView from '../views/HomeView.vue'

Vue.use(VueRouter)

const routes = [
  {
    path: '/',
    name: 'home',
    component: HomeView
  },
  {
    path: '/about',
    name: 'about',
    // route level code-splitting
    // this generates a separate chunk (about.[hash].js) for this route
    // which is lazy-loaded when the route is visited.
    component: () => import(/* webpackChunkName: "about" */ '../views/AboutView.vue')
  }
]

const router = new VueRouter({
  mode: 'history',
  base: process.env.BASE_URL,
  routes
})

export default router
```

Vue Router - <router-link>

- 사용자 네비게이션을 가능하게 하는 컴포넌트
- 목표 위치는 'to' prop로 지정됨.
- 기본적으로는 올바른 href를 갖는 <a> 태그로 렌더링 되지만 'tag' prop로 구성될 수 있음.
- HTML5 히스토리 모드에서 router-link는 클릭 이벤트를 차단하여 브라우저가 페이지를 다시 로드하지 않도록 함.

```
<router-link to="/">Home</router-link>
<router-link to="/about">About</router-link>
```

https://v3.router.vuejs.org/kr/api/#router-link

Vue Router - <router-view>

- 주어진 라우트에 대해 일치하는 컴포넌트를 렌더링 하는 함수형 컴포넌트
- 실제 component가 DOM에 부착되어 보이는 자리를 의미.
- router-link를 클릭하면 해당 경로와 연결되어 있는 index.js에 정의한 컴포넌트가 위치

```
<router-view/>
```

https://v3.router.vuejs.org/kr/api/#router-view

Vue Router - History Mode

- vue-router의 기본 모드는 hash mode. (URL에 #이 붙어 URL이 변경될 때 페이지가 다시 로드 X)
- 해시를 제거하기 위해 HTML History API를 사용해서 router를 구현함.
- 페이지를 다시 로드 하지 않고 URL을 탐색할 수 있음.

https://v3.router.vuejs.org/kr/guide/essentials/history-mode.html#html5-historymode

이름을 가지는 라우트 (Named Routes)

- 라우트에 이름을 명명할 수 있음.
- 명명된 경로로 이동하려면 객체를 router-link 컴포넌트의 to로 전달할 수 있음.

```
import Vue from 'vue'
import VueRouter from 'vue-router'
import HomeView from '../views/HomeView.vue'
import AboutView from '../views/AboutView.vue'

Vue.use(VueRouter)

const routes = [
  {
    path: '/',
    name: 'home',
    component: HomeView
  },
  {
    path: '/about',
    name: 'about',
    component: AboutView
  }
]

const router = new VueRouter({
  mode: 'history',
  base: process.env.BASE_URL,
  routes
})

export default router
```

```
<template>
  <div id="app">
    <nav>
      <router-link to="/">Home</router-link> |
      <router-link :to="{name: 'about'}">About</router-link>
    </nav>
    <router-view/>
  </div>
</template>
```

https://v3.router.vuejs.org/kr/guide/essentials/named-routes.html

프로그래밍 방식 네비게이션 (Programmatic Navigation)

- <router-link>를 사용하여 선언적 네비게이션용 <a> 태그를 만드는 것 외에도 router의 인스턴스 메소드를 사용하여 프로그래밍으로 이를 수행할 수 있음.
- Vue 인스턴스 내부에서 라우터 인스턴스에 $router로 접근할 수 있음.
- 따라서 다른 URL로 이동하려면 this.$router.push를 호출할 수 있음
  해당 메서드는 새로운 항목을 히스토리 스택에 넣기 때문에 사용자가 브라우저의 뒤로 가기 버튼을 클릭하면 이전 URL로 이동하게 됨.
- <router-link>를 클릭 할 때 내부적으로 호출되는 메소드이므로
  <router-link :to="…"> 를 클릭하면, router.push(…)을 호출하는 것과 같음.

https://v3.router.vuejs.org/kr/guide/essentials/navigation.html

프로그래밍 방식 네비게이션 (Programmatic Navigation)

- 작성 예시

```
// 리터럴 string
router.push('home')

// object
router.push({ path: 'home' })

// 이름을 가지는 라우트
router.push({ name: 'user', params: { userId: 123 }})

// 쿼리와 함께 사용, 결과는 /register?plan=private 입니다.
router.push({ path: 'register', query: { plan: 'private' }})
```

https://v3.router.vuejs.org/kr/guide/essentials/navigation.html

동적 라우트 매칭 (Dynamic Route Matching)

- 주어진 패턴을 가진 라우트를 동일한 컴포넌트에 매핑해야 하는 경우.
- 동적 인자 전달
- 예를 들어 모든 User에 대해 동일한 레이아웃을 가지지만, 다른 User ID로 렌더링 되어야 하는 예시

```
const routes = [
    // 동적 세그먼트는 콜론으로 시작합니다.
    { path: '/user/:id', component: User }
]
```

- 위의 예에서 /user/foo 와 /user/bar 같은 URL은 모두 같은 경로에 매핑

https://v3.router.vuejs.org/kr/guide/essentials/dynamic-matching.html

동적 라우트 매칭 (Dynamic Route Matching)

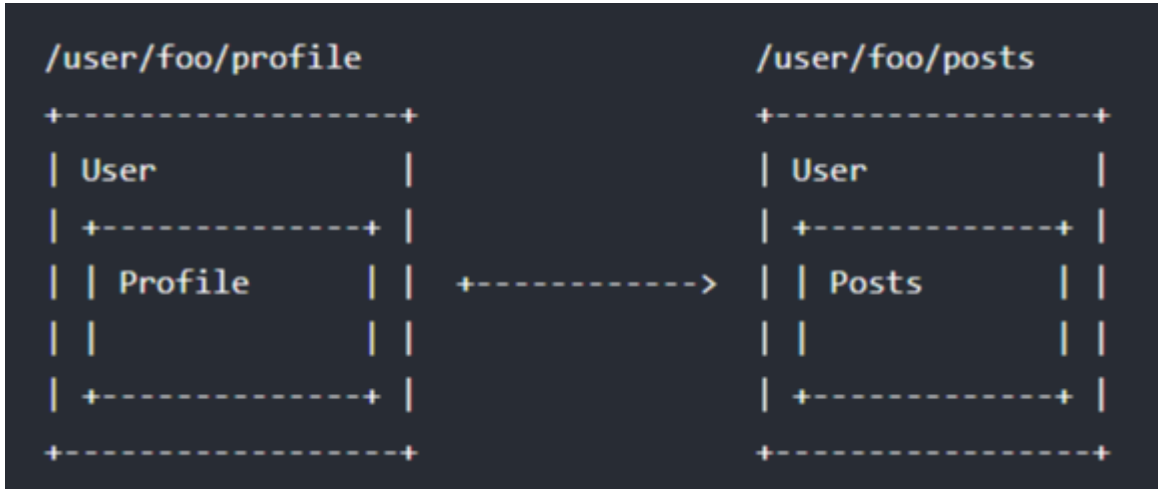
- 동적 인자는 : (콜론)으로 시작
- 컴포넌트에서 this.$route.params 로 사용 가능

패턴	일치하는 패스	$route.params
/user/:username	/user/evan	{ username: 'evan' }
/user/:username/post/:post_id	/user/evan/post/123	{ username: 'evan', post_id: '123' }

https://v3.router.vuejs.org/kr/guide/essentials/dynamic-matching.html

중첩된 라우트 (Nested Routes)

- 실제 앱 UI는 일반적으로 여러 단계로 중첩된 컴포넌트로 구조임.
- URL의 세그먼트가 중첩된 컴포넌트의 특정 구조와 일치하는 것을 활용



https://v3.router.vuejs.org/kr/guide/essentials/nested-routes.html

중첩된 라우트 (Nested Routes)

```
{
  path: "/board",
  name: "board",
  component: BoardView,
  children: [
    {
      path: "list",
      component: BoardList,
    },
    {
      path: "write",
      component: BoardWrite,
    },
  ],
},
```

```
<template>
  <div>
    <h3>게시판</h3>
    <router-link to="/board/list">게시판 목록</router-link>
    <router-link to="/board/write">게시글 등록</router-link>
    <router-view/>
  </div>
</template>
```

리다이렉트 (Redirect)

- routes 설정

```
const router = new VueRouter({
  routes: [
    { path: '/a', redirect: '/b' }
  ]
})
```

- 이름이 지정된 라우트 지정

```
const router = new VueRouter({
  routes: [
    { path: '/a', redirect: { name: 'foo' }}
  ]
})
```

- 동적 리다이렉트

```
const router = new VueRouter({
  routes: [
    { path: '/a', redirect: to => {
      // 함수는 인수로 대상 라우트를 받습니다.
      // 여기서 path/location 반환합니다.
    }}
  ]
})
```

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

다음 방송에서 만나요!

삼성 청년 SW 아카데미