Vue Router-1

vue에서 라우팅 처리를 위해 사용하는 플러그인

- 라우팅 : 사용자가 접속한 주소에 따라 페이지(컴포넌트)가 달라지는 것

> Vue Router
> The official Router for Vue.js
> https://router.vuejs.org/guide/

라우터란? (Router)

- 라우터라고 하면 일반적으로 네트워크간에 데이터를 전송하는 장치를 말합니다.
- 뷰에서 말하는 라우터는 쉽게 말해서 URL에 따라 어떤 페이지를 보여줄지 매핑 해주는 라이브러리라고 보시면 됩니다

예를 들어 `/home` 경로로 요청이 들어왔을때 `Home.vue` 컴포넌트를 화면에 렌더링 해라!" 라는 역할을 수행하는 라이브러리 입니다

그리고 이때 `/home` → `Home.vue` 이러한 매핑정보를 라우트(Route)라고도 합니다.

라우트란? (Route)

어떤 URL에 대해 어떤 페이지를 표시해야 하는지에 대한 정보

설치

```
npm i vue-router
```

페이지 컴포넌트 생성

`HomeView.vue`와 `AboutView.vue`라는 **페이지용 컴포넌트**를 만든후 `'/'` 경로로 들어왔을 경우 `HomeView.vue` 페이지(컴포넌트)를 렌더링 하고 `/about` 경로로 들어왔을 경우 `AboutView.vue` 페이지(컴포넌트)를 렌더링 하는 실습을 진행해 보도록 하겠습니다.

- `'/'` → `HomeView.vue`
- `'/about'` → `AboutView.vue`

`HomeView.vue`와 `AboutView.vue` 페이지(컴포넌트)를 생성

src/views/HomeView.vue

```
<template>
  <h1>Home Page</h1>
</template>

<script></script>
```

src/views/AboutView.vue

```
<template>
  <h1>About Page</h1>
</template>

<script></script>
```

src/router/index.js 파일 생성 ⇒ 컴포넌트 등록

```
//1. vue-router를 연결(임포트)
import { createRouter, createWebHistory } from 'vue-router'
import HomeView from '@/views/HomeView.vue'
import AboutView from '@/views/AboutView.vue'

// 2. 경로를 정의하고, 각 경로를 컴포넌트와 매핑.
const routes = [
  {
      //path : 브라우저에서 접속하는 url주소를 정의
    path: '/',
    name: 'home',
      // component : 지정된 path로 들어왔을 때 보여줄 vue 컴포넌트, 앞으로 구현할 vue파일을 연결하고, 해당 파일을 실행 시킵니다.
    component: HomeView  // 사용자가 해당 path에 접근하지 않더라도 이미 vue 파일을 import함
  },
```

```
  {
    path: '/about',
    name: 'about',
    component: AboutView
        // path에 접근하기 전까지는 vue파일에 대한 import가 일어나지 않음
    }
]

// 3. `routes`를 옵션으로 전달해 라우터 인스턴스를 생성.
const router = createRouter({
  history: createWebHistory('/'),
  routes
})

//4. router 내보내기
export default router
```

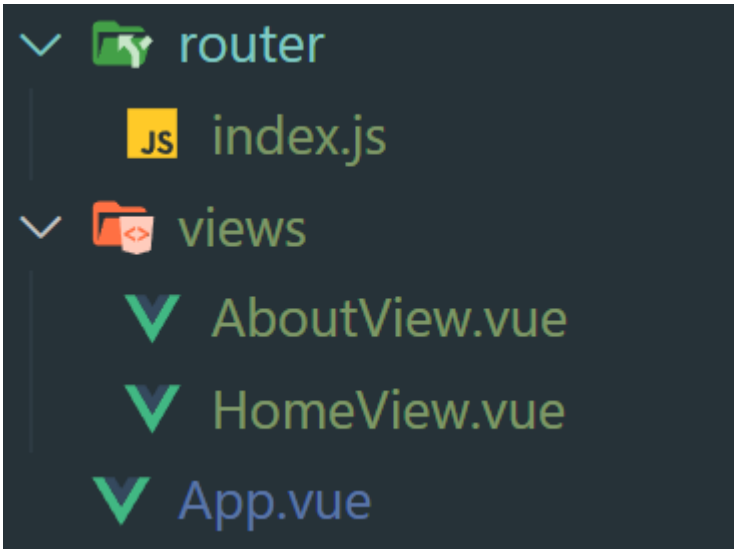
main.js 에 router 추가

```
import { createApp } from 'vue'
import App from './App.vue'
//router임포트
import router from './router'

// 앱이 라우터를 인식하도록, 라우터 인스턴스가 use()로 등록 됨
createApp(App).use(router).mount('#app');
```

`app.use(router)`를 호출 함으로써 컴포넌트 내부에서 `$router`, `$route` 객체에 접근할 수 있습니다.

router로 연결할 HomeView와 AboutView컴포넌트가 view폴더에 추가됨



네이게이션

뷰 라우터를 HTML과 JavaScript로 사용하는 방법에 대해 알아보도록 하겠습니다.

App.vue

```
<template>
  <nav>
    <!-- 탐색을 위해 router-link 컴포넌트를 사용. -->
     <!-- `to`라는 prop으로 링크를 지정. -->
     <!-- `<router-link>`는 `href` 속성이 있는 `<a>` 태그로 렌더링됨. -->
    <RouterLink to="/">Home</RouterLink>
    <span> | </span>
    <RouterLink to="/about">About</RouterLink>
  </nav>
  <main>
     <!-- 경로 출력 -->
     <!-- 현재 경로에 매핑된 컴포넌트가 렌더링됨. -->
     <RouterView></RouterView>
     </main>
</template>
```

- `<RouterLink>`

Vue Router 에서는 페이지를 이동할 때는 일반 `a` 태그를 사용하는 대신 **커스텀 컴포넌트인** `<RouterLink>` 를 사용하여 다른 페이지 링크를 만들어야 합니다.

이를 통해 **Vue Router는 페이지를 리로딩 하지 않고 URL에 매핑된 페이지를 렌더링**할 수 있습니다.

- `<RouterView>`

  `<RouterView>` 는 URL에 매핑된 컴포넌트를 화면에 표시합니다.

useRouter() , useRoute() 훅으로 페이지 이동

위에서 `router` 를 설정할 때 `app.use(router)` 를 호출했습니다.

이렇게 호출 함으로써 모든 자식 컴포넌트에 `router` , `route` 같은 객체를 사용할 수 있습니다.

그리고 이러한 객체는 페이지 이동 또는 현재 활성 라우트(경로 매핑)정보 에 접근하는 데 사용할 수 있습니다.

- `router`

  라우터 인스턴스로 JavaScript에서 다른 페이지(컴포넌트)로 이동할 수 있다.

  - Options API : **this.$router**
  - Composition API : **useRouter()**
  - template : $router

- `route`

  현재 활성 라우트 정보에 접근할 수 있다. (이 속성은 읽기 전용 입니다.)

  - Options API : **this.$route**
  - Composition API : **useRoute()**
  - template : $route

HomeView.vue 수정

```
<template>
  <h1>Home Page</h1>
  <button @click="goAboutPage">About 페이지로 이동</button>
</template>

<script>
```

```
//useRoute와 useRouter훅 사용
import { useRoute, useRouter } from 'vue-router';

export default {
  setup() {
    const router = useRouter();
    const route = useRoute();
    console.log('route.name: ', route.name);
    console.log('route.path: ', route.path);
    const goAboutPage = () => router.push('/about');

    return {
      goAboutPage
    }
  }
}
</script>
```

AboutView.vue 수정

```
<template>
  <h1>About Page</h1>
  <ul>
    <li>$route.name: {{ $route.name }}</li>
    <li>$route.path: {{ $route.path }}</li>
  </ul>
  <button @click="$router.push('/')">Home 페이지로 이동</button>
</template>
```