

Vue Router 설정

🕒 생성일	@2024년 11월 4일 오후 12:36
☰ 태그	

- 라우팅이란?
 - 클라이언트에서 접속한 주소에 따라 페이지(컴포넌트)가 전환되는 것
- Vue-Router 설치

공식 사이트

Vue Router

The official Router for Vue.js

🟢 <https://router.vuejs.org/>

▼ 공식 한국어 버전

Vue Router

Vue.js 공식 라우터

🟢 <https://v3.router.vuejs.org/kr/>

- 태그 종류

종류	설명
<code><router-link to="path" /></code>	네비게이션, 기본적으로 ' <code><a></code> ' 태그로 렌더링
<code><router-view /></code>	라우트 아울렛, 현재 경로에 맞는 컴포넌트 렌더링

- Route 기본 옵션

옵션	설명
<code>path</code>	브라우저에서 접속하는 경로 (url)
<code>name</code>	라우트에 연결하거나 탐색할때 사용하는 이름
<code>component</code>	지정된 path로 접속했을 때 보여줄 컴포넌

- Lazy Load 적용하기(비동기 컴포넌트)
 - Lazy Load
 - 리소스를 컴포넌트 단위로 분리
 - 컴포넌트 혹은 라우터 단위로 필요한 것들만 그때 그때 다운로드
 - 내부적으로 Vue CLI의 prefetch 기능 사용
 - prefetch
 - 미래에 사용될 수 있는 리소스를 캐시에 저장
 - 장점
 - 사용자가 접속했을 때 빠르게 리소스를 다운로드
 - 단점
 - 사용 시점과 상관없이 모든 리소스를 저장하므로 성능 저하
 - 사용
 - vue.config.js

```
module.exports = {
  // 기본으로 적용된 prefetch를 해제
  chainWebpack: config => {
    config.plugins.delete('prefetch');
  }
}
```

2. 적용될 컴포넌트

```
import(/* webpackPrefetch: true */ './views/About.vue');
```

▼ 특수 주석 문법(Magic Comments)

Module Methods | 웹팩

웹팩은 모듈 번들러입니다. 주요 목적은 브라우저에서 사용할 수 있도록 JavaScript 파일을 번들로 묶는 것이지만, 리소스나 애셋을 변환하고 번들링 또는 패키징할 수도 있습니다.

 <https://webpack.kr/api/module-methods/>



- Dynamic import 문
- 특정 기능을 작동시키기 위한 인라인 주석
- chunk 파일의 이름을 지정하거나 다른 모드를 선택
 - ▼ chunk
 - 하나의 번들 파일을 여러가지의 파일로 다시 나누는 것
- 종류

```
// Single target
import(
  /* webpackChunkName: "my-chunk-name" */
  /* webpackMode: "lazy" */
  /* webpackExports: ["default", "named"] */
  /* webpackFetchPriority: "high" */
  'module'
);

// Multiple possible targets
import(
  /* webpackInclude: /\.json$/ */
  /* webpackExclude: /\.noimport\.json$/ */
  /* webpackChunkName: "my-chunk-name" */
  /* webpackMode: "lazy" */
  /* webpackPrefetch: true */
  /* webpackPreload: true */
  `./locale/${language}`
);
```

```
import(/* webpackIgnore: true */ 'ignored-module.js');
```

- `/* webpackChunkName: "my-chunk-name" */`
 - 코드를 분할 후 별도의 chunk 파일을 생성하며 해당 파일의 이름을 지정
- `/* webpackPrefetch: true */`
 - 해당 컴포넌트에 prefetch 기능 적용